Méthodes optimistes d'apprentissage actif pour la classification

Timothé Collet



UNIVERSITE DE LORRAINE

Ecole Doctorale : IAEM Lorraine
Département : Informatique
Spécialité : Informatique

Thèse

présentée pour l'obtention du titre de

Docteur en Sciences de l'Université de Lorraine, Metz-Nancy

par **Timothé COLLET**

Méthodes Optimistes d'Apprentissage Actif pour la Classification

Soutenue le 15 Juillet 2016

Membres du jury

Rapporteurs :	Ludovic Denoyer	Professeur, Université Pierre et Marie Curie, Paris
	Thierry Artieres	Professeur, Laboratoire d'Informatique Fondamentale, Marseille
Examinateurs :	Marc Sebban	Professeur, Laboratoire Hubert Curien, Saint-Etienne
	Olivier Buffet	Chargé de Recherche, INRIA, Nancy
Directeur de thèse :	Olivier Pietquin	Professeur, Université Lille 1, Lille

---

Supélec
équipe MaLIS

2 rue Edouard Belin
57070 Metz

Remerciements

Je tiens tout particulièrement à remercier mon directeur de thèse, Olivier Pietquin, pour avoir toujours su me donner les plus précieux conseils sans lesquels cette thèse n'aurait jamais pu aboutir. Je tiens également à complimenter son sens du relationnel qui a fait de cette thèse un moment agréable. Je souhaiterais aussi remercier Olivier Buffet, qui m'a toujours reçu avec enthousiasme et avec qui nous avons mené des discussions de qualité quant à l'avancement de ma thèse. Je remercie le campus CentraleSupélec de Metz pour m'avoir donné la possibilité d'effectuer cette thèse ainsi que l'ensemble de son personnel dont certains avec lesquels j'ai pu avoir des relations amicales. J'accorde également toute ma gratitude envers l'ensemble de l'équipe *Sequel* pour m'avoir si bien accueilli et intégré lors de mes déplacements. Je tiens aussi à saluer les doctorants et amis qui ont effectué tout ou partie de leur thèse en même temps que moi et qui resteront dans mon cœur : Hadrien, Denis, Guillaume, Bilal, Édouard, Lucie, Shane, Chakon. Une pensée va également à mes amis les plus chers qui ont toujours été là lorsqu'il le fallait. Enfin, j'aimerais remercier profondément mes parents et ma famille qui m'ont soutenus dans les moments difficiles de ma thèse et ont toujours su m'indiquer la voie de l'optimisme.

Table des matières

Chapitre 1

Introduction

1.1 Motivations

Actuellement, nous sommes en train d'assister à une explosion en matière de données. Autant vis-à-vis de la quantité et de l'aspect sophistiqué des techniques permettant de les exploiter, ce qui offre de nouvelles possibilités quant à leur utilisation, que du volume de données rendu disponible par l'amélioration des outils de stockage et de communication ainsi que par l'évolution de notre rapport aux données. Toutefois, de nombreuses méthodes ne peuvent supporter une telle charge de données, rien que pour que celles-ci puissent être accédées cela nécessite un certain temps, il devient donc nécessaire d'en effectuer une sélection et si possible de profiter de ce degré de liberté pour ne retenir que les plus importantes. Les travaux présentés dans ce manuscrit ont pour but de proposer une nouvelle solution pour le processus de sélection de données relatif à une technique spécifique : la classification.

Tout d'abord, la classification est un sous-problème de l'apprentissage supervisé, celui-ci est un cadre de travail général pour tout problème visant à prédire les liens intrinsèques existant entre des données d'entrée et des données de sortie. La nature des données mises en jeu peut être diverse : discrètes ou réelles, simples ou multivariées, structurées ou non. Ces liens, initialement inconnus, peuvent être révélés par un expert, ou oracle, en lui soumettant une donnée d'entrée à laquelle il attribue une donnée de sortie. Le problème consiste à apprendre, à partir d'un ensemble fini de données d'entrée soumises à l'oracle et de ses réponses respectives, une fonction de prédiction permettant d'attribuer à n'importe quelle nouvelle donnée d'entrée, une donnée de sortie sans avoir recours à l'oracle. Le but est alors que la prédiction établie corresponde au mieux à la donnée de sortie qu'aurait fourni l'oracle si on lui avait soumis la même donnée d'entrée. L'ensemble des données annotées servant à l'apprentissage est appelé base d'entrainement. Deux cas particuliers de l'apprentissage supervisé font l'objet d'une dénomination spécifique : la régression dans laquelle les données de sortie sont des valeurs réelles simples, et la classification que nous détaillons maintenant.

La classification est un cas particulier de l'apprentissage supervisé dans lequel les données de sorties possibles sont limitées à un ensemble discret et fini. Elles sont appelées étiquettes et n'ont aucune relation entre elles, la fonction de prédiction est quant à elle appelée classifieur. L'objectif visé est de catégoriser les données d'entrée en les séparant en différentes classes, chacune étant définie par l'association à une même étiquette. Contrairement à un apprentissage non supervisé de type k-moyennes, ici il existe une étiquette intrinsèque, qui peut être vérifiée via l'oracle. Dans le cas où il n'existe que deux étiquettes possibles, on parle de classification binaire, dans le cas contraire, on parle de classification multi-classes. Quelques fois, la réponse de l'oracle n'est pas

évidente et celui ci ne renvoie pas toujours la même étiquette lorsqu'une même donnée d'entrée lui est présentée, on parle alors de classification bruitée. On considère toutefois qu'il existe une distribution fixe intrinsèque des étiquettes renvoyées par l'oracle pouvant varier en fonction de la données présentée. Notez que pour minimiser l'erreur, le classifieur doit, lui, être déterministe. Nos travaux se limitent à la classification binaire et bruitée, dans ce cas, la distribution des étiquettes ne dépend que d'un seul paramètre : la proportion d'une classe par rapport à l'autre.

Dans le cas général de l'apprentissage supervisé, afin de constituer la base d'entrainement, l'ensemble des données ayant été collectées est habituellement soumis à l'oracle. En effet, plus le nombre de données contenues dans la base d'apprentissage est grand, meilleure est la qualité de la prédiction. Cependant, il existe des cas où l'appel à l'oracle est coûteux, que ce soit en termes de temps, d'argent ou toute autre pénalité. Afin de garantir un certain coût maximal, le nombre de données pouvant être incluses dans la base d'apprentissage peut être limité. Réduire simplement le nombre de données collectées impacterait grandement la performance de la fonction de prédiction. Toutefois, il est possible de collecter un nombre de données important mais de ne soumettre à l'oracle qu'une partie de ces données soigneusement sélectionnées. En effet, cette liberté peut permettre de ne choisir que les données les plus informatives, en s'abstenant des redondances. La sélection se fait en général de façon séquentielle, afin de pouvoir tenir compte de toute l'information apportée par l'oracle dans le choix de chaque nouvelle donnée à lui présenter.

Il existe deux utilisations majeures de cette sélection de données en fonction de l'emploi que l'on souhaite faire de la fonction de prédiction. La première est au centre de nos travaux et est appelée apprentissage actif, elle découle naturellement de la fonction principale de l'apprentissage supervisé, c'est à dire la prédiction. Elle vise à apprendre avec un minimum d'exemples, une fonction de prédiction ayant la même qualité quelle que soit la donnée d'entrée. De cette façon, lors de l'utilisation de cette fonction de prédiction sur des données d'entrée tirées aléatoirement ou bien suivant leur distribution naturelle, l'erreur moyenne engendrée sera minimum. Dans ce cas, une donnée peut être considérée informative si peu de données similaires ont déjà été annotées, les résultats des données similaires sont variés ou si de nombreuses données non annotées sont proches. Ce domaine a déjà été largement étudié, on distingue plusieurs approches pour résoudre ce problème. Parmi elles, une méthode est de sélectionner la donnée d'entrée pour laquelle le classifieur courant commet la plus grande erreur, étant donné que cette donnée apporte de l'information, l'erreur commise maximale est garantie de descendre. La mesure de l'erreur est habituellement le risque réel, c'est à dire l'espérance de l'erreur de classification. Cependant, le risque réel pouvant être difficile à utiliser, il est possible de se servir d'une mesure alternative comme la variance ou l'entropie de Shannon. Une autre méthode est de sélectionner la donnée d'entrée ayant le plus d'influence sur la performance globale du classifieur. Cela permet de prendre en compte la densité de données d'entrée dans la décision afin de favoriser la minimisation l'erreur relative à une donnée qui sera d'avantage sollicitée. Une autre vision du problème ayant donné lieu à de nombreux travaux est de considérer l'ensemble des classifieurs cohérents avec les données, appelé espace des versions, et de sélectionner la donnée d'entrée qui permet de rejeter le plus de classifieurs de cet ensemble. Ainsi, cela revient à effectuer une dichotomie sur l'espace des versions et permet de trouver en un minimum d'exemples le meilleur classifieur de l'ensemble.

Une seconde utilisation de la sélection de données est l'optimisation, ce qui se fait uniquement dans le cadre d'une régression. Ce problème consiste à trouver la donnée d'entrée associée à la valeur maximale. Pour cela, une fonction de prédiction est apprise à partir des données annotées et le maximum de cette fonction est utilisé comme estimateur du maximum réel. Dans ce contexte, la sélection de données a pour but d'approcher le maximum réel le plus rapidement possible, c'est à dire avec le moins d'évaluations ou encore le moins de requêtes à l'oracle. Dans le cas où aucune

hypothèse de convexité peut être effectuée, il peut exister plusieurs maxima locaux, de plus, les données peuvent être bruitées et donner de mauvais tirages. Ainsi, une zone dont l'estimation est peu précise peut potentiellement contenir le maximum. Le but de la sélection de données est alors d'une part d'explorer l'espace d'entrée afin de découvrir de nouveaux maxima potentiels, et d'autre part d'exploiter les maxima potentiels connus afin d'améliorer la précision sur l'emplacement exact du maximum réel. Un compromis doit être fait entre ces deux aspect, ce qui a donné lieu à de nombreux travaux sous le nom de dilemme entre exploration et exploitation.

Le dilemme entre exploration et exploitation a notamment été étudié dans le cadre des bandits à bras multiples. Ce problème considère une machine à sous composée de plusieurs bras, chacun représenté par une distribution de récompenses inconnue mais fixe. A chaque étape, un unique bras est choisi et une récompense tirée selon la distribution associée est obtenue. Le but est de définir la meilleure stratégie à adopter pour maximiser le gain de récompense. Ce problème est le cas le plus simple faisant intervenir le dilemme entre exploration et exploitation. En effet, il faut tirer le bras qui donne la meilleure récompense tout en recherchant un meilleur bras. Plusieurs algorithmes ont été développés afin de résoudre ce problème, parmi eux se trouve l'algorithme de la borne de confiance supérieure (UCB) introduisant le principe de l'optimisme face à l'incertitude. Cela consiste, à chaque étape, à construire pour chaque bras un intervalle de confiance contenant la valeur de la récompense espérée avec une forte probabilité, puis de sélectionner le bras pour lequel la borne supérieure de cet intervalle est la plus grande. En général, ces intervalles de confiance sont construits à partir d'inégalités de concentration ne tenant pas compte de la famille de distributions car généralement inconnue. Dans le cas où cette famille est connue, il est possible d'utiliser l'inférence Bayésienne pour construire des intervalles de confiance appelés intervalles de crédibilité Bayésiens, qui ont la particularité d'être les plus étroits possible. Récemment, des travaux ont montré que l'algorithme UCB adapté à des intervalles de crédibilité Bayésiens donnait des résultats optimaux. Notez que le cadre des bandits à bras multiples ne permet de traiter que des données d'entrée discrètes et indépendantes. Ceci peut être remédié par l'emploi des Processus Gaussiens, ceux-ci étant la généralisation de la distribution Gaussienne aux fonctions, et offrent un cadre Bayésien au modèle de régression linéaire standard.

La thèse que nous postulons ici est que le principe de l'optimisme face à l'incertitude peut être adapté aux problèmes d'apprentissage actif, en particulier pour la classification. En effet, tout d'abord nous montrons que le problème d'apprentissage actif fait intervenir le dilemme entre exploration et exploitation. En effet, plaçons nous en un instant précis du processus, dans lequel un certain nombre de données ont déjà été soumises à l'oracle et une fonction de prédiction a été apprise suivant les observations reçues. Dans ce cas, le choix de la prochaine donnée à fournir à l'oracle peut soit accomplir une augmentation directe de la performance en choisissant une donnée informative déterminée par l'estimation courante, soit envisager une augmentation sur le long terme en améliorant l'estimation courante, ce qui peut potentiellement révéler d'autres données informatives. Ensuite, le problème d'apprentissage actif peut être reformulé comme un problème d'optimisation, pour lequel l'optimisme face à l'incertitude est une solution qui a fait ses preuves. En effet, bien que l'on cherche à connaitre la fonction de prédiction avec précision sur tout l'espace d'entrée, la sélection de la prochaine donnée à soumettre à l'oracle se fait sur la base d'un critère d'informativité dont on cherche le maximum. Finalement, la classification apporte l'avantage de connaitre la famille de distribution utilisée. Par exemple, dans le cas de la classification binaire auquel nous nous intéressons ici, la distribution des étiquettes suit une loi de Bernoulli dont le paramètre varie en fonction de la donnée d'entrée considérée. La connaissance de cette distribution permet d'utiliser efficacement l'inférence Bayésienne pour définir une croyance sur la valeur du paramètre, et de se servir des intervalles de crédibilité Bayésiens permettant une utilisation optimale

de l'optimisme face à l'incertitude.

Certains travaux vont déjà dans ce sens, appliquant le principe de l'optimisme face à l'incertitude à l'estimation de la moyenne de plusieurs distributions avec une précision uniforme. Ceci peut être aussi vu comme de l'apprentissage actif sur un problème de régression par des fonctions constantes par morceau s'appuyant sur une partition fixe de l'espace d'entrée. Le problème est traité sous l'angle des bandits à bras multiples en définissant un nouvel objectif d'allocation optimale lié à l'incertitude sur chaque bras. Deux algorithmes sont proposés ayant pour but de réduire soit l'erreur de prédiction maximale sur les bras, soit la somme pondérée des erreurs de prédiction sur tous les bras. Le même auteur propose aussi un autre algorithme se basant sur une partition adaptative, c'est à dire qu'elle peut être redécoupée plus finement au fur et à mesure de l'acquisition de nouveaux échantillons. Ce qui permet de ne pas voir les possibilités de prédiction être limitées par une découpe initiale trop grossière. Ces travaux fournissent de bonnes perspectives quant à l'utilisation de l'optimisme face à l'incertitude pour l'apprentissage actif.

Ainsi, les enjeux de cette thèse sont doubles. Premièrement, il s'agit de montrer que le principe de l'optimisme face à l'incertitude convient tout autant au problème d'apprentissage actif dans le cas de la classification, et qu'il existe un réel intérêt à développer une méthode spécialement conçue pour ce dernier. Il pourrait paraitre efficace d'utiliser directement un algorithme d'apprentissage actif conçu pour la régression dans le cadre de la classification. En effet, une solution couramment employée pour résoudre un problème de classification binaire est d'effectuer une régression de la probabilité intrinsèque d'obtenir une des deux classes, puis de définir un seuil de prédiction pour chaque classe. Ainsi, sachant que plus l'estimation est bonne plus la prédiction est de qualité, on pourrait s'imaginer qu'un algorithme dont le but est d'obtenir la meilleure estimation de la probabilité possible implique aussi d'obtenir la meilleure prédiction possible. Or, étant donné que le budget d'annotations est limité, et que la qualité de la prédiction n'est pas une fonction linéaire de celle de l'estimation, alors les données les plus rentables à sélectionner peuvent être différentes selon l'objectif considéré. Notamment, on cherchera à ce que la précision de l'estimation soit d'autant meilleure que cette dernière est proche du seuil. D'autre part, étudier spécifiquement la classification donne l'avantage de pouvoir préciser la famille de distributions relative aux données de sorties. Ceci permet de construire des intervalles de crédibilité Bayésiens et de les utiliser en lieu et place des intervalles de confiances. L'intérêt de ceux-ci est d'être les plus étroits possible, et ainsi de minimiser les erreurs d'allocation. Deuxièmement, le but est d'élaborer une méthode générale d'apprentissage actif et de montrer comment celle-ci peut être adaptée à différents algorithmes populaires de classification. Notamment, les travaux mentionnés précédemment ne montraient comment employer le principe de l'optimisme face à l'incertitude en apprentissage actif que sur un problème impliquant une partition, fixe ou adaptative. Or, ceci revient à se ramener à des données d'entrée discrètes et complètement décorrélées, ce qui ne permet qu'une estimation grossière de la probabilité sous-jacente, ne pouvant pas considérer ses variations rapides sur l'espace d'entrée. Ainsi, les algorithmes résultants ne permettaient pas de concurrencer les méthodes d'apprentissage actif existantes. L'objectif est donc de construire un algorithme se basant sur un classifieur performant pouvant traiter des données d'entrée de façon continues. Idéalement, notre volonté est d'utiliser le principe de l'optimisme face à l'incertitude pour l'apprentissage d'un classifieur basé sur les moindres carrés régularisés à noyau, ou encore les Processus Gaussiens qui ajoutent la possibilité d'exploiter l'incertitude qu'ils renvoient sur leur estimation. Pour cela, nous procéderont étapes par étapes en adaptant ce principe sur des classifieurs de complexité progressives. La démarche suivie consiste à commencer par développer un algorithme dans des conditions parfaites, mais irréalistes, où les paramètres du problème sont entièrement contrôlés et où les comportements de l'algorithme sont facilement interprétables. Par la suite, des difficultés supplémentaires viendront s'ajouter unes

à unes de sorte à contrôler indépendamment leurs effets sur l'algorithme, en étant capable d'expliciter les approximations nécessaires, pour enfin aboutir au classifieur souhaité.

1.2 Résumé des contributions

Dans cette section, nous essayons d'apporter une vue d'ensemble sur nos travaux. Il s'agit d'expliquer la démarche envisagée pour résoudre le problème posé, celle-ci servant aussi de fil conducteur dans la structure du manuscrit. Nous présenterons également les moyens mis en œuvre pour répondre aux enjeux cités précédemment. En particulier, nous tenterons de mettre l'accent aussi clairement que possible sur ce qui relève de nos contributions.

Tout d'abord, rappelons que la finalité de nos travaux est de montrer, à travers une application à au moins un classifieur existant et performant, que le principe de l'optimisme face à l'incertitude se présente comme une solution efficace pour traiter le problème d'apprentissage actif. Cependant, la conception d'un tel algorithme à partir de zéro est une tâche difficile. En effet, les données sur lesquelles il s'applique peuvent être continues, en grande dimensions, avec des liens entre elles mal connus. Ainsi, les comportements observés provenant de ces algorithmes peuvent être difficiles à interpréter et leurs causes peuvent ne pas être comprises. D'autant plus, il n'est pas évident de savoir si les performances obtenues sont optimales ou non. Dans ce cas, il est compliqué d'identifier les défauts de l'algorithme lors de sa construction et pour ensuite tacher d'y remédier. Ainsi, plutôt que de s'acharner directement à l'adaptation de ce principe sur un classifieur performant, nous jugeons préférable de procéder étapes par étapes en s'intéressant en premier lieu à un classifieur simple dont on contrôle aisément les paramètres, quitte à ce qu'il soit peu performant voire inutilisable en pratique, pour ensuite le complexifier petit à petit en introduisant les difficultés unes par unes. De cette façon, chaque aspect du problème pourra être traitée indépendamment afin d'obtenir algorithme bien compris. De plus, ceci permettra de limiter les approximation en commençant par développer un algorithme exact sur le problème initial simple, et en introduisant les approximation uniquement lorsque cela est nécessaire.

Dans la suite, nous tacherons de détailler chaque étape de la démarche, chacune étant associée à son propre classifieur, ainsi que la façon dont elles s'articulent. Mais avant cela, il est important de signaler que chaque algorithme sera développé en suivant le même schéma. Nous nous attardons donc d'abord ici sur la présentation des caractéristiques communes de la conception des algorithmes correspondant à chaque classifieur. Tout d'abord, la méthodologie habituellement employée pour concevoir un algorithme optimiste consiste à commencer par caractériser la solution idéale, c'est à dire celle qui serait suivie si l'ensemble des paramètres du problème étaient connus. Ceci a pour but de savoir ce qui est visé pour tenter de s'en rapprocher, elle sert aussi de base à la définition d'une solution réaliste qui doit intégrer l'incertitude dont elle est sujette. Dans notre cas, celle-ci prend la forme d'un critère de sélection idéal. Nous verrons dans l'état de l'art que la plupart des méthodes d'apprentissage actif utilisant la même approche peuvent être vues comme différentes dérivations d'un même critère de sélection idéal, dans lesquels l'incertitude est gérée différemment. Notre volonté est donc de réutiliser le même critère de sélection idéal afin de proposer le principe de l'optimisme face à l'incertitude comme une nouvelle solution pour gérer l'incertitude dans le critère existant. Cependant, ce dernier doit quelques fois être adapté aux spécificités du problème ainsi qu'intégrer des aspects supplémentaires, comme par exemple une vision à long terme. Pour chacun des classifieurs étudiés, une de nos contributions va donc être de définir un critère de sélection idéal adapté. Plusieurs d'entre eux pourront être introduits selon l'approche d'apprentissage actif sous laquelle on désire se placer. Ensuite, il s'agit de préciser comment prendre en compte ce critère idéal dans le critère de sélection effectif, celui-ci n'ayant pas accès à la valeur des paramètres. Pour cela, les algorithmes optimistes passent habituellement par la construction d'intervalles de confiance autour du critère idéal, le principe étant alors de ne considérer que leur borne supérieure. Ici, nous pensons plus judicieux d'employer des intervalles de crédibilité Bayésiens. Nous montrons alors

comment dériver de tels intervalles pour le critère considéré. Ces intervalles sont construits à partir de la distribution *a posteriori* sur le critère, calculée à partir des observations reçues. La forme que celle-ci prend dépend du classifieur utilisé et des approximations nécessaires.

Pour résumer, les contributions majeures communes à chaque classifieur sont les suivantes :

- comment définir un critère de sélection idéal adapté à la classification,
- comment construire des intervalles de crédibilité Bayésiens sur ce critère.

D'autre part, chacun des algorithmes proposé est évalué empiriquement face à des algorithmes de l'état de l'art, en utilisant des bases de données provenant du monde réel lorsque c'est possible ou bien sur un problème construit. La plupart de ces contributions ont fait l'objet d'une publication. Dans la suite, nous détaillons brièvement les différentes étapes de la démarche.

1.2.1 Partition fixe

Pour la première étape de notre démarche, nous avons choisi d'étudier le cas d'une partition fixe de l'espace d'entrée. Cette étude préliminaire, bien que trop simple et peu performante, permet de bien comprendre les phénomènes intervenant en apprentissage actif et de développer les solutions qui seront réutilisées par la suite, le tout au sein d'un environnement contrôlé.

Dans celui-ci, l'espace d'entrée est découpé en plusieurs zones distinctes et indépendantes. Toutes les données provenant d'une même zone étant traitées de la même manière et l'algorithme étant complètement aveugle à la provenance des données à l'intérieur de la zone. La sélection des données devra donc se limiter à choisir une zone dans laquelle tirer une donnée aléatoirement. Cela revient à considérer plusieurs sacs contenant des étiquettes dans lesquels on peut piocher pour en obtenir une et à vouloir étiqueter ces sacs selon l'étiquette majoritaire qu'ils contiennent. Chaque zone n'est alors caractérisée que par sa proportion d'étiquettes, celle-ci étant par conséquent associée à une distribution de Bernoulli. L'indépendance des zones implique que cette proportion ne peut être estimée qu'à l'aide des étiquettes provenant de la zone considérée et non des zones voisines. Cela nous ramène donc à un problème extrêmement simple caractérisé par un nombre fini de paramètres.

Ceci permet de mettre en exergue la présence du dilemme entre exploration et exploitation au sein du problème d'apprentissage actif avec un fort impact sur les performances obtenues. En effet, l'indépendance des zones fait en sorte que celles qui sont ignorées par la sélection n'apprennent jamais leur paramètre, ce qui résulte en de faibles performances de classification. Un autre avantage de l'indépendance des zones est de permettre de travailler avec une connaissance absolue des familles de distribution mises en jeu, ce qui est particulièrement utile lors de l'inférence Bayésienne.

En outre, ce problème peut être vu comme un bandit à bras multiple, cadre servant régulièrement à l'étude du dilemme entre exploration et exploitation ainsi qu'au méthodes optimistes. C'est notamment ce problème qui est considéré dans [13] et [14] qui traitent l'utilisation de méthodes optimisme pour l'apprentissage actif dans le cadre de la régression. Ceci nous permet donc de définir un point de départ pour la création de notre algorithme en adaptant simplement ces derniers au problème de classification. Quatre algorithmes sont proposés pour ce problème, ceux-ci se divisent en deux catégories.

Indépendants aux observations

Dans un premier temps, nous reprenons exactement le schéma proposé dans [13] et [14] en l'adaptant à la classification. Ici, le critère idéal, pour qui les paramètres du problème sont connus,

définit une allocation du budget indépendamment des observations reçues. L'objectif qui lui est donné est de minimiser l'erreur de prédiction espérée sur les tirages. Deux algorithmes sont proposés, l'un minimise l'erreur espérée maximale sur les zones, l'autre minimise la somme des erreurs espérées, pondérée par la densité des données, ce qui revient à minimiser l'erreur espérée pour une donnée d'entrée tirée aléatoirement selon sa distribution. Ces algorithmes ont fait l'objet d'une publication :

Timothé Collet and Olivier Pietquin. Active learning for classification : An optimistic approach. In *Proc. of ADPRL*, 2014.

Dépendants aux observations

Dans ce nouveau contexte, le critère idéal est désormais autorisé à dépendre des observations reçues. Ceci permet à la stratégie de sélection de prendre une décision différente selon que la bonne étiquette soit déjà attribuée par le classifieur courant ou non. Deux algorithmes sont encore proposés minimisant l'erreur maximale ou la somme pondérée des erreurs de chaque zone. Ces algorithmes ont fait l'objet d'une publication :

Timothé Collet and Olivier Pietquin. Optimism in Active Learning. In *Computational Intelligence and Neuroscience*, 2015

1.2.2 Autres classifieurs

Partition adaptative

La partition fixe n'était pas autorisée à changer au cours du processus. Cela était particulièrement limitant, en voyant sa performance maximale diminuer si le nombre de zones est trop faible, et voyant sa vitesse d'apprentissage limitée si il est trop important. Ici, cette contrainte est relâchée. Cela doit cependant suivre certaines règles. En effet, afin que l'estimation du paramètre dans chaque zone soit correcte, il est nécessaire que les données étiquetées y soient reparties selon leur distribution naturelle. Cela implique de ne pas pouvoir changer les frontières des zones ni d'en supprimer. Il est toutefois possible de redécouper une zone pour en obtenir deux plus petites. Les arbres incrémentaux fournisse l'algorithme permettant de guider la découpe des zones dans ce contexte. Nous proposons donc un critère de sélection optimiste pour ce problème, celui-ci n'étant que très légèrement modifié par rapport à celui conçu pour une partition fixe. Nous proposons également un nouvel algorithme d'arbre incrémental spécialement conçu pour la classification, ré-utilisant les concepts manipulés dans les méthodes précédentes, notamment les intervalles de crédibilité Bayésiens. Cet algorithme a fait l'objet d'une publication :

Timothé Collet and Olivier Pietquin. Bayesian Credible Intervals for Online and Active Learning of Classification Trees. In *Proc. of ADPRL*, 2015.

Partition fixe généralisant

Plutôt que d'autoriser la partition à évoluer, une autre direction que nous pouvons emprunter pour complexifier le classifieur est d'autoriser les zones à partager de l'information. La vitesse d'apprentissage dans le cas des partitions fixes est limitée car il faut apprendre les paramètres de chaque zone indépendamment. Ceci permet donc de ne pas contraindre la vitesse d'apprentissage par le nombre de zones. Ici, le classifieur se sert de toutes les observations reçues quelle que soit leur zone de provenance, pondérées par la proximité de ladite zone, pour estimer le paramètre de chaque zone et ensuite définir les prédictions effectuées. Il est alors nécessaire de considérer dans

le critère de sélection l'influence d'un tirage dans une zone sur l'ensemble des prédictions dans les autres zones pour évaluer son impact sur la performance globale. La difficulté est aussi que la distribution de l'étiquette prédite en fonction des tirages dépend d'un grand nombre de paramètres. Un algorithme est proposé, en approximant cette distribution par une distribution Gaussienne. Cet algorithme a fait l'objet d'une publication :

Timothé Collet and Olivier Pietquin. Optimism in Active Learning. In *Computational Intelligence and Neuroscience*, 2015

Processus Gaussien

En poussant la discrétisation de l'espace d'entrée à son maximum dans le problème précédent, il est possible de se ramener à un traitement continu des données d'entrée. Nous remarquons alors que l'ensemble des taux de combinaison de chaque paire de zones se rapprochent de la matrice de covariance utilisée dans les méthodes à noyau. L'étape suivante est alors d'utiliser directement une méthode à noyau, à savoir les Processus Gaussiens ce qui peut être vu comme l'approche Bayésienne équivalente de la méthode des moindres carrés régularisés à noyau. Notez que qu'une différence existe toutefois entre ces deux étapes. Précédemment, les taux de combinaisons n'étaient pas autorisés à changer au cours du processus, au risque de bouleverser l'allocation du budget déjà établie dans la partition. Ici, les poids sont recalculés à chaque pas de temps. Nous nous posons la question de savoir si le principe de l'optimisme face à l'incertitude peut toujours être utilisé dans ce cas. Nous proposons un algorithme d'apprentissage actif pour les Processus Gaussiens. Cet algorithme a fait l'objet d'une publication :

Timothé Collet and Olivier Pietquin. Optimism in Active Learning with Gaussian Processes. In *Proc. of ICONIP*, 2015.

Première partie

État de l’art

Chapitre 2

Position du problème d'Apprentissage Actif

2.1 La Classification

Avant d'entrer dans le vif du sujet, il est nécessaire de commencer par situer le contexte dans lequel cette thèse est mise en œuvre ainsi que de présenter les différents concepts sur lesquels elle s'appuie. Cette section ainsi que la suivante vont ainsi servir à introduire les bases de la classification et de l'apprentissage actif. Cette description restera cependant très générale en laissant pour l'instant de côté l'usage de notations formelles. Ce sera l'occasion pour le lecteur novice de se familiariser avec les termes que nous supposerons connus par la suite ainsi que d'appréhender le point de vue depuis lequel les différents problèmes du domaine sont traités et d'avoir une idée des solutions existantes.

Lorsque l'on travaille avec des données, il est souvent envisagé de leur attribuer une étiquette. En effet, cela peut être utilisé pour synthétiser leur description, permettant par exemple de leur appliquer un traitement différent en fonction de leur étiquette, mais aussi de faciliter la compréhension que l'on a de la base de données ou encore d'en accélérer l'accès dans un espace de stockage. Un exemple bien connu est celui de la catégorisation des médias dans une bibliothèque à l'aide de mots clés, ce qui permet d'effectuer des recherches ainsi que des recommandations de manière plus efficace. D'autre part, cela peut aussi être utilisé pour faire ressortir un attribut en particulier dans le but de conditionner une décision. Par exemple pour identifier et marquer les produits défectueux sur une chaine de montage pour qu'ils soient ensuite rejetés. Précisons que le fait de donner une même étiquette à plusieurs données séparément est équivalent à les regrouper au sein d'une même catégorie, ou classe. De façon formelle, une étiquette se définit par un élément provenant d'un ensemble fini et dénombrable.

Un parallèle peut être effectué avec la tendance naturelle des êtres humains à nommer les choses à travers le langage. Ainsi, un nom regroupera un ensemble d'objets pouvant prendre des formes bien différentes. Par exemple, une table peut être carrée, ronde, avoir quatre pieds, six pieds, être en bois ou en métal, mais l'ensemble de ces choses sont regroupées sous le même nom. En philosophie, deux courants de pensée tentent d'en expliquer le fonctionnement. Le relativisme de Protagoras exprime que *L'homme est la mesure de toutes choses*. Cette notion implique que l'homme attribue lui-même un même nom à plusieurs objets parce qu'ils sont selon lui semblables ou différents de certains autres. Au contraire, Platon introduit quant à lui la notion de *monde des idées* dans laquelle toute chose est la projection d'une idée parfaite. Le nom préexiste donc à l'homme et l'apprentissage consiste à le révéler.

Bien que le domaine de l'apprentissage automatique soit complètement indépendant des études philosophiques, on retrouve ces deux aspects dans la définition des deux types d'apprentissages existants et pouvant être associées au problème d'étiquetage des données. Ainsi, on distingue l'apprentissage supervisé de l'apprentissage non-supervisé. Commençons par décrire le second cas. Dans celui-ci, le choix des étiquettes possibles est laissé libre à l'algorithme, ceci incluant leur nombre et leur dénomination. L'objectif est alors de créer différentes classes contenant des données les plus similaires possibles. Ceci se basant uniquement sur la description des données faite initialement. Par exemple, nous pouvons imaginer disposer d'un échantillon de sable et que nous souhaitions trier les grains pour les placer dans différents bocaux en fonction de leur taille : petits, moyens ou grands. Il n'y a pas ici d'*a priori* sur la définition d'un petit grain de sable, la taille étant déterminée uniquement relativement aux autres grains. Il est alors nécessaire que l'algorithme définisse lui-même la taille approximative des grains de chaque classe de sorte à remplir les bocaux de manière équitable. Un autre problème pourrait être que le sable soit composé de grains provenant de trois constituants inconnus. On souhaite mieux connaitre la composition du sable, et donc déterminer la nature et la proportion de chaque constituant. Sachant que la taille moyenne des grains de sable diffère selon le constituant, il est envisagé de séparer les grains de sable en trois classes en fonction de leur taille. Notez que les proportions de chaque constituant peuvent être différentes. Ainsi, il ne faut plus séparer les grains en classes équitables mais étudier la densité/fréquence de chaque taille de grain pour en déduire les classes, les tailles les moins fréquentes se trouvant probablement à la frontière entre deux classes. A titre d'information, une des méthodes les plus classiques d'apprentissage non-supervisé est la méthode des k-moyennes. Celle-ci nécessite de définir par avance le nombre de classes. Elle consiste à attribuer initialement les étiquettes au hasard, puis effectuer plusieurs itérations en alternant le calcul de la description moyenne des données chaque classe et la ré-attribution des étiquettes en fonction de la description moyenne la plus proche.

L'autre contexte d'attribution d'étiquettes, la classification, se place dans le cadre de l'apprentissage supervisé. Ici, contrairement au cas précédent, il existe une vraie étiquette intrinsèquement associée à chacune des données. Seulement, celles-ci ne sont pas connues, on souhaite donc être capable de prédire leur valeur. Pour cela, il est possible de se baser sur des exemples, prenant la forme d'un ensemble limité de données dont l'étiquette a été révélée par un expert, ou oracle. Cet ensemble porte le nom de base d'apprentissage. Le but va alors être de créer une fonction, appelée classifieur, permettant de prédire une étiquette pour toute donnée qui lui est présentée. L'oracle peut aussi être vu comme la fonction de prédiction parfaite. L'algorithme tente donc de créer un classifieur se rapprochant le plus possible de la fonction de l'oracle. Notez qu'il sera demandé à l'algorithme d'effectuer une prédiction pour des données qui ne font pas partie de la base d'apprentissage et donc qui n'ont pas vu leur étiquette révélée, celui-ci devra donc généraliser à partir des étiquettes de données similaires incluses dans la base d'apprentissage. Précisons enfin que l'apprentissage supervisé n'est pas réservé à l'emploi d'étiquettes mais est défini plus largement pour tout problème faisant intervenir une vérité intrinsèque devant être prédite/reproduite à l'aide d'une une connaissance partielle provenant d'un oracle. Celle-ci peut aussi se présenter sous la forme de valeurs réelles, auquel cas on parle de régression, de données structurée, ou autre. La classification est le nom donné au cas particulier de l'apprentissage supervisé qui travaille avec des étiquettes.

Servons nous d'un exemple afin d'illustrer le problème de classification. Imaginons qu'un fabriquant d'appareil photo souhaite inclure une fonctionnalité permettant de déclencher l'appareil lorsqu'une personne sourit. Il faut alors qu'il soit capable d'identifier si un visage sourit ou non. Ici, la notion de sourire est relative à l'être humain et l'algorithme doit s'y contraindre. De son point de vue, c'est une vérité qu'il doit apprendre. Notez bien qu'ici, on ne s'intéresse pas au degré

de sourire, même si il existe, le but est uniquement de déclencher l'appareil ou non. Pour cela, le fabriquant dispose d'un grand nombre d'images de visage collectées sur le terrain. Cet ensemble d'images est ensuite soumis à un oracle, ici un annotateur humain, qui attribue l'étiquette "sourit" ou "ne sourit pas" à chacun des visages. L'algorithme va donc utiliser cet ensemble pour calculer une fonction de prédiction qui sera inclue dans l'appareil photo et qui sera capable pour chaque nouveau visage rencontré de déterminer si il sourit ou non. Remarquez qu'il y a peu de chances que le visage détecté lors de l'utilisation de l'appareil soit strictement identique à l'un de ceux qui a déjà été annoté. Cependant, la similarité entre ces images peut être utilisée, notamment en fonction de caractéristiques telles que l'écartement des lèvres, la position des joues, etc.

Le cas qui nous intéresse ici et que nous traiterons dans nos travaux est celui de l'apprentissage supervisé et donc du problème de classification. Nous nous permettons donc de nous attarder sur celui-ci afin de décrire les différentes caractéristiques que peut avoir un tel problème. Tout d'abord, on distingue deux types de problèmes en fonction du nombre de valeurs que peuvent prendre les étiquettes. Lorsque celles-ci ne peuvent prendre que deux valeurs possibles, on parle de classification binaire. Dans le cas contraire, on parle de classification multi-classe. Notez qu'une méthode permet d'appliquer n'importe quelle solution d'un probème de classification binaire sur un problème de classification multi-classe, il s'agit de la méthode du "un-contre-tous". Celle-ci consiste à entrainer un nombre de classifieurs binaires égal au nombre de classes du problème, chacun d'entre eux traitant une classe en particulier et considérant toutes les autres comme une seule et même classe. Ensuite les décisions provenant de chacun de ces classifieurs sont combinés pour former la décision finale sur l'étiquette à prédire. Dans nos travaux, nous nous limitons à l'étude de la classification binaire.

Une autre caractéristique que peut présenter un problème de classification est l'aspect bruité. Dans ce cas, si l'on soumet plusieurs fois la même donnée à l'oracle, celui-ci ne donne pas toujours le même résultat. Les causes qui peuvent provoquer l'aspect bruité d'un problème sont diverses. Tout d'abord, les données peuvent être naturellement bruitées. Dans l'exemple des grains de sable précédents, il existe une variabilité sur la taille des grain d'un constituant donné, et il se peut alors que deux grains ayant une même taille proviennent de deux constituants différents. Un bruit de mesure sur l'acquisition des données donnerait le même résultat. Ensuite, l'oracle peut ne pas être parfait ou il peut exister un bruit sur l'étiquette reçue. Par exemple, si les données sont annotées par un groupe d'experts humains et que leurs avis divergent, alors une même donnée pourrait mener à des étiquettes différentes selon la personne effectuant l'annotation. Une autre cause peut être le manque de représentativité, par exemple si l'oracle a accès à un nombre supérieur de paramètres décrivant la donnée que le classifieur. Dans l'exemple des images de visages précédent, seules quelques caractéristiques extraites, comme les proportions du visage, de la bouche et des yeux peuvent être données au classifieur alors que l'annotateur voit réellement l'image avec l'ensemble de ses pixels. Ainsi, du point de vue du classifieur, plusieurs données lui paraissant identiques car ayant les mêmes caractéristiques, peuvent être étiquetées différemment. Dans un problème de classification bruitée, on suppose qu'il existe une distribution de probabilité fixe sur l'étiquette qu'il renvoie. Celle-ci n'est bien sûr pas connue du classifieur. Le but va alors être de construire un classifieur qui permette de prédire l'étiquette que renverrait l'oracle le plus souvent possible. Notez qu'il est inutile de faire intervenir un aléa dans la fonction de prédiction des étiquettes. En effet, il est impossible de savoir à quel moment l'oracle renverra l'une ou l'autre étiquette. Ainsi, la meilleure stratégie est de toujours prédire l'étiquette qui a le plus de chances d'être renvoyée par l'oracle. Dans un problème de classification bruité, la difficulté est qu'il ne faut pas se fier entièrement aux étiquettes reçues. En effet, celles-ci peuvent correspondre à l'annotation la moins fréquente par l'oracle. Il est donc nécessaire d'inclure une sorte de prudence. Le but d'un classifieur est d'estimer

les distributions pour pouvoir ensuite déterminer l'étiquette la plus probable et l'utiliser comme prédiction. Les étiquettes reçues peuvent être vues comme des observations des ces distribution. Notez que puisque les étiquettes sont en nombre fini, et dénombrables, les distributions relatives à chaque donnée ne dépendent que d'un certain nombre de paramètres. Par exemple, dans le cas de la classification binaire, un seul paramètre intervient, il s'agit de la probabilité de révéler une des deux étiquettes.

Parmi les méthodes de classification les plus standards, nous trouvons : la méthode des k plus proches voisins, qui affecte l'étiquette majoritaire parmi les k données annotées les plus proches, les machines à vecteur support, qui tentent de trouver le séparateur entre deux classes qui passe le plus loin des données annotées, les arbres de décision, qui divisent hiérarchiquement l'espace de sorte à obtenir des zones homogènes vis-à-vis des étiquettes, puis qui basent leur prédiction sur l'étiquette majoritaire dans chaque zone, les forêts aléatoires, qui font intervenir plusieurs arbres de décisions avec un paramètre aléatoire, puis combinent l'ensemble des prédictions obtenues dans une prédiction finale, la régression logistique, qui effectue une régression des paramètres de l'oracle avec une fonction objectif basée sur la vraisemblance. Nous pouvons aussi constater certains cas de classification binaire employant les moindres carrés pour régresser les paramètres, c'est à dire en minimisant l'erreur quadratique entre les paramètres estimés et les étiquettes révélées, ceux-ci étant ensuite seuillés entre 0 et 1 pour convenir à une probabilité, pour ensuite baser la prédiction sur l'arrondi du paramètre estimé.

Nous pouvons distinguer une classe de classifieur appelés classifieurs probabilistes. Ces classifieurs, plutôt que de prédire directement une étiquette, délivrent un score dont la valeur indique la tendance à appartenir à chaque classe. L'étiquette est alors déterminée en appliquant un seuil sur ce score. Bien qu'à l'origine, ces classifieurs aient pour but d'estimer la distribution probabilité sur l'étiquette renvoyée par l'oracle, comme c'est la cas pour la Régression Logistique ou les Moindres Carrés, la fonction de score en est le plus souvent déformée. Certain classifieurs n'étant pas naturellement probabilistes peuvent toutefois être adaptés pour délivrer un score. Par exemple, ce score peut être, dans l'algorithme des k plus proches voisins, la proportion d'étiquettes d'une classe parmi les k voisins sélectionnés, ou encore pour les SVMs, la distance à la frontière.

En plus des deux types d'apprentissage que l'on vient de voir, il existe celui de l'apprentissage semi-supervisé qui se place à mi-chemin entre l'apprentissage supervisé et non-supervisé. Celui-ci reprend la même situation que l'apprentissage supervisé, à savoir qu'il existe une vraie étiquette associée à chaque donnée et que l'algorithme doit se contenter de les prédire. Cependant, des données non-étiquetées viennent s'ajouter à la base d'apprentissage. Bien que ces dernières n'apportent pas directement d'information supplémentaire quant à la valeur de la fonction de l'oracle, elle permettent d'obtenir une meilleure représentation de la répartition des données et ainsi d'améliorer les performances de classification. En effet, la forme que prennent les données peut fournir un *a priori* sur la façon dont les différentes classes sont reparties entre celles-ci. Par exemple, si l'on observe que les données sont toutes regroupées autour de deux centres distincts, il y a de forte chances que l'ensemble des données regroupées autour du même centre appartiennent à la même classe. L'idée est donc d'intégrer dans la fonction objectif des coûts provenant de techniques d'apprentissage non-supervisé et ainsi de contraindre la prédiction du classifieur pour la faire tendre vers un meilleur corrélation entre les classes et les groupes de données. Assez grossièrement on peut imaginer une méthode qui définirait dans un premier temps des groupes de données à l'aide d'une méthode non-supervisée sur l'ensemble des données disponibles avant de déterminer l'étiquette à prédire pour chaque groupe uniquement en fonction des données annotées dans chacun d'eux. Ce type de problèmes est particulièrement utile lorsqu'il y a très peu de données étiquetées ou que celles-ci ne suivent pas la même répartition que les données brutes. Par exemple, dans le cas des

images de visages, si un annotateur humain ne peut pas déterminer l'étiquette d'un visage qui porte une expression neutre et doive rejeter l'image.

Pour finir, évoquons le fait qu'il existe principalement deux modes d'acquisition des données avec lequel s'effectue l'apprentissage supervisé ou semi-supervisé. Le premier est le cas hors-ligne, dans lequel l'ensemble des données étiquetées sont obtenues simultanément. Le classifieur est alors calculé une seule fois en prenant en compte toutes celles-ci en même temps. Le second est le cas en-ligne où, au contraire, les données étiquetées sont obtenues les unes après les autres. La prédiction du classifieur est toutefois requise à tout instant, de sorte que le classifieur doit être recalculé après chaque réception d'une nouvelle étiquette afin de prendre en compte le maximum d'information disponible, et ainsi d'obtenir les meilleures performances à chaque instant. Notez que le fait de recalculer entièrement le classifieur peut être extrêmement couteux en temps de calcul. Sachant que la plupart des étiquettes ont déjà été prises en compte lors de l'étape précédente, certains algorithmes ont été développés afin de permettre au classifieur d'être simplement mis-à-jour en considérant uniquement la donnée étiquetée nouvellement acquise.

Dans le cas en-ligne, serait-il possible que la prédiction du classifieur à chaque instant conditionne la prochaine donnée qui sera soumise à l'oracle ? Cela pourrait permettre de ne recevoir que des données utiles à l'apprentissage du classifieur et ainsi améliorer sa vitesse d'apprentissage. Ceci est étudié dans le domaine de l'apprentissage actif qui va faire l'objet de la prochaine section.

2.2 L'Apprentissage Actif

Dans la section précédente, nous avons vu qu'un problème d'apprentissage supervisé nécessitait l'utilisation d'une base d'entrainement contenant des données d'entrée associées à des données de sortie ; ces dernières étant, dans le cas de la classification, des étiquettes. Pour obtenir ces associations les étiquettes doivent être révélées par un oracle. Habituellement, l'ensemble des données collectées est soumis à l'oracle pour en tirer le maximum d'information et ainsi obtenir la meilleure prédiction. Cependant, il existe des situations où il est impossible ou dans lesquelles on ne désire pas annoter toutes les données. Dans ce cas, il y a différentes possibilités pour les données qui seront soumises à l'oracle. Or, toutes les données ne sont pas équivalentes, certaines données présentant un meilleur intérêt pour l'apprentissage du classifieur. Cette liberté sur le choix des données peut donc être mise au profit de l'obtention de meilleures performances pour le classifieur. On appelle apprentissage actif le processus dans lequel les données d'entrées qui serviront à l'apprentissage sont autorisées à être choisies. Il s'agit alors de définir une stratégie de sélection des données à adopter dans le but de maximiser la vitesse d'apprentissage. De manière équivalente, cela revient à maximiser les performances sous un nombre de requête à l'oracle fixe ou bien à minimiser le nombre de requêtes permettant d'atteindre une performance donnée. Dans la suite, nous présentons les motivations d'un tel problème ainsi qu'une description générale de sa mise en œuvre, et détaillons les différents scénarios possible.

De nos jours, collecter des données à des fins d'annotation est devenu extrêmement aisé. En effet, le développement des outils matériels et logiciel, ainsi que des techniques de calcul ont rendu possible l'acquisition, le stockage et le traitement d'un grand volume de données. De plus, le rapport du grand public aux données à fortement évolué, de sorte qu'une importante quantité de données soit disponible et facilement accessible. Cependant, l'annotation proprement dite de ces données requiert la sollicitation de l'oracle. Ceci étant, le recours à l'oracle peut quelques fois être limité ou bien pénaliser l'apprentissage. En effet, premièrement, ce dernier peut posséder des capacités de traitement de l'information restreintes ou voir son accès soit limité, de sorte qu'un certain temps soit nécessaire pour obtenir l'étiquette d'une donnée. Celui-ci, dépendant du nombre de données d'entrée annotées, s'ajoute au temps de calcul nécessaire pour l'apprentissage et peut même le surpasser significativement. Ainsi, pour que l'ensemble du dispositif d'apprentissage se fasse dans un temps raisonnable, il peut être nécessaire de limiter le nombre de fois que l'on fait appel à l'oracle. Ensuite, outre la durée d'apprentissage, d'autres pénalités peuvent aussi accompagner l'annotation des données. Par exemple, lors d'un diagnostic médical, où on cherche a prédire une affection à l'aide de symptômes, obtenir la vérité terrain peut nécessiter d'effectuer des examen médicaux extrêmement coûteux au niveau financier voire associés à des problèmes éthiques. De même, lors d'une analyse des matériaux, l'échantillon annoté peut se voir détruit lors du processus d'analyse, ce qui peut être indésirable en fonction de la rareté de l'échantillon (*ex :* archéologie). On cherche alors à limiter le nombre de donnée annotées.

Le champs d'applications de l'apprentissage actif est extrêmement large. Nous venons de citer brièvement quelques exemples qui permettent une première approche du domaine. Cependant, il est à noter que la définition est beaucoup plus générale. Par exemple, le concept d'oracle, peut prendre de nombreuses formes, et, bien que l'emploi du terme d'*annotation* le laisse entendre, ne se limite pas à l'usage d'un expert humain. Nous allons poursuivre en présentant de façon plus détaillée certains exemples intéressants de par leur diversité, leur non-conformité et car ils correspondent à une application réelle. De cette façon, cela nous permettra d'apprendre à reconnaitre les cas pouvant profiter de l'apprentissage actif ainsi que d'entrevoir l'étendue des applications possibles.

- **Sondage d'opinion :** L'oracle ne se présente pas obligatoirement comme un spécialiste,

mais détient la vérité terrain. Par exemple, dans [14], les auteurs mentionnent le cas d'un sondage d'opinion (on peut s'imaginer qu'il soit réalisé sur internet) où le but est de récolter l'avis des utilisateurs à propos de différents produit (au sens général, cela peut être aussi bien une idée politique, un produit marketing, etc.) à travers une fonction de score. Le données d'entrée caractérisent ici différents produits. Il s'agit alors de prédire les réponses des utilisateurs. Chaque utilisateur participant se voit alors proposé différent produits à évaluer. Ici, le coût d'annotation n'est pas très élevé pour l'utilisateur, que ce soit en temps ou en argent. Cependant, celui-ci n'a pas une patience infinie et peut décider d'arrêter à tout moment. La quantité d'annotations n'est pas connue par avance et il est donc nécessaire de lui fournir les données les plus pertinentes en premier.

- **Cartographie de séisme en temps réel :** Lorsqu'un séisme se produit, il est quelques fois envisagé de dresser une carte de l'intensité de celui-ci. En effet, cela peut servir à connaitre l'étendue des dégâts pour aller secourir les zones les plus touchées, ou bien à étudier sa propagation à des fins géologiques. Le réseau de capteur existant fournit des données éloignées d'une dizaines de kilomètres, rendant la carte peu précise. Dans [16], les auteurs envisagent d'utiliser un réseau de capteurs peu couteux mais de moins bonne qualité, comme les accéléromètres déjà présent dans les smart-phones de nombreux utilisateurs répartis sur la planète. Cela donne donc accès à une quantité importante de données. De plus, celles-ci sont déjà annotées puisque chaque smart-phone relie directement son accélération à sa position. Seulement, au vu de la quantité de données, le temps de communication nécessaire pour y accéder est limitant. Ainsi, pour que la cartographie puisse s'effectuer en temps réel, il est nécessaire de sélectionner uniquement une sous-partie des appareils à interroger.

- **Robotique :** Avant toute chose, il est nécessaire de faire la distinction entre prédiction et contrôle. Dans un cas, le but est d'étudier des données disponibles de sorte à être capable de les comprendre et de prédire un résultat pour des données suivant la même distribution. Dans l'autre, le but est toujours d'être capable de prendre un décision relativement à des paramètres donnés, mais la nature des situations qui seront rencontrées dans le futur est inconnue. Ainsi, dans un cas la prédiction doit prendre en compte la distribution des données tandis que dans l'autre cela est impossible. Un exemple de problème de contrôle est celui de la robotique. Dans celui-ci, les effets des différentes actions sont apprises à travers l'expérience en effectuant des simulations et en acquérant un retour de l'environnement. Par exemple, [19] décrit le cas d'un bras robotique composé de plusieurs articulations dont il est possible de commander les angles. Le but est d'apprendre les coordonnées absolues de l'extrémité du bras en fonction des angles affectés, afin de pouvoir définir la commande à fournir pour positionner l'extrémité à l'emplacement désiré. Pour cela, plusieurs positions sont testée et les coordonnées relevées. Ainsi, l'oracle peut ici être associé à ce retour de l'environnement, il n'y a donc pas d'entité extérieure à solliciter. Pour apprendre efficacement, il est nécessaire de définir une stratégie d'exploration de l'environnement qui peut se traduire par un dialogue avec l'oracle. Chaque simulation devant faire intervenir le corps matériel du robot, les contraintes mécaniques imposent de prendre un certain temps. Pour que l'apprentissage se fasse rapidement, le nombre de simulations doit être limité.

Nous voyons donc que, que ce soit en termes de temps, d'argents ou tout autre pénalité, les motivations pour réduire le nombre de requêtes à l'oracle sont nombreuses. Pourtant, réduire simplement le nombre de données présentées à l'oracle, en conservant la même distribution, par exemple en sous-échantillonnant le jeu de données disponibles, ne permet pas d'atteindre la performance

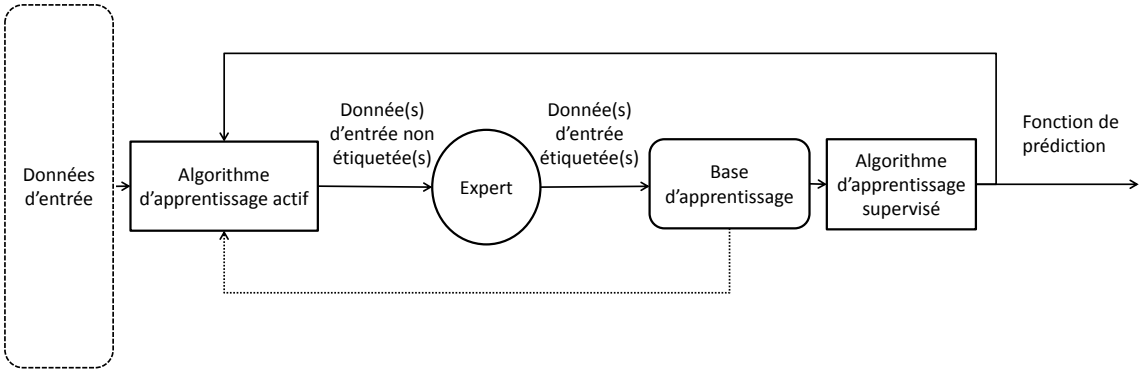


FIGURE 2.1 – Schéma synthétique de l'apprentissage actif

prédiction optimale. En effet, parmi l'ensemble des données annotées, il est possible que certaines d'entre elles soient redondantes et n'apportent pas d'information supplémentaire, si bien qu'il aurait été possible de s'en passer tout en obtenant les mêmes performances. L'idée est alors de sélectionner uniquement les données les plus informatives avant de les soumettre à l'oracle. Notez que cela revient à modifier la distribution des données d'entrée incluses dans la base d'apprentissage et de trouver la distribution adéquate permettant de maximiser les performances. Notez enfin que ce problème n'a pas la prétention d'obtenir de meilleures performances en collectant moins de données, mais de tirer parti du faible coût de collection pour s'autoriser un choix sur celles qui seront annotées.

Le processus d'apprentissage actif consiste à interagir directement avec l'oracle à l'aide d'un feedback provenant de la fonction de prédiction. Celui-ci est illustré sur la Figure 2.1. Tour à tour, l'algorithme d'apprentissage actif choisi une ou plusieurs données à soumettre à l'oracle parmi un ensemble de données possibles. Pour cela il établit un critère d'informativité associé à chaque donnée en se basant sur l'état courant de la fonction de prédiction tout en ayant connaissance des données contenues dans la base d'apprentissage. Ensuite, l'oracle renvoie une étiquette pour chaque donnée d'entrée qui lui est présentée, et les inclus dans la base d'apprentissage. Enfin, l'algorithme d'apprentissage supervisé met à jour la fonction de prédiction en prenant en compte la nouvelle base d'apprentissage.

Il existe plusieurs scénarios possibles pour l'acquisition des données d'entrée, et sur la manière dont celles-ci sont traitées par l'algorithme d'apprentissage actif. En se référant à la figure 2.1, cela consiste à définir de quoi est composé le bloc *Données d'entrée*. Nous détaillons ici les trois scénarios les plus représentés dans la littérature qui sont : la synthèse de requêtes, l'échantillonnage sélectif sur un flux, et l'échantillonnage dans un réservoir.

- **Synthèse de requêtes :** Un des premier scénario d'acquisition à avoir été mis en place est celui de la synthèse de requêtes [1]. Dans celui-ci, l'algorithme d'apprentissage actif peut décider de soumettre à l'oracle n'importe quelle donnée de l'espace d'entrée. On suppose alors qu'il est très facile d'obtenir une donnée particulière. En se référant à la figure 2.1, le bloc *Données d'entrée* correspond simplement à l'espace d'entrée dans sa totalité. En conséquence, il ne contient aucune information quant à la distribution naturelle de ces données. Ce scénario possède l'avantage de pouvoir choisir l'endroit exact de l'espace des entrée ou le critère d'informativité est maximal. Cependant, cela engendre certaines limitations. Premièrement, la donnée ainsi crée peut être difficile à interpréter par l'oracle. Par exemple,

dans [8], le problème étudié consiste à faire de la reconnaissance de chiffres manuscrits. L'oracle est ici un humain qui lit le chiffre inscrit sur l'image qui lui est présentée. Les auteurs ont été confrontés au fait que la majorité des données crées artificiellement n'incluaient aucun chiffre, ou bien un chiffre hybride, et donc que l'oracle ne pouvait pas leur attribuer d'étiquette. Nous pouvons aussi noter que dans le cas où l'apprentissage se fait en utilisant des caractéristiques extraites, comme c'est souvent la cas lorsque l'on traite des images. Alors, il peut être difficile de recréer des données originales à partir des seules caractéristiques demandées. Deuxièmement, la distribution naturelle des données n'est pas connue. Or, certaines stratégies de sélection profitent de la connaissance de la densité des données afin d'accroitre les performances en demandant plus de précision sur les données qui seront plus sollicitées dans la suite, ce qui n'est pas possible ici. Toutefois, ce scénario est bien adapté aux cas où les données d'entrée ne suivent pas une distribution naturelle précise, par exemple dans un problème de contrôle comme celui du bras robotique étudié dans [19]. Dernièrement, afin de sélectionner la prochaine donnée à annoter, il est nécessaire de pouvoir calculer analytiquement le maximum du critère d'informativité sur l'espace d'entrée, ce qui n'est pas toujours trivial.

- **Échantillonnage sélectif sur un flux :** L'échantillonnage sélectif sur un flux [17] consiste à recevoir les données successivement. Pour chacune d'entre elles, l'algorithme d'apprentissage actif peut décider de la soumettre à l'oracle ou bien de la rejeter. En se référant à la figure 2.1, le bloc *Données d'entrée* correspond à une unique donnée d'entrée tirée selon la distribution naturelle. Ainsi, chacune des données reçues est assurée de correspondre à une donnée possible et donc exploitable par l'oracle. De plus, il est possible d'estimer la distribution naturelle des données d'entrée à partir de l'ensemble des données reçues, sélectionnées ou non. A chaque fois qu'une décision est prise, la donnée disponible est remplacée par une nouvelle. Il est important de noter qu'il est impossible de revenir sur sa décision et de faire annoter des données précédentes non-sélectionnées. Le problème majeur est ici que la valeur du critère d'informativité n'est pas relative aux autres données. Ainsi, lors de la prise de décision pour une donnée courante, il est difficile de savoir si elle est assez informative pour être sélectionnée ou bien si il est plus intéressant d'attendre un autre donnée plus informative. Plusieurs solution sont alors envisagées. Tout d'abord, [25] propose de sélectionner les données de façon aléatoire avec une probabilité dépendant de la mesure d'informativité. Un autre approche [17] est de définir à chaque instant une région d'incertitude sur l'espace d'entrée dans laquelle le critère d'informativité est supérieur à un seuil (ici un désaccord d'au moins deux classifieurs sur l'étiquette à attribuer), et de sélectionner la donnée d'entrée proposée uniquement si elle appartient à cette région.

- **Échantillonnage dans un réservoir :** L'échantillonnage dans un réservoir [55] consiste à sélectionner successivement les données à étiqueter parmi un ensemble de données initialement non-étiquetées. En se référant à la figure 2.1, le bloc *Données d'entrée* correspond donc à cet ensemble que l'on appelle réservoir. Le réservoir initial est constitué à partir de données collectées selon leur distribution naturelle, il est donc possible d'estimer cette dernière grâce à celles-ci. Notez que le réservoir est donné une fois pour toute au début du processus, aucune nouvelle donnée d'entrée ne venant s'ajouter, et seules les données sélectionnées pour être étiquetées étant retirées du réservoir. Ainsi, l'estimation de la distribution naturelle des données ne change pas au cours du processus et n'a donc pas besoin d'être recalculée. A chaque pas de temps, le critère d'informativité est calculé sur l'ensemble des données du réservoir et celle pour laquelle il est maximal est sélectionnée pour annotation. L'avantage

de ce scénario vis-à-vis de la synthèse de requêtes est sa rapidité puisque le critère d'informativité n'est calculé que pour un nombre fini de données d'entrée. Cependant, il est plus lent que l'échantillonnage sélectif dans un flux car ce dernier n'évalue à chaque pas de temps le critère d'informativité que pour une seule donnée d'entrée. Toutefois, l'échantillonnage dans un réservoir possède l'avantage de pouvoir calculer exactement la donnée possédant le critère d'informativité maximal.

Finalement, le scénario à utiliser dépend fortement du problème considéré et de ses contraintes. En effet, pour un problème de contrôle, la synthèse de requêtes parait plus adapté car toutes les positions sont accessibles et aucune n'est à privilégier. Si il existe une contrainte de stockage de l'information, comme dans le cas d'un système embarqué, ou si le problème est naturellement en-ligne, alors l'échantillonnage sélectif dans un flux est à choisir. Si un ensemble de données est déjà disponible, et qu'il s'agit de ne soumettre à l'oracle qu'une partie de ces données, alors l'échantillonnage dans un réservoir se prête naturellement au problème.

Nous venons de décrire plusieurs scénarios décrivant la disponibilité des données vis-à-vis de l'algorithme d'apprentissage actif. Il existe en outre différents scénario pour la façon dont ce dernier les soumet à l'oracle. En effet, il est possible de lui passer les données unes par unes ou bien de lui donner tout un groupe de données à annoter simultanément. La différence réside dans le fait que le premier cas la fonction de prédiction est réapprise entre chaque annotation ce qui permet de donner une indication précise pour le choix de la donnée suivantes. Dans le second cas, l'ensemble des données d'un même groupes sont sélectionnées uniquement avec la connaissance des données déjà annotées. Le choix est donc plus fin dans le premier cas. Il existe toutefois plusieurs motivations pouvant justifier un tel choix. Tout d'abord, réapprendre la fonction de prédiction à chaque étape peut être coûteux en temps de calcul, il s'agit alors de minimiser la durée globale en trouvant un compromis entre temps d'annotation et temps d'apprentissage. Ensuite, dans le cas où l'oracle peut gérer plusieurs annotations en parallèle, par exemple en ayant recours à plusieurs annotateurs humains, il serait contre productif d'attendre la réponse d'un d'entre eux avant de soumettre une nouvelle donnée au suivant. Pour profiter de ce parallélisme, il est préférable de sélectionner un groupe de données à être traitées en même temps. Enfin, en pratique, les phases de collecte des données, d'annotation et d'apprentissage peuvent nécessiter d'être séparées dans le temps ou dans l'espace. Ainsi un projet peut être découpé en plusieurs phases de collecte mais il s'agit tout de même de minimiser le nombre de phases requises. Souvent dans ce cas, un travail préalable appelé plan d'expérience est effectué, consistant à définir comment répartir les différentes ressources disponibles (nombre d'échantillons analysés, nombre de personnes affectés, ...) sur les différents produits afin d'en tirer le maximum d'informations, dans ce contexte, l'apprentissage actif peut être relié au domaine des plans d'expérience optimaux. Il existe trois scénarios concernant la soumission des données à l'oracle : l'apprentissage actif entièrement séquentiel, par paquets, et d'une traite. Nous détaillons maintenant ces trois scénarios.

- **Apprentissage actif par paquets :** L'apprentissage actif par paquets [44] [40] [45] est un scénario dans lequel les données sont soumises à l'oracle par groupe de taille fixe, ou paquet, et la fonction de prédiction n'est réapprise qu'une fois que l'ensemble du groupe a été annoté. Ceci permet donc de réduire le temps consacré à l'apprentissage, ainsi que de respecter les contraintes matérielles, comme la disponibilité de plusieurs experts en parallèle. La plupart du temps, cela passe par un problème d'optimisation combinatoire, qui tente de sélectionner l'ensemble de données minimisant le critère d'informativité calculé sur tout le groupe. Ceci demande donc de considérer l'augmentation du temps de calcul pour le critère de sélection.

- **Apprentissage actif entièrement séquentiel :** L'apprentissage actif entièrement séquentiel [27] [74] [77] [6] [52] correspond au scénario précédent pour une taille de paquet fixée à 1, les données sont donc soumises les unes après les autres à l'oracle et la fonction de prédiction est réapprise après avoir reçu chacune de ses réponses. Ce scénario est le plus efficace en termes de performance car le niveau d'informativité de chaque donnée sélectionnée est estimée en considérant le maximum d'information. En outre, si un algorithme d'apprentissage en-ligne est utilisé, il peut donc simplement être mis-à-jour en considérant uniquement la dernière donnée annotée, alors ce scénario peut être effectué avec un temps de calcul raisonnable.

- **Apprentissage actif d'une traite :** L'apprentissage actif d'une traite [75] [37] [38] [36] est un scénario un peu particulier dans la mesure où toutes les données d'entrée à annoter sont déterminées à l'avance avant de n'avoir encore recueilli aucune étiquette. Il n'est donc pas possible de se baser sur l'estimation des paramètres de l'oracle pour déterminer les données d'entrée faisant preuve d'un manque de précision dans leur prédiction. Cependant, à la manière des techniques d'apprentissage non-supervisé, l'étude de la distribution des données disponibles peut donner une propension sur la répartition des classes et ainsi guider le choix des données vers celles ayant le plus de chance de permettre de distinguer des classes. Notez que ce scénario n'est absolument pas séquentiel ou adaptatif, le dialogue avec l'oracle n'étant pas exploité. Alors que dans les scénarios mentionnés précédemment, l'annotation de données pouvaient, en plus d'améliorer la prédiction, servir à mieux sélectionner les données par la suite, ici cet aspect est totalement négligé et seul l'intérêt immédiat est visée.

Dans cette section, nous avons présenté l'apprentissage actif dans son ensemble. Tout d'abord, nous avons fait part des raison motivant la nécessité d'un tel processus. Nous avons vu que la limitation du nombre d'annotations effectuées peut se justifier de diverses façons et intervient dans de nombreux domaines. Nous avons également définit le cadre dans lequel évolueront les algorithmes d'apprentissage actif en entrevoyant les différents scénarios possibles. Ceux-ci définissant la façon dont les données sont acquises au cours du processus ainsi que la façon dont elles sont soumises à l'oracle. Pour l'instant nous n'avons cependant rien dit de la façon dont s'effectue le choix des prochaines données à soumettre à l'oracle. Nous savons uniquement que celui-ci se base sur l'état courant du système, c'est à dire la prédiction du classifieur et la liste des données étiquetées et non-étiquetées disponibles. De nombreuses méthodes ont été développées pour tenter de répondre à cette question et différentes approches ont été envisagées. Dans le reste du chapitre, nous allons donc présenter les principales approches en étudiant les méthodes les plus représentatives pour chacune d'entre elles.

Chapitre 3

Méthodes usuelles d'Apprentissage Actif

3.1 Introduction

Le problème principal de l'apprentissage actif est de savoir quelles sont les meilleures données à faire annoter par l'oracle pour que l'apprentissage du classifieur se fasse le plus rapidement possible. Idéalement, la prédiction effectuée par le classifieur après avoir été entrainé à partir de celles-ci doit posséder une performance plus élevée qu'avec n'importe quel autre jeu, contenant le même nombre de données annotées, qu'il aurait été possible d'obtenir. La solution à ce problème n'est cependant pas évidente. En effet, même en connaissant l'ensemble des paramètres, devant le grand nombre de données disponibles, le calcul direct de la solution optimale peut s'avérer extrêmement lourd. De plus, un grand nombre de paramètres du problèmes ne sont pas connus à l'avance. Par exemple, d'une part il est impossible de prédire la réponse de l'oracle avant de l'avoir effectivement sollicité, et donc de prévoir avec exactitude comment la prédiction du classifieur s'en trouvera modifiée. Ainsi, l'ensemble des données sélectionnées dans le futur ne peuvent pas être déterminé à l'avance. Les algorithmes doivent donc inclure une certaine capacité d'adaptation et faire preuve d'une part d'incertitude et/ou de prudence sur le choix des données. D'autre part, la performance réel d'un classifieur peut être difficile à évaluer. En effet, ni la distribution exacte des étiquettes renvoyées par l'oracle ni la prédiction optimale ne sont connues par avance. En plus de cela, peut venir s'ajouter un bruit, qui empêche de tirer une quelconque certitude des données acquises. Il est donc nécessaire de se définir des heuristiques basées sur les paramètres connus tels que la prédiction courante, l'estimation des paramètres de l'oracle et la liste des données étiquetées et non-étiquetées pour rendre compte de l'intérêt de sélectionner une donnée dans l'atteinte de l'objectif final. Celles-ci vont ensuite être utilisées pour former le critère de sélection des données.

De nombreuses méthodes ont déjà été introduites pour tenter de résoudre le problème d'apprentissage actif. Cette section aura pour but de présenter les axes majeurs autours desquels ces méthodes s'articulent. Nous décrirons ainsi le principe de fonctionnement des méthodes représentatives de chaque approche. Cela nous permettra de nous familiariser avec les notations et les solutions mises-en-œuvres pour traiter le problème. Nous aurons aussi l'occasion de remarquer et d'insister sur le fait que l'ensemble des méthodes dans chaque approche ne sont que différentes variantes d'une même méthode générique. Nos travaux se basant également sur cette méthode générique, il est utile de la présenter ainsi que ses variantes auxquelles nous nous comparerons pour évaluer la pertinence de notre solution. Ceci nous permettra aussi par la suite de situer plus précisément la nature de nos contributions.

Dans la suite, trois approches seront étudiées. La première est l'échantillonnage selon l'incertitude. Elle consiste a sélectionner la donnée pour laquelle le classifieur est le plus indécis. Nous verrons que cela revient à considérer uniquement l'influence locale de l'annotation d'une donnée sur la performance du classifieur. La deuxième est la réduction de l'erreur espérée. Elle consiste à simuler l'échantillonnage des données et à sélectionner celle qui fait le plus accroitre les performances globales du classifieur. Nous verrons que cela revient à considérer l'influence de l'annotation d'une donnée sur la prédiction des données voisines. Finalement, nous présenterons l'approche de réduction de l'espace des version. Celle-ci considère simultanément l'ensemble des classifieurs possibles et tente de sélectionner la donnée qui permet de faire ressortir un classifieur particulier le plus rapidement possible. Bien entendu, d'autres approches existent telles que la réduction de la variance, qui utilise la matrice d'information de Fisher, ou le changement de modèle prévu, qui sélectionne la donnée qui impacte le plus le classifieur. Cependant, nous nous limitons à la présentation de ces premières, d'une part car ce sont les plus appréciées dans la littérature, et d'autre part car ce sont celles qui ont été reprises dans nos travaux et permettent d'en appréhender le contexte. Pour plus de renseignements sur le sujet, nous invitons le lecteur intéressé à lire l'excellent *Active Learning Literature Survey* [66] qui nous a en grande partie inspiré pour constituer cette section.

3.2 Notations

Nous introduisons ici les notations qui seront utilisées dans le reste du manuscrit et qui sont nécessaires à la définition des stratégie d'apprentissage actif.

Tout d'abord, les notations relatives à la classification :

- X l'espace d'entrée, $x \in X$ une donnée d'entrée et $P(x)$ la distribution naturelle des données d'entrée,
- Y l'ensemble des étiquettes, $y \in Y$ une étiquette et $P(y|x)$ la distribution de probabilité représentant l'oracle,
- $\mu(x) = P(y = 1|x)$ dans le cas de la classification binaire où $Y = \{0, 1\}$, notez que dans le cas de la classification non-bruitée $\mu(x) \in \{0, 1\}$,
- $\hat{}$ désigne un estimateur, par exemple : $\hat{P}_\theta(y|x)$ est l'estimateur de $P(y|x)$ étant associé à des paramètres θ,
- $f : X \to Y$ une fonction de prédiction associant une donnée de sortie à chaque donnée d'entrée, celle-ci peut posséder des paramètres θ, on la note alors f_θ.

Les ensembles discrets sont dénotés par des lettres majuscules calligraphiées. Ainsi, on note :

- $\mathcal{R}$ le réservoir,
- $\mathcal{L}$ l'ensemble des données étiquetées,
- $\mathcal{U}$ l'ensemble des données non-étiquetées,
- $\mathcal{T}$ la base d'apprentissage,

De même que les loi de probabilité standards.

- $\mathcal{B}er(p)$ la loi de Bernoulli de paramètre p,

- $\mathcal{B}in(n,p)$ la loi binomiale de paramètres n et p,
- $\mathcal{B}eta(a,b)$ la loi Beta de paramètres a et b,
- $\mathcal{N}(\mu,\sigma^2)$ la loi normale de moyenne μ et de variance σ^2,
- $\mathcal{I}_{a,b}(.)$ la fonction Beta incomplète qui est la fonction de répartition de la loi Beta de paramètres a et b,
- Φ la fonction de répartition de la loi normale.

On note également :

- $\mathbb{E}_{va\sim\nu}[.]$ l'espérance par rapport à une variable aléatoire va tirée selon la distribution ν,
- $\mathbb{P}_{va\sim\nu}(.)$ la probabilité d'une variable aléatoire va tirée selon la distribution ν,
- $\mathbb{1}_{test}$ la fonction indicatrice valant 1 si $test$ est vrai et 0 sinon.
- $\pm_{test}$ la fonction "plus ou moins" valant $+1$ si $test$ est vrai et -1 sinon.

A cela s'ajoutent les notations :

- $|.|$ l'opérateur valeur absolue,
- $\lfloor . \rceil$ l'opérateur arrondi.

3.3 Mesures de performances

Afin d'évaluer la qualité d'une fonction de prédiction f, différentes mesures de performance peuvent être utilisées. En général, on évalue plutôt la contre-performance qui s'exprime à l'aide d'un risque, celui-ci croît lorsque la performance décroît. Nous décrivons ici plusieurs de ces mesures, chacune pouvant mener à une évaluation de la performance locale ou de la performance globale. Pour une mesure donnée m, le risque de f, notée $R_m(f)$, représente la contre-performance globale obtenue en lui soumettant une donnée d'entrée tirée aléatoirement selon leur distribution naturelle. Nous définissons ici le risque local de f, notée $r_m(f,x)$, une fonction d'une donnée d'entrée, correspondant au risque obtenu lorsqu'on soumet à la fonction de prédiction uniquement cette donnée d'entrée. Ainsi,

$$R_m(f) = \mathbb{E}_{x\sim P(x)}[r_m(f,x)]$$

Les risques suivants se basent sur une fonction de perte l_m, celle-ci permet de comparer deux données de sortie en leur attribuant une erreur. Le risque local est alors la perte moyenne commise par la fonction de prédiction :

$$r_m(f,x) = \mathbb{E}_{y\sim P(y|x)}[l_m(f(x),y)]$$

Risque basé sur la perte quadratique Dans le cas de la régression, le but est que la valeur prédite soit proche de la donnée de sortie associée par l'oracle. Ainsi, la perte utilisée est l'erreur quadratique :

$$\forall (y,y') \in Y^2, \quad l_2(y,y') = (y-y')^2.$$

Ce qui mène à

$$\begin{aligned} r_2(f,x) &= \mathbb{E}_{y\sim P(y|x)}[(f(x)-y)^2] \\ &= (f(x) - \mathbb{E}_{y\sim P(y|x)}[y])^2 + Var(y|x). \end{aligned}$$

Risque basé sur la perte binaire En classification, la fonction de perte binaire, ou perte-0/1, est utilisée. Elle représente l'erreur de classification, qui est nulle si l'étiquette prédite est juste et vaut 1 autrement. Ceci qui permet de compter les données mal-classifiées :

$$\forall (y, y') \in Y^2, \quad l_{0/1}(y, y') = \begin{cases} 0 & \text{si } y = y' \\ 1 & \text{sinon.} \end{cases}$$

Ainsi, le risque local est donc la proportion de données mal classifiées :

$$r_{0/1}(f, x) = \sum_{i=1}^{K} \mathbb{1}_{f(x) \neq y}(1 - P(y = i|x))$$

Log-vraisemblance Une mesure de la performance peut être la log-vraisemblance du paramètre estimé par rapport aux observations, utilisée en régression logistique par exemple. Il est possible de définir une fonction de perte menant à cette mesure, représentant l'opposé de la log-vraisemblance d'un paramètre quelconque vis à vis d'une seule observation :

$$\forall (y, p) \in Y \times [0, 1]^K t.q. \sum_{i=1}^{K} p(i) = 1, \quad l_{\mathcal{L}}(y, p) = -\sum_{i=1}^{K} \mathbb{1}_{y=i} \log(p(i)).$$

Ainsi, du fait de la linéarité de la log-vraisemblance par rapport aux observations, le risque local devient :

$$r_{\mathcal{L}}(f_\theta, x) = -\sum_{i=1}^{K} P(y = i|x) \log(\hat{P}_\theta(y = i|x)).$$

Perte charnière La perte utilisée dans les machines à vecteurs support est la perte charnière définie pour la classification binaire par :

$$\forall (y, p) \in Y \times \mathbb{R}, \quad l_h(y, p) = \max(0, 1 - y.p).$$

Erreur de prédiction Quelques fois, seul prédire l'étiquette correcte est demandé, sans s'intéresser à l'erreur engendrée. Dans ce cas, le risque local associé est :

$$r(f, x) = \mathbb{1}_{f(x) \neq y^{opt}(x)},$$

où $y^{opt}(x) = \arg\max_i P(y = i|x)$.

3.4 Échantillonnage d'incertitude

Deux caractéristiques sont importantes dans le choix de la donnée à sélectionner, l'informativité représente le potentiel à améliorer la prédiction tandis que la représentativité mesure à quel point la donnée est similaire aux données disponibles pour l'annotation. Dans cette section, nous nous intéressons particulièrement aux stratégies de sélection se basant uniquement sur une mesure locale de l'informativité. C'est à dire que la sélection d'une donnée d'entrée n'a pour but que d'améliorer la prédiction pour cette même donnée d'entrée, son influence sur son voisinage étant totalement ignorée. Toutefois, la représentativité peut elle aussi être prise en compte afin de guider cette sélection vers les données les plus probables. En effet, si une donnée d'entrée sera plus sollicitée qu'une autre lors de l'utilisation de la fonction de prédiction, alors une erreur de prédiction impactera d'autant plus l'erreur globale, il est donc nécessaire que la prédiction associée à une donnée plus probable soit plus fiable. Les méthodes présentées dans cette section se divisent donc en deux parties, dans la première, seule l'informativité est considérée. Dans la deuxième partie, le critère de sélection se base à la fois sur la représentativité des données et sur une mesure locale de l'informativité.

Les travaux présentés dans [55] et [54], appelées échantillonnage d'incertitude, constituent la première approche de l'apprentissage actif en classification binaire. Pour cela, un classifieur probabiliste est utilisé afin que le score résultant fournisse une mesure de la certitude de la prédiction. L'idée est alors d'appliquer le classifieur sur toutes les données non-étiquetées, celles dont le score est proche de $\frac{1}{2}$ sont celles pour lesquelles le classifieur est le plus incertain de l'étiquette à lui attribuer. Ainsi, la stratégie proposée est de sélectionner cette donnée pour l'annoter afin d'en améliorer la prédiction :

$$x_{t+1} = \arg\min_{x \in \mathcal{U}} |\hat{P}_\theta(y = 1|x) - \frac{1}{2}|.$$

Une interprétation possible de cette stratégie est de considérer que l'objectif visé est de garantir une précision minimum quelle que soit la données d'entrée reçue. Ainsi, cela revient à évaluer la performance globale d'une fonction de prédiction à travers le risque local maximal sur l'espace d'entrée : pour une mesure de performance m quelconque,

$$L_m(f) = \max_{x \in X} r_m(f, x).$$

Pour réduire le risque local maximal, la solution est d'échantillonner la donnée d'entrée atteignant ce maximum. En effet, en échantillonnant à cet endroit, le risque local est garanti de décroître, changeant alors la valeur du risque local maximal et/ou son emplacement sur l'espace d'entrée. Cependant, le risque local est fonction de la distribution de l'oracle qui est inconnue. Par exemple, pour la perte binaire, et en classification binaire :

$$r_{0/1}(f, x) = \mathbb{1}_{f(x)=0}\mu(x) + \mathbb{1}_{f(x)=1}(1 - \mu(x)),$$

et le paramètre $\mu(x)$ est inconnu. Il est toutefois possible d'estimer ce paramètre de la distribution de l'oracle par $\hat{P}_\theta(y = 1|x)$, de même la prédiction $f_\theta(x)$ se base sur cette estimation : $f_\theta(x) = 1 \iff \hat{P}_\theta(y = 1|x) > 0$. Dans ce cas, le risque local estimé est

$$\hat{r}_{0/1,\theta}(f_\theta, x) = \frac{1}{2} - |\hat{P}_\theta(y = 1|x) - \frac{1}{2}|.$$

Sélectionner la donnée pour laquelle le risque local estimé est maximal nous donne alors exactement la stratégie de sélection de l'échantillonnage d'incertitude.

Une autre interprétation est liée à la notion de séparateur. Celui-ci est la frontière dans l'espace d'entrée qui sépare les données étiquetées différemment. Ainsi, cette stratégie sélectionne les données les plus proche du séparateur, car c'est celles qui permettent d'affiner au mieux son emplacement.

L'échantillonnage d'incertitude peut aussi être adapté avec des classifieurs non-probabilistes. Par exemple, dans [54], des arbres de décision sont utilisés, le score servant de mesure de l'incertitude étant ici la proportion d'étiquettes reçues dans la feuille de l'arbre correspondante à la donnée traitée. De même, dans [29] ou encore [57], la méthode est adaptée aux K-plus-proches-voisins en utilisant la proportion de chaque classe parmi les étiquettes des voisins considérés comme mesure de l'incertitude. Enfin, cette approche peut aussi être reliée à la méthode dite de simple marge pour les machines à vecteurs support [74]. Ici, le score utilisé est la distance de la donnée traitée au séparateur, la donnée non étiquetée la plus proche étant sectionnée pour l'annotation. De plus, cette méthode est fournie avec des justifications sur la performance obtenue, en étudiant l'effet sur l'espace des versions que nous aborderons dans une prochaine section.

Dans le cas multi-classes, l'utilisation du principe de l'échantillonnage de l'incertitude est sujette à différentes interprétations, provenant de la mesure de performance utilisée. Dans [24] et [67], la méthode utilisée consiste à échantillonner la donnée pour laquelle la prédiction courante est la plus incertaine, c'est à dire pour laquelle la probabilité estimée que l'oracle renvoie l'étiquette prédite par le classifieur courant est la plus faible :

$$x_{t+1} = \arg\min_{x \in \mathcal{U}} \hat{P}_\theta(y = f_\theta(x)|x),$$

sachant que $f_\theta(x) = \arg\max_{i \in \{1,\dots,K\}} \hat{P}_\theta(y = i|x)$. Comme précédemment, cette méthode peut être interprétée par le fait que l'on cherche à minimiser le risque local maximal, en le basant sur la fonction de perte binaire. Rappelons que ce risque local s'exprime par :

$$\begin{aligned} r_{0/1}(f, x) &= \sum_{i=1}^{K} \mathbb{1}_{f(x) \neq y}(1 - P(y = i|x)) \\ &= 1 - P(y = f(x)|x) \end{aligned}$$

Ainsi, en remplaçant la probabilité de l'oracle par son estimée :

$$\hat{r}_{0/1,\theta}(f_\theta, x) = 1 - \hat{P}_\theta(y = f_\theta(x)|x).$$

Et donc, la stratégie précédente permet en effet de minimiser le risque local maximal.

Dans [48] et [67], l'idée est d'utiliser l'entropie de Shannon [70] comme mesure de l'incertitude. Ainsi, la stratégie adoptée est de sélectionner la donnée d'entrée pour laquelle l'entropie est maximale :

$$x_{t+1} = \arg\max_{x \in \mathcal{U}} -\sum_{i=1}^{K} \hat{P}_\theta(y = i|x) \log(\hat{P}_\theta(y = i|x)).$$

Ici, la mesure de performance utilisée serait la log-vraisemblance. En effet :

$$r_{\mathcal{L}}(f_\theta, x) = -\sum_{i=1}^{K} P(y = i|x) \log(\hat{P}_\theta(y = i|x)),$$

si la probabilité de l'oracle est estimée, alors

$$\hat{r}_{0/1,\theta}(f_\theta, x) = -\sum_{i=1}^{K} \hat{P}_\theta(y = i|x) \log(\hat{P}_\theta(y = i|x)),$$

ce qui est exactement l'entropie.

Dans le cas de la régression [18] [19], la mesure de performance utilisée est la perte quadratique. Le risque local s'exprime ainsi

$$r_2(f,x) = (f(x) - \mathbb{E}[y|x])^2 + Var(y|x),$$

et en remplaçant l'espérance et la variance de la distribution de probabilité de l'oracle par leur estimation :

$$\hat{r}_{2,\theta}(f_\theta, x) = \widehat{Var}_\theta(y|x),$$

en rappelant que la valeur de la prédiction est ici égale à l'estimation de l'espérance des données de sortie.

Pour toutes les stratégies présentée jusqu'ici, la valeur des paramètres inconnus étaient remplacés directement par la valeur de leur estimation. Dans le cas où une approche Bayésienne permet d'obtenir une distribution *a posteriori* sur les paramètres de la distribution de l'oracle, il est possible de l'utiliser dans l'estimation du risque local. Ainsi, dans [49], les auteurs étudient un stratégie de sélection locale des données d'entrée dans le cadre des Processus Gaussiens pour la classification binaire. Ici, une distribution *a priori* Gaussienne sur le paramètre $\mu(x)$ est considérée, soit $\hat{\mu}(x)$ et $\hat{\sigma}^2(x)$ les paramètres de la distribution *a posteriori* :

$$\mu(x) \sim \mathcal{N}(\hat{\mu}(x), \hat{\sigma}(x)^2).$$

Le risque local utilisé est l'erreur de prédiction :

$$r(f,x) = \mathbb{1}_{f(x) \neq \arg\max_i P(y=i|x)},$$

Le but est toujours de sélectionner la donnée d'entrée pour laquelle le risque local est maximal, mais plutôt que de remplacer le paramètre par son estimation, l'idée est d'utiliser risque local espéré sur la distribution *a posteriori*. Ainsi,

$$\hat{r}(f,x) = \mathbb{E}_{\mu(x) \sim \mathcal{N}(\hat{\mu}(x), \hat{\sigma}(x)^2)}[\mathbb{1}_{f(x) \neq \arg\max_i P(y=i|x)}].$$

Sachant que $f(x) = \lfloor \hat{\mu}(x) \rceil$,

$$\begin{aligned}
\hat{r}(f,x) &= \mathbb{1}_{\lfloor \hat{\mu}(x) \rceil = 1} \Phi(\frac{1}{2}, \hat{\mu}(x), \hat{\sigma}(x)^2) + \mathbb{1}_{\lfloor \hat{\mu}(x) \rceil = 0}(1 - \Phi(\frac{1}{2}, \hat{\mu}(x), \hat{\sigma}(x)^2)) \\
&= \mathbb{1}_{\lfloor \hat{\mu}(x) \rceil = 1} \Phi(\frac{\frac{1}{2} - \hat{\mu}(x)}{\hat{\sigma}(x)}, 0, 1) + \mathbb{1}_{\lfloor \hat{\mu}(x) \rceil = 0} \Phi(\frac{\hat{\mu}(x) - \frac{1}{2}}{\hat{\sigma}(x)}, 0, 1) \\
&= \Phi(-\frac{|\frac{1}{2} - \hat{\mu}(x)|}{\hat{\sigma}(x)}, 0, 1),
\end{aligned}$$

avec Φ la fonction de répartition de la loi normale.

Ainsi, sachant que Φ est strictement croissante, échantillonner la donnée d'entrée pour laquelle le risque local estimé est maximal mène à la stratégie de sélection suivante :

$$x_{t+1} = \arg\min_{x \in \mathcal{U}} \frac{|\frac{1}{2} - \hat{\mu}(x)|}{\hat{\sigma}(x)}.$$

Notez que ce critère de sélection est très similaire à celui de l'échantillonnage d'incertitude, la différence portant uniquement sur le fait que l'incertitude est pondérée par la variance de l'estimateur de $\mu(x)$. Pour justifier cela, plaçons nous dans un problème de classification bruitée, la

vraie valeur du paramètre de la distribution de Bernoulli peut alors potentiellement être elle même proche de $\frac{1}{2}$. Si c'est le cas, une bonne estimation de ce paramètre sera aussi proche de $\frac{1}{2}$ sans pour autant que cela n'indique que la prédiction soit incertaine puisque l'étiquette optimale ne dépend que de l'arrondi du vrai paramètre. De plus, l'acquisition d'une nouvelle réponse de l'oracle pour cette donnée d'entrée n'améliorera pas le critère d'incertitude car même estimation parfaite donnera la même valeur du critère. La véritable incertitude sur la prédiction découle de la variance de l'estimateur, qui est calculé en fonction du nombre et de la distance des données d'entrée ayant servi à l'estimation. Toutefois, plus le paramètre estimé est loin de $\frac{1}{2}$, moins la variance de l'estimateur affecte l'incertitude sur la prédiction. Ainsi, la stratégie de sélection présentée ici permet de combiner ces deux aspect, en particulier dans le cas où un a priori Gaussien sur le paramètre peut être effectué.

Toutes les stratégies qui ont été présentées jusqu'à présent ne tiennent compte que de l'informativité des données et non de leur représentativité, elles n'incluent aucune information quant à la distribution naturelle des données d'entrée. Elles trouvent toutefois leur utilité si cette dernière peut être supposée uniforme. D'autre part, dans certains cas, parler d'une distribution naturelle pour les données d'entrée n'a pas de sens, comme pour les problèmes de contrôle où les données d'entrée utilisées pour l'apprentissage sont entièrement choisies par l'utilisateur et où l'utilisation faite du système ne peut pas être connue *a priori*. Dans ce cas, les stratégies de sélection doivent se limiter au seul critère d'informativité.

Dans les stratégies de sélection qui vont suivre, la représentativité vient s'ajouter à la mesure de l'informativité locale. Ceci possède plusieurs avantages, tout d'abord, nous avons déjà évoqué que la distribution naturelle des données d'entrée implique que lors de l'utilisation du classifieur certaines données soient plus sollicitées que d'autre. Ainsi, l'erreur de prédiction commise pour une donnée d'entrée affectera le risque global d'autant plus qu'elle est sollicitée. Dans le but de répartir équitablement l'erreur moyenne de prédiction sur l'espace d'entrée, il est nécessaire que la qualité de la prédiction soit plus grande pour une donnée d'entrée plus probable. Ensuite, un problème récurrent en apprentissage actif est que le jeu de données étiquetées ne suit pas la distribution naturelle des données d'entrée. Autrement dit, les données auxquels sera soumis le classifieur et les données servant à son apprentissage ne sont pas identiquement distribuées. La plupart des classifieur reposant sur une hypothèse *i.i.d.*, cela peut affecter gravement la prédiction. Pour contrer cela, la méthode dite de pondération selon l'importance [9] permet d'affecter un poids aux données de la base d'entrainement en fonction de leur densité. Il en advient que les données possédant un plus grand poids sont plus informatives car influant plus sur la prédiction. Les deux problèmes qui viennent d'être décrits sont en réalité deux aspects d'un même problème nécessitant de prendre en compte la densité des données dans leur sélection.

Dans [60] et [67], les auteurs introduisent une stratégie de sélection combinant à la fois l'informativité des données et leur densité par un simple produit. Ainsi, soit $CI(x)$ un critère d'informativité pure, $D(x)$ une estimation de la densité des données, et β un paramètre quelconque servant à gérer le compromis entre informativité et densité, alors :

$$x_{t+1} = \arg\min_{x \in \mathcal{U}} CI(x) \times D(x)^{\beta}.$$

Nous pouvons voir qu'une donnée sera ainsi privilégiée pour la sélection si elle est plus informative ou si elle est liée à une densité plus forte.

Dans [60], le critère utilisé est un cas particulier du critère ci-dessus appliqué à un comité d'experts, c'est à dire un ensemble de classifieurs dont les prédictions sont combinées pour définir la prédiction finale. Les classifieurs utilisés sont des classifieurs probabilistes appelés classifieurs naïfs de Bayes. Le facteur β est ici égal à 1.

Dans [67], le critère d'informativité pure est quelconque, cela peut être n'importe quel critère décris précédemment. La densité est, elle, estimée en effectuant la moyenne des similarités entre la donnée d'entrée considérée et toutes les autres données d'entrée disponibles :

$$D(x) = \frac{1}{card(\mathcal{U})} \sum_{x' \in \mathcal{U}} sim(x, x'),$$

avec $sim \in \mathbb{R}^{X^2}$ la mesure de similarité considérée.

Une interpretation de ce critère est que la fonction a minimiser est

$$L_m(f) = \max_{x \in X} r_m(f, x) \times dP(x).$$

En effet, rappelons que le risque global est

$$R_m(f) = \int_{x \in X} r_m(f, x) \times dP(x),$$

ainsi, $L_m(f)$ borne $R_m(f)$, le minimiser permet donc de minimiser le risque global. La solution serait alors d'échantillonner la donnée pour laquelle le critère $r_m(f, x) \times dP(x)$ est maximal, celui-ci est alors estimé en estimant indépendamment $r_m(f, .)$ par CI et dP par D. Ceci nous permet une intuition quant à la valeur de β à utiliser, en effet dans l'expression du risque global ci-dessus le risque local et la densité sont à niveau égal, le paramètre β permettant de relier le critère définit dans [67] et celui ci-dessus est donc égal à 1. Notez que dans leur expérimentations, les auteurs de [67] fixent la valeur du paramètre à 1.

Cette méthode, en estimant indépendamment l'informativité et la densité permet de se servir de deux jeux de données différents (mais pas forcement disjoints). En effet, pour calculer la densité des données, il n'est pas nécessaire de connaitre leur étiquette. Ainsi, dans le scénario d'échantillonnage dans un réservoir, la densité peut être calculée une seule fois au début du processus grâce à l'ensemble des données du réservoir. De cette façon, d'une part l'estimation de la densité est très bonne car elle prend en compte un grand nombre de données, et d'autre part cette estimation n'a pas besoin d'être recalculée à chaque pas de temps, car aucune nouvelle donnée ne vient s'ajouter au réservoir au cours du processus, ce qui permet de limiter le temps de calcul. De plus, le fait que l'estimation de la densité ne change pas permet de l'utiliser comme telle, sans considérer l'incertitude liée à son estimation, et de ne pas avoir à repenser la répartition des données précédemment sélectionnées. Dans le scénario d'échantillonnage dans un flux, les données rejetées peuvent tout de même servir à améliorer l'estimation de la densité.

Dans le problème de pondération selon l'importance, le coût $\ell(f(x_t), y_t)$ associé à chaque donnée d'entrée x_t contenue dans la base d'apprentissage peut être pondéré par un facteur $w(x_t)$ afin de mener à une estimation non biaisée du risque global engendré par un tirage des données selon leur distribution naturelle. Soit $\mathcal{P}_\mathcal{T}$ la distribution des données d'entrée dans la base d'apprentissage, rappelons que la distribution naturelle des données d'entrée est notée P, les poids optimaux à utiliser sont :

$$\forall x \in X, w(x) = \frac{P(x)}{P_\mathcal{T}(x)}.$$

De sorte que

$$\begin{aligned}\mathbb{E}_{x_t\sim P_{\mathcal{T}}(x),y_t\sim P(y|x)}[w(x_t)\ell(f(x_t),y_t)] &= \int_{X\times Y} w(x_t)\ell(f(x_t),y_t)P_{\mathcal{T}}(x)P(y|x)dxdy \\ &= \int_{X\times Y} \ell(f(x_t),y_t)P(x)P(y|x)dxdy \\ &= \mathbb{E}_{x_t\sim P(x),y_t\sim P(y|x)}[\ell(f(x_t),y_t)] \\ &= R(f).\end{aligned}$$

Le problème qui se pose est alors que les distributions mises en jeux sont difficiles à estimer avec un nombre limité de données d'entrée. L'idée proposée dans [9] est de ne pas chercher à estimer les distributions mais de contrôler directement le rapport de ces distributions. En effet, il est possible de biaiser le tirage aléatoire des données d'entrée selon leur distribution naturelle avec une probabilité que l'on connait entièrement, puis de définir le poids affecté à la donnée tirée en fonction de cette probabilité.

L'algorithme proposé est défini pour le scénario d'échantillonnage dans un flux. A chaque pas de temps t, une donnée d'entrée est tirée selon sa distribution naturelle $P(x_t)$. Celle-ci est ensuite sélectionnée pour être annotée par l'oracle avec une probabilité $p_t(x_t)$ établie par l'algorithme et donc entièrement connue. Au total, la donnée a été tirée selon la probabilité $P_{\mathcal{T}}(x_t) = P(x_t) \times p_t(x_t)$. Le poids affecté à la donnée est alors le rapport de la probabilité naturelle et de la probabilité finale avec laquelle elle a été tirée :

$$w_t = \frac{1}{p_t(x_t)},$$

il ne reste donc que la probabilité établie et entièrement connue.

Ainsi, il est possible d'imposer une tendance au tirage des données d'entrée sans affecter l'estimation du risque du classifieur. Il reste à savoir comment définir cette tendance pour que les données ainsi sélectionnées servent le problème d'apprentissage actif. En prenant comme valeur pour $p_t(x_t)$ n'importe quel critère d'informativité pure défini précédemment, ramené dans l'intervalle $[0, 1]$ si nécessaire, alors les données seront privilégiées si elles correspondent à une forte valeur du risque local supposé. Dans [9], le critère d'informativité pure utilisé est le désaccord maximal entre deux classifieurs d'un comité d'experts.

Cet algorithme a aussi été adapté au scénario d'échantillonnage dans un réservoir dans [31]. En effet, les données contenues dans le réservoir ont déjà été tirées selon $P(x)$ de sorte que sélectionner une donnée de façon équiprobable dans le réservoir revient à tirer cette donnée selon la distribution naturelle. A chaque pas de temps t, cet algorithme associe à chaque donnée du réservoir $x_u \in \mathcal{U}$ une probabilité $p_t(x_u)$ calculée à partir d'un critère d'informativité pure quelconque qui est normalisé pour sommer à 1 sur l'ensemble des données du réservoir. La donnée à soumettre à l'oracle est ensuite tirée aléatoirement selon la distribution de probabilité ainsi crée sur le réservoir.

$$x_{t+1} \sim p_t(x_u).$$

Au total, la probabilité de sélectionner à l'instant t une donnée x_t quelconque de l'espace d'entrée X est égale à $P(x_t) \times p_t(x_t)$, ce qui nous ramène au cas précédent où le poids associé à cette donnée d'entrée est $w_t = \frac{1}{p_t(x_t)}$. Dans [31], le critère d'informativité pure utilisé est l'entropie de Shannon.

Finalement, cette approche nous donne une autre façon de combiner l'informativité et la représentativité. Notez que bien que la densité des données d'entrée n'intervienne pas directement dans

le critère de sélection, elle influence tout de même le choix de la prochaine donnée d'entrée à annoter de façon cachée. En effet, puisque toutes les données d'entrée sont considérée dans la sélection, que le critère d'informativité pure est complètement décorrélé de la densité, et que la probabilité de sélectionner la prochaine donnée dans une certaine zone est la somme des probabilités de chacune des données de la zone, alors la probabilité de sélectionner cette données dans une zone dense est plus forte que dans une zone faiblement peuplée. L'avantage par rapport aux méthodes précédentes considérant la représentativité est que la densité n'a pas besoin, ici, d'être estimée car la densité réelle est prise en compte de façon sous-jacente. Cela limite les imprécisions mais aussi le temps de calcul lié à l'estimation de la densité. Cette approche permet donc de gérer le compromis entre informativité et représentativité efficacement en partant de la sélection aléatoire suivant la distribution naturelle des données d'entrée et en la modifiant pour privilégier les données informatives. Ceci permet en outre de gérer laisser une place à l'exploration de l'espace d'entrée dans le but d'améliorer l'estimation de l'informativité. Ainsi, cette méthode nous fourni une manière de gérer le compromis entre exploration et exploitation, ceci étant au centre de nos travaux, il sera abordé plus en détail dans la suite.

Dans cette section, nous avons vu différentes stratégies de sélection utilisant une mesure de l'informativité locale d'une donnée d'entrée. Dans un premier temps, nous avons étudié des critères d'informativité pure, permettant uniquement de mesurer la qualité de la prédiction. En considérant qu'un critère de sélection optimal nécessite la connaissance du paramètre de la distribution de l'oracle, nous avons vu deux manières de gérer l'incertitude sur ce dernier. Soit il est directement remplacé par son estimation, où nous avons vu que plusieurs mesures de performances pouvaient être utilisées, mais que de celles-ci ne découlaient des stratégies différentes que dans le cas multi-classe. Soit, dans le cadre d'une approche Bayésienne, le critère optimal est espéré sur la distribution *a posteriori* du paramètre. Dans un deuxième temps, nous avons étudié des stratégies permettant de sélectionner des données d'entrée à la fois informative et représentatives, c'est à dire dont la qualité de prédiction est faible et qui sont beaucoup sollicitées. Nous avons vu deux manières de prendre en compte la distribution naturelle des données d'entrée. Soit la densité est estimée et est incluse dans le critère de sélection. Soit la distribution avec laquelle les données disponible ont été tirées est entretenue et simplement altérée par le critère d'informativité pure. Dans la prochaine section, nous verrons que le critère de sélection peut être amélioré si l'on tient compte de l'influence de l'acquisition d'une donnée d'entrée sur la qualité de prédiction d'autres données d'entrée.

3.5 Réduction de l'erreur

Dans la partie précédente, une donnée d'entrée était considérée informative si la prédiction du classifieur associée à cette donnée d'entrée était peu précise et avait des chances d'être améliorée. Cependant, il n'y avait aucune prise en compte de l'influence de cette donnée d'entrée sur les autres parties de l'espace d'entrée. En effet, tout classifieur se doit de généraliser afin de pouvoir attribuer une étiquette à une donnée d'entrée encore jamais rencontrée (n'appartenant pas à la base d'entrainement). Ainsi, lorsqu'une donnée d'entrée est annotée et incluse dans la base d'apprentissage, elle améliore non seulement la prédiction locale, mais aussi celle de l'ensemble de l'espace d'entrée, avec en général une influence plus prononcée pour les données d'entrée proches. En conséquent, l'informativité associée à une donnée d'entrée doit considérer l'ensemble de ces améliorations afin de minimiser le plus rapidement possible le risque global. Dans cette section, nous présentons différentes stratégies de sélection construits dans cet optique.

Notez qu'il peut être difficile de calculer formellement l'influence que pourrait avoir l'annotation d'une donnée d'entrée sur l'ensemble des prédictions du classifieur. De plus, comme dans la partie précédente, il serait nécessaire de pondérer ces influences par la densité des données d'entrée, qui n'est pas connue *a priori* et doit être estimée. L'idée principale commune à tous les algorithmes présentés ici est que plutôt que de faire ça, il est possible de simuler l'annotation d'une donnée d'entrée en lui affectant une étiquette sans passer par l'oracle, de réapprendre le classifieur sur cette nouvelle base d'entrainement, puis d'estimer le gain résultant sur le risque global. Ceci possède plusieurs avantages, tout d'abord, l'information de densité étant déjà comprise dans le risque global, cela permet d'étudier la représentativité des données conjointement avec leur informativité, et de ne pas devoir la traiter comme un problème séparé. Deuxièmement, cela permet d'utiliser un classifieur méconnu, sans s'intéresser à la façon dont celui-ci est construit, en lui soumettant aveuglement une base d'apprentissage et en récupérant un score sur l'étiquette à prédire.

Les algorithmes présentés ici diffèrent sur la manière dont est gérée l'incertitude relative aux paramètres inconnus de la distribution de l'oracle. Celle-ci intervient dans deux étapes de l'estimation du critère d'informativité. Tout d'abord, comme dans la partie précédente, ces paramètres déterminent le risque réel, en effet, selon la prédiction effectuée, l'erreur moyenne dépend de la proportion de chaque classe dans les étiquettes renvoyées par l'oracle. Ainsi, la prévision du gain sur le risque global dépend de la qualité de l'estimation de chacun des risques. D'autre part, pour avoir une cohérence entre le gain prédit et le gain effectif, l'étiquette attribuée lors de la simulation d'annotation doit être la plus proche possible de l'étiquette qui sera reçu lors de la requête à l'oracle. Ainsi, celle-ci dépend aussi du paramètre de la distribution de l'oracle qu'il est nécessaire d'estimer.

En plus du fait que la distribution de l'oracle ne soit pas connue, le calcul du risque global nécessite la connaissance de la distribution naturelle des données d'entrée. Dans le cas de l'échantillonnage dans un réservoir, l'estimation de cette dernière se base sur la répartition des données d'entrée dans le réservoir et peut prendre en compte aussi bien les données déjà étiquetées que non-étiquetées. Ainsi, le risque global pour une mesure de performance m quelconque

$$R_m(f) = \mathbb{E}_{x\sim P(x), y\sim P(y|x)}[\ell(f(x), y)]$$

est estimé par

$$\hat{R}_{m,\mathcal{R}}(f) = \frac{1}{card(\mathcal{R})} \sum_{x_r \in \mathcal{R}} \mathbb{E}_{y\sim P(y|x_r)}[\ell(f(x_r), y)].$$

Notez qu'ici l'estimation ne porte que sur la distribution naturelle des données d'entrée, la distribution de l'oracle étant toujours considérée connue. En effet, une seconde estimation portant sur la

distribution de l'oracle sera définie dans la suite mais celle-ci diffère selon les méthodes, nous la détaillerons donc dans chaque cas.

La première méthode que nous présentons a été introduite en parallèle dans [65] et [78]. L'idée est, de la même manière que dans l'échantillonnage selon l'incertitude présenté dans la section précédente, de remplacer directement le paramètre de la distribution de l'oracle par son estimation. Ainsi, soit $\mathcal{T}$ la base d'entrainement utilisée pour apprendre le classifieur $f_{\theta(\mathcal{T})}$, le risque global est estimé par

$$\hat{R}_{m,\mathcal{R},\theta(\mathcal{T})}(f_{\theta(\mathcal{T})}) = \frac{1}{card(\mathcal{R})} \sum_{x_r \in \mathcal{R}} \hat{P}_{\theta(\mathcal{T})}(y=1|x_r)\ell(f(x_r),1) + \hat{P}_{\theta(\mathcal{T})}(y=0|x_r)\ell(f(x_r),0).$$

Par exemple, dans le cas où le risque est basé sur la perte binaire, comme dans [65] et [78] :

$$\hat{R}_{0/1,\mathcal{R},\theta(\mathcal{T})}(f_{\theta(\mathcal{T})}) = \frac{1}{card(\mathcal{R})} \sum_{x_r \in \mathcal{R}} (1 - \max_{y \in Y} \hat{P}_{\theta(\mathcal{T})}(y|x_r)).$$

Et dans le cas où le risque est la log-vraisemblance, comme dans [65] :

$$\hat{R}_{\mathcal{L},\mathcal{R},\theta(\mathcal{T})}(f_{\theta(\mathcal{T})}) = \frac{1}{card(\mathcal{R})} \sum_{x_r \in \mathcal{R}} \sum_{y \in Y} \hat{P}_{\theta(\mathcal{T})}(y|x_r) \log(\hat{P}_{\theta(\mathcal{T})}(y|x_r)).$$

En réalité, dans [65], le risque global n'est pas estimé sur l'ensemble du réservoir mais uniquement sur les données non-étiquetées, qui sont supposées se trouver en très grand nombre. Cependant, rappelons que cela ne sert qu'à estimer la distribution naturelle, il n'y a donc pas de problème à utiliser les données déjà étiquetées comme dans [78] car elles suivent aussi cette distribution. De plus, considérer plus de données mène à une meilleure estimation.

Voyons maintenant comment se déroule la simulation. Soit $x_s \in \mathcal{R}$ la donnée d'entrée non étiquetée pour laquelle on simule l'annotation et y_s l'étiquette attribuée, on note $\mathcal{T} \cup (x_s, y_s)$ la base d'entrainement résultante de cette simulation. L'étiquette que renverrais l'oracle pour l'échantillon simulé x_s serait tirée selon $P(y_s|x_s)$. Cette distribution étant inconnue, elle est ici estimée par $\hat{P}_{\theta(\mathcal{T})}(y_s|x_s)$. L'estimation de l'espérance du risque global sur la valeur de l'étiquette reçue consiste alors en une simulation pour chaque étiquette possible, suivies de la moyenne pondérée par $\hat{P}_{\theta(\mathcal{T})}(y_s|x_s)$ de chaque risque estimé ainsi obtenu. Alors,

$$\hat{R}_{m,\mathcal{R},\theta(\mathcal{T}\cup(x_s,.))}(f_{\theta(\mathcal{T}\cup(x_s,.))}) = \sum_{y_s \in Y} \hat{P}_{\theta(\mathcal{T})}(y=y_s|x_s)\hat{R}_{m,\mathcal{R},\theta(\mathcal{T}\cup(x_s,y_s))}(f_{\theta(\mathcal{T}\cup(x_s,y_s))}).$$

Finalement, la donnée d'entrée effectivement sélectionnée pour l'annotation sera celle qui est supposée mener à la plus grande décroissance du risque global :

$$x_{t+1} = \arg\min_{x_s \in \mathcal{R}} \hat{R}_{m,\mathcal{R},\theta(\mathcal{T}\cup(x_s,.))}(f_{\theta(\mathcal{T}\cup(x_s,.))}) - \hat{R}_{m,\mathcal{R},\theta(\mathcal{T})}(f_{\theta(\mathcal{T})}).$$

Notez que le deuxième terme de la différence ne dépend pas de la donnée d'entrée simulée et peut donc être négligé.

Outre la mesure de performance pouvant être utilisée, la différence entre [65] et [78] se situe dans le classifieur considéré. En effet, dans [65], le classifieur utilisé est le classifieur naïf de Bayes, tandis que dans [78], les auteurs utilisent un algorithme d'apprentissage semi-supervisé basé sur les champs aléatoires Gaussiens ainsi que sur les fonctions harmoniques.

Une deuxième approche est présentée dans [39], elle consiste à utiliser alternativement deux stratégies différentes. La première stratégie est basée sur la réduction de l'erreur tandis que la

deuxième est basée sur l'échantillonnage selon l'incertitude. Le principe est de calculer la décroissance du risque en attribuant à la donnée simulée l'étiquette prédite par le classifieur, puis, dans le cas où l'étiquette prédite n'était pas la bonne, de se rabattre sur la seconde stratégie pour un pas de temps.

Formellement, soit la stratégie $\mathbf{MM}$ celle basée sur la réduction de l'erreur. Le risque global pour une base d'entrainement donnée est toujours calculé en remplaçant directement le paramètre de la distribution de l'oracle par son estimation. Dans [39], le risque local est estimé par l'entropie de Shannon, et sa version globale utilise uniquement les données non-étiquetées. Soit x_s la données d'entrée pour laquelle on simule l'annotation, l'étiquette y_s qui lui est attribuée est celle qui minimise le risque global estimé, soit

$$y_s(x_s) = \arg\min_{y \in Y} \hat{R}_{m,\mathcal{R},\theta(\mathcal{T} \cup \{(x_s,y)\})}(f_{\theta(\mathcal{T} \cup \{(x_s,y)\})}).$$

Puis, la décroissance du risque estimé occasionnée par l'inclusion du couple $\{(x_s, y_s(x_s))\}$ est évaluée.

$$\Delta_{x_s,y_s(x_s)} \hat{R}_{m,\mathcal{R},\theta(\mathcal{T})} = \hat{R}_{m,\mathcal{R},\theta(\mathcal{T}^+)}(f_{\theta(\mathcal{T}^+)}) - \hat{R}_{m,\mathcal{R},\theta(\mathcal{T})}(f_{\theta(\mathcal{T})}),$$

avec $\mathcal{T}^+ = \mathcal{T} \cup \{(x_s, y_s(x_s))\}$. La donnée effectivement sélectionnée est celle maximisant la décroissance supposée :

$$x_{t+1}^{\mathbf{MM}} = \arg\min_{x_s \in \mathcal{U}} \Delta_{x_s,y_s(x_s)} \hat{R}_{m,\mathcal{R},\theta(\mathcal{T})}.$$

Soit une deuxième stratégie $\mathbf{M}$ basée sur l'échantillonnage selon l'incertitude. La donnée sélectionnée est simplement celle pour laquelle l'estimation du risque local est maximal.

$$x_{t+1}^{\mathbf{M}} = \arg\max_{x_s \in \mathcal{U}} \hat{r}_{m,\theta(\mathcal{T})}(f_{\theta(\mathcal{T})}, x_s).$$

La stratégie finale $\mathbf{MM} + \mathbf{M}$ consiste alors à employer par défaut la stratégie $\mathbf{MM}$, puis à chaque pas de temps de comparer l'étiquette prédite $y_s(x_{t+1})$ avec l'étiquette effectivement reçue de l'oracle y_{t+1}, et dans le cas où elles différent, d'employer la stratégie $\mathbf{M}$ pour la sélection suivante. Le déroulement de l'algorithme est décrit dans l'Algorithme 1.

De cette façon, cela permet de saisir l'occasion si l'annotation d'une donnée d'entrée pourrait dans le meilleur des cas mener à une forte décroissance du risque global, tout en prenant en compte que la prédiction peut être plus ou moins fiable. En effet, lorsque l'incertitude sur la prédiction est grande, à cause du faible nombre ou de la distance des données déjà étiquetées, alors la probabilité de prédire une étiquette erronée est forte, et la stratégie de sélection fera plus souvent appel à l'échantillonnage selon l'incertitude qui se concentre sur l'obtention d'une bonne qualité de prédiction. Ainsi, on peut s'imaginer qu'au début du processus, la stratégie $\mathbf{M}$ sera appelée à chaque fois que possible, puis au fur et à mesure que la connaissance de la distribution de l'oracle s'affine, cette stratégie sera laisser de côté au profit de la stratégie $\mathbf{MM}$. Il est intéressant de noter que les auteurs qualifient la stratégie MM d'optimiste, car considérant comme acquise l'étiquette prédite, tandis que la stratégie $\mathbf{MM} + \mathbf{M}$ partage des similarités avec le principe de l'optimisme face à l'incertitude que nous aborderons plus tard. En effet, l'idée est ici d'aborder l'incertitude sur les paramètres en combinant deux stratégies, une certaine de l'étiquette prédite et l'autre moins, tout en laissant la façon dont les deux se mêlent être gérée par un le taux de succès.

Un autre algorithme de réduction de l'erreur est introduit dans [46], celui-ci base sa stratégie de sélection sur une approche min-max du risque global engendré. Alors que dans l'algorithme précédent, l'estimation du risque global se faisait en remplaçant les paramètres de la distribution

Soumettre $x_1 \sim P(x)$ à l'oracle, recevoir y_1
$\mathcal{T} = (x_1, y_1)$
pour *t = 1, ..., n* **faire**
  suivant *Strat* **faire**
    cas où *MM* **faire**
      $\forall x_s \in \mathcal{U}$ calculer $y_s(x_s)$
      $\forall x_s \in \mathcal{U}$ calculer $\Delta_{x_s,y_s(x_s)}\hat{R}_{m,\mathcal{R},\theta(\mathcal{T})}$
      $x_{t+1} \leftarrow \arg\min_{x_s \in \mathcal{R}} \Delta_{x_s,y_s(x_s)}\hat{R}_{m,\mathcal{R},\theta(\mathcal{T})}$
    fin
    cas où *M* **faire**
      $\forall x_s \in \mathcal{U}$ calculer $\hat{r}_{m,\theta(\mathcal{T})}(f_{\theta(\mathcal{T})}, x_s)$
      $x_{t+1} \leftarrow \arg\max_{x_s \in \mathcal{R}} \hat{r}_{m,\theta(\mathcal{T})}(f_{\theta(\mathcal{T})}, x_s)$
      $Strat \leftarrow MM$
    fin
  fin
  Soumettre x_{t+1} à l'oracle, recevoir y_{t+1}
  $\mathcal{T} = \mathcal{T} \cup (x_{t+1}, y_{t+1})$
  si $Strat = MM \ \ \& \ \ y_s(x_{t+1}) \neq y_{t+1}$ **alors**
    $Strat \leftarrow M$
  fin
fin

Algorithme 1 : MM+M

de l'oracle correspondant à chaque donnée non-étiquetée par leur estimation, hormis pour la donnée simulée, ici toutes ces données non-étiquetée sont affectées de l'étiquette la plus probable. La donnée simulée quant à elle, est affectée de l'étiquette menant à la décroissance du risque global la moins bonne. Cela permet de garantir que, si cette donnée est effectivement sélectionnée, quelle que soit l'issue de l'annotation, la décroissance du risque qui s'en suit est au moins égale à celle préalablement considérée. Enfin, la stratégie de sélection consiste à choisir la donnée d'entrée pour laquelle cette pire décroissance du risque global est maximale.

Ainsi, en supposant que la donnée d'entrée simulée x_s soit annotée y_s, alors on attribue au reste des données non-étiquetées les étiquettes prédites par le classifieur

$$\{y_u(y_s), x_u \in \mathcal{U}\backslash\{x_s\}\} = \underset{\{y_u\} \in Y^{car(\mathcal{U})-1}}{\arg\min} \hat{R}_{m,\mathcal{R},\mathcal{T}^*}(f_{\theta(\mathcal{T}^*)}),$$

avec $\mathcal{T}^* = \mathcal{T} \cup \{(x_s, y_s)\} \cup \{(x_u, y_u), x_u \in \mathcal{U}\backslash\{x_s\}\}$ la base d'entrainement contenant toutes les données du réservoir incluant les données non étiquetées associées à l'étiquette qui leur a été attribuée, celle-ci varie donc en fonction de toutes les étiquettes attribuées artificiellement. Notez que l'estimation du risque global était auparavant indicée par $\theta(\mathcal{T})$ et ici uniquement $\mathcal{T}^*$, ceci est dû au fait que les paramètres de la distribution de l'oracle ne sont pas estimés mais simplement remplacés par l'étiquette reçue ou attribuée ($\hat{P}(y = y_r|x_r) = 1, \hat{P}(y \neq y_r|x_r) = 0$ avec y_r l'étiquette associée à une donnée x_r du réservoir).

Parmi les différentes étiquettes pouvant être renvoyées par l'oracle à la suite de l'annotation de x_s, celle menant à la pire décroissance du risque est considérée. Donc,

$$y_s = \underset{y \in Y}{\arg\max}\, \hat{R}_{m,\mathcal{R},\mathcal{T}^{**}}(f_{\theta(\mathcal{T}^{**})}),$$

avec $\mathcal{T}^{**} = \mathcal{T} \cup \{(x_s, y)\} \cup \{(x_u, y_u(y)), x_u \in \mathcal{U} \backslash \{x_s\}\}$ la base d'entrainement dans laquelle les étiquettes attribuées aux données non-étiquetées sont fonction de l'étiquette que l'on attribue à la donnée simulée, elle n'est alors soumise plus qu'a deux degrés de liberté : quelle donnée d'entrée est simulée et quelle étiquette lui est attribuée.

Enfin, la donnée d'entrée pour laquelle la pire décroissance est maximale est effectivement soumise à l'oracle. Ainsi,

$$x_{t+1} = \arg\min_{x_s \in \mathcal{U}} \hat{R}_{m,\mathcal{R},\mathcal{T}^+}(f_{\theta(\mathcal{T}^+)}),$$

avec $\mathcal{T}^+ = \mathcal{T} \cup \{(x_s, y_s)\} \cup \{(x_u, y_u(y_s)), x_u \in \mathcal{U} \backslash \{x_s\}\}$ la base d'entrainement dans laquelle l'étiquette attribuée à la donnée d'entrée simulée est fixée, elle n'est alors plus soumise qu'a un seul degré de liberté : quelle donnée d'entrée est simulée.

Ainsi, cette approche permet de sélectionner une donnée d'entrée avec prudence ne considérant que la pire issue à son annotation. Cela permet alors d'éviter de sélectionner une donnée selon une décroissance supposée et que celle-ci ne soit pas effective. L'inconvénient est que cette solution ne permet pas de considérer la probabilité d'apparition de chaque cas. En effet, dans la situation où, pour une donnée d'entrée, le pire cas est très peu probable, voire impossible, cette stratégie pourrait s'empêcher à tort de la sélectionner.

La stratégie de sélection étudiée dans [50] considère une approche Bayésienne du problème à l'aide des Processus Gaussiens. Le classifieur utilisé est alors un processus stochastique, cela veut dire qu'à chaque donnée d'entrée correspond non pas une prédiction unique, mais une distribution de probabilités sur la valeur de l'étiquette prédite. Ici, seule la classification binaire est considérée. Afin d'apprendre cette distribution relative au classifieur, la valeur du vrai paramètre de la distribution de l'oracle fait d'abord l'objet d'une inférence Bayésienne pour en tirer une distribution *a posteriori* sur à partir des observations, c'est à dire des associations "donnée d'entrée/étiquette" présentes dans la base d'apprentissage. Cette distribution *a posteriori* est ensuite utilisée pour générer un classifieur. Notez que ces deux distributions sont indépendantes, l'une étant uniquement utilisée pour définir l'autre, la distribution de la prédiction ne dépend pas de la vraie valeur du paramètre. L'idée est alors d'estimer le risque global en effectuant l'espérance sur la distribution de la prédiction et la distribution *a posteriori* du paramètre. Ainsi, la réduction de l'erreur engendrée par l'annotation d'une donnée d'entrée est estimée en calculant la différence entre l'espérance du risque avant et après prise en compte de la donnée simulée.

Dans le cadre des Processus Gaussien, la famille de distribution considérée est Gaussienne, elles sont donc caractérisées seulement par deux paramètres : leur moyenne et leur variance, soit $\hat{\mu}(x)$ et $\hat{\sigma}^2(x)$ ces paramètres *a posteriori*. La probabilité que le paramètre soit inférieur à $\frac{1}{2}$ est noté

$$p(x) = \mathbb{P}(\mu(x) \leq \frac{1}{2}) = \Phi(\frac{1}{2}, \hat{\mu}(x), \hat{\sigma}^2(x)),$$

avec Φ la fonction de répartition de la loi normale. Dans la suite, on allégera les notations en notant simplement $\mathcal{N}(x)$ à la place de $\mathcal{N}(\hat{\mu}(x), \hat{\sigma}^2(x))$.

Le classifieur est défini de la manière suivante : un score $f(x)$ est tiré aléatoirement en suivant la distribution *a posteriori* $\mathcal{N}(x)$ du paramètre, puis l'étiquette est prédite en comparant ce score au seuil $\frac{1}{2}$. Ainsi, la prédiction effectuée est égale à $\mathbb{1}_{f(x) \geq \frac{1}{2}}$. Notez quela probabilité que le classifieur prédise l'étiquette 0 est aussi égale à $p(x)$.

Ici, l'estimation du risque local est différente pour les données étiquetées et non-étiquetées.

Pour une donnée étiquetée, l'oracle est supposé renvoyer toujours la même étiquette :

$$\begin{aligned}\forall x_t \in \mathcal{T}, \hat{r}_m(f, x_t) &= \mathbb{E}_{f(x_t)\sim\mathcal{N}(x_t)}[\ell(\mathbb{1}_{f(x_t)\geq\frac{1}{2}}, y_t)] \\ &= p(x_t)\ell(0, y_t) + (1 - p(x_t))\ell(1, y_t)\end{aligned}$$

Pour une donnée non-étiquetée, l'étiquette inconnue est remplacée par sa distribution *a posteriori* :

$$\begin{aligned}\forall x_u \in \mathcal{U}, \hat{r}_m(f, x_u) &= \mathbb{E}_{f(x_u)\sim\mathcal{N}(x_u),\mu(x_u)\sim\mathcal{N}(x_u),y_u\sim\mathcal{B}(\mu(x_u))}[\ell(\mathbb{1}_{f(x_u)\geq\frac{1}{2}}, y_u)] \\ &= \mathbb{E}_{\mu(x_u)\sim\mathcal{N}(x_u)}[p(x_u)(1 - \mu(x_u))\ell(0, 1) + (1 - p(x_u))\mu(x_u)\ell(1, 0)],\end{aligned}$$

en considérant que $\ell(0, 0) = \ell(1, 1) = 0$. Ce qui donne

$$\forall x_u \in \mathcal{U}, \hat{r}_m(f, x_u) = p(x_u)(1 - p(x_u))\ell(0, 1) + (1 - p(x_u))p(x_u)\ell(1, 0).$$

Finalement, le risque global est estimé par :

$$\hat{R}_{m,\mathcal{R},\mathcal{T}}(f) = \sum_{x_t\mathcal{T}} \hat{r}_m(f, x_t) + \sum_{x_u\mathcal{U}} \hat{r}_m(f, x_u).$$

Afin d'estimer l'espérance sur l'étiquette reçue de la décroissance du risque, la vraie probabilité $\mu(x_s)$ que l'oracle renvoie l'étiquette 1 est estimée en effectuant l'espérance sur la distribution *a posteriori*. Ainsi, la simulation de décroissance du risque en attribuant l'étiquette 1 est pondérée par $1 - p(x_s)$ et celle attribuant l'étiquette 0 par $p(x_s)$.

$$\Delta_{x_s}\hat{R}_{m,\mathcal{R},\mathcal{T}}(f) = (1 - p(x_s))\hat{R}_{m,\mathcal{R},\mathcal{T}^+}(f) + p(x_s)\hat{R}_{m,\mathcal{R},\mathcal{T}^-}(f) - \hat{R}_{m,\mathcal{R},\mathcal{T}}(f),$$

avec $\mathcal{T}^+ = \mathcal{T} \cup \{(x_s, 1)\}$ et $\mathcal{T}^- = \mathcal{T} \cup \{(x_s, 0)\}$.

La donnée d'entrée choisie par l'algorithme pour être soumise à l'oracle est donc celle maximisant la décroissance. Ainsi,

$$x_{t+1} = \arg\min_{x_s\in\mathcal{U}} \Delta_{x_s}\hat{R}_{m,\mathcal{R},\mathcal{T}}(f).$$

L'intérêt de cette méthode par rapport aux méthodes précédentes est qu'elle prend en compte non seulement une estimation du paramètre mais aussi la confiance en cette estimation. En effet, en utilisant une approche Bayésienne, la variance de la distribution *a posteriori* nous donne une mesure de cette confiance, si la variance est élevée alors il existe une grande plage de valeurs du paramètre hautement probables, et lui attribuer une valeur unique a de faibles chances d'être juste. Ici, la variance pour une donnée d'entrée spécifique dépend uniquement du nombre et de la distance des données d'entrée considérée pour l'estimation. Dans cette approche, effectuer l'espérance sur la distribution *a posteriori* permet de considérer tout les cas possibles du paramètre en même temps, plutôt que de le remplacer par la valeurs la plus ou la moins probable.

Cependant, plusieurs remarques peuvent être faites. Ici, le classifieur considéré est stochastique, pourtant la stratégie de classification optimale est déterministe et doit baser sa prédiction sur la valeur la plus probable de l'arrondi du paramètre, même lorsque celui-ci fait l'objet d'une distribution *a posteriori*. Il serait donc préférable de considérer un classifieur déterministe, en considérant toujours la distribution *a posteriori* du paramètre pour l'estimation du risque. Ensuite, les étiquettes déjà reçues ne sont qu'un échantillon de la distribution de l'oracle, ici elles sont considérées comme des valeurs de référence pour le classifieur. Elles pourraient cependant être considérées au même titre que les données non-étiquetées dans l'estimation du risque. Enfin, la décroissance du risque

global résulte de la différence entre le risque estimé avant et après ajout de la données simulée. De ce fait, le risque antérieur est estimé en effectuant l'espérance sur la distribution *a posteriori* apprise sur la base d'entrainement initiale. Une meilleure solution serait de considérer la décroissance du risque dans sa globalité, en calculant l'espérance de la décroissance du risque réel sur la distribution *a posteriori* apprise sur la base d'entrainement résultant de la simulation. Notez toutefois que dans ce cas, la distribution *a posteriori* initiale servirai toujours à contrôler le poids affecté à chaque étiquette de la simulation.

Dans la suite, nous présentons les méthodes introduites dans [36] et [35] qui partagent la même approche, l'idée est de calculer une borne supérieure sur le risque global et de trouver les données d'entrée qui engendrent la borne minimum. Ici, la stratégie de sélection est définie dans le but d'améliorer la régression du paramètre de la distribution de l'oracle. En effet, étant donné que la prédiction optimale dépend du vrai paramètre, alors une estimation précise de ce paramètre engendre une prédiction de qualité. Il est alors envisagé d'utiliser le budget de données à étiqueter dans le but d'obtenir une estimation précise du paramètre, puis de se servir de ce paramètre pour la prédiction. Bien que cette méthode ne soit pas optimale, en particulier car la précision de l'estimation impacte la prédiction différemment en fonction de la vraie valeur du paramètre, elle partage le but commun de l'amélioration de la prédiction.

De plus, la première de ces méthodes [36] se place dans le cadre de l'apprentissage actif d'une traite. Rappelons que ce scénario consiste à sélectionner préalablement l'ensemble des données d'entrée qui seront soumises à l'oracle. Ainsi, aucune observation n'étant disponible initialement, le choix de ces données ne peut pas se baser sur une quelconque estimation courante du paramètre mais uniquement sur la répartition des données non-étiquetées sur l'espace d'entrée. En conséquence, la stratégie d'apprentissage actif n'est pas pénalisée par sa restriction au problème de régression du paramètre. En effet, il est alors impossible de savoir sur quelle partie de l'espace d'entrée une meilleur précision de l'estimation serait plus rentable, car aucune information n'est disponible sur la valeur du vrai paramètre.

Le but est alors de calculer une borne sur le risque global ne prenant en compte que la répartition des données d'entrée incluses dans la base d'apprentissage. Ici, l'objectif étant d'effectuer une régression du paramètre de l'oracle, la fonction de prédiction utilisée est apprise par moindres carrés régularisés Laplaciens. Ceci étant la version d'apprentissage semi-supervisée des moindres carrés classiques, elle permet de prendre en compte la distribution des données d'entrée. La fonction de prédiction est alors :

$$\forall x \in \mathbb{R}^d, \quad f_{\boldsymbol{\theta}}(x) = x^t \boldsymbol{\theta},$$

avec x une donnée d'entrée, d la dimension de l'espace d'entrée, et $\boldsymbol{\theta} \in \mathbb{R}^d$ un vecteur de paramètres. La fonction objectif utilisée est alors :

$$\boldsymbol{\theta}_{\mathcal{T}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\arg\min} ||\mathbf{X}_{\mathcal{T}}^t \boldsymbol{\theta} - \mathbf{y}_{\mathcal{T}}||_2^2 + \frac{\lambda_A}{2}||\boldsymbol{\theta}||_2^2 + \frac{\lambda_I}{2}\boldsymbol{\theta}^t \mathbf{X}_{\mathcal{T}} \mathbf{L}_{\mathcal{T}} \mathbf{X}_{\mathcal{T}}^t \boldsymbol{\theta},$$

avec $(\lambda_A, \lambda_I) \in \mathbb{R}_+^{*2}$ des paramètres de régularisation, $||.||_2$ la norme ℓ_2, $\mathbf{X}_{\mathcal{T}}$ la matrice dont les colonnes sont les données d'entrée incluses dans la base d'apprentissage, $\mathbf{y}_{\mathcal{T}}$ le vecteur contenant les étiquettes reçues pour les données dans la base d'apprentissage et $\mathbf{L}_{\mathcal{T}} = \mathbf{D}_{\mathcal{T}} - \mathbf{W}_{\mathcal{T}}$ où $\mathbf{W}_{\mathcal{T}}$ est une matrice de similarité entre chaque donnée de la base d'apprentissage et $\mathbf{D}_{\mathcal{T}}$ est une matrice diagonale telle que $D_{ii} = \sum_{j=1}^{n} W_{ij}$ avec n le nombre de données dans la base d'apprentissage. On voit que les termes de cette fonction objectif sont respectivement : une mesure de la performance, un terme de régularisation permettant d'éviter de complexifier le modèle, un terme de régularisation permettant de lisser la fonction de prédiction.

Ainsi, le vecteur de paramètre peut être calculé directement :

$$\boldsymbol{\theta}_{\mathcal{T}} = (\mathbf{X}_{\mathcal{T}}\mathbf{X}_{\mathcal{T}}^t + \lambda_A \mathbf{I}_n + \lambda_I \mathbf{X}_{\mathcal{T}}\mathbf{L}_{\mathcal{T}}\mathbf{X}_{\mathcal{T}}^t)\mathbf{X}_{\mathcal{T}}\mathbf{y}_{\mathcal{T}},$$

avec $\mathbf{I}$ la matrice identité de taille n.

Alors, l'estimation du risque global quadratique peut s'exprimer comme

$$\hat{R}_{2,\mathcal{R},\mathcal{T}}(f_{\boldsymbol{\theta}_{\mathcal{T}}}) = \mathbb{E}_{\mathbf{y}_{\mathcal{T}} \sim \mu(\mathbf{X}_{\mathcal{T}})}[||\mathbf{R}^t\boldsymbol{\theta}_{\mathcal{T}} - \mathbf{R}^t\boldsymbol{\theta}^*||_2^2],$$

avec $\mathbf{R}$ la matrice dont les colonnes sont les données d'entrée du réservoir, comprenant les données non-étiquetées et étiquetées, et $\boldsymbol{\theta}^*$ est le vecteur de paramètres optimaux connaissant parfaitement la distribution de l'oracle. Notez qu'ici l'estimation ne porte que sur la distribution naturelle des données d'entrée et non sur la distribution de l'oracle. Dans [36], il est montré que ce risque peut être borné par

$$\hat{R}_{2,\mathcal{R},\mathcal{T}}(f_{\boldsymbol{\theta}_{\mathcal{T}}}) \leq (B + \sigma^2) tr(\mathbf{R}^t(\mathbf{X}_{\mathcal{T}}\mathbf{X}_{\mathcal{T}}^t + \lambda_A \mathbf{I}_n + \lambda_I \mathbf{X}_{\mathcal{T}}\mathbf{L}_{\mathcal{T}}\mathbf{X}_{\mathcal{T}}^t)^{-1}\mathbf{R}),$$

avec $tr(.)$ la trace, B est une constante telle que $\lambda_A||\boldsymbol{\theta}^*||_2^2 + \lambda_I \boldsymbol{\theta}^{*t}\mathbf{X}_{\mathcal{T}}\mathbf{L}_{\mathcal{T}}\mathbf{X}_{\mathcal{T}}^t\boldsymbol{\theta}^* \leq B$ et σ^2 la variance maximale de la distribution de l'oracle.

La stratégie de sélection adoptée consiste donc à choisir le sous ensemble de données minimisant la borne sur le risque de sorte que cela minimise aussi le risque global réel. Ainsi,

$$x_{t+1} = \underset{\mathbf{X}_{\mathcal{T}} \in X^n}{\arg\min}\, tr(\mathbf{R}^t(\mathbf{X}_{\mathcal{T}}\mathbf{X}_{\mathcal{T}}^t + \lambda_A \mathbf{I}_n + \lambda_I \mathbf{X}_{\mathcal{T}}\mathbf{L}_{\mathcal{T}}\mathbf{X}_{\mathcal{T}}^t)^{-1}\mathbf{R}).$$

Notez que cette stratégie de sélection ne se base que sur la répartition des données d'entrées. Ce choix est ensuite un problème d'optimisation combinatoire, où le but est de trouver l'ensemble de données minimisant ce critère parmi celles disponibles. Ceci n'étant pas l'objet de notre étude, nous ne détaillerons pas cet aspect.

La deuxième méthode [35] découle de la précédente en l'adaptant au scénario d'échantillonnage par paquets. Dans ce cas, à chaque pas de temps, la décision du prochain paquet à soumettre à l'oracle peut se baser sur sa réponse pour les paquets précédents. Ici, chaque paquet est de taille fixe et contient un nombre b de données d'entrée. Le classifieur utilisé est la régression logistique, où le score servant à la prédiction peut s'exprimer comme

$$\forall x \in \mathbb{R}^d, \quad f_{\boldsymbol{\theta}}(x) = \sigma(x^t\boldsymbol{\theta}),$$

avec $\sigma(.)$ la fonction sigmoïde telle que $\forall z \in \mathbb{R}, \sigma(z) = \frac{1}{1+\exp^{-z}}$, x une donnée d'entrée et $\boldsymbol{\theta}$ le vecteur de paramètre utilisé. Les paramètres sont appris afin de maximiser la vraisemblance du score lorsque utilisé pour générer les étiquettes connues. Ainsi, ces paramètres vérifient

$$\boldsymbol{\theta}_{\mathcal{T}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\arg\min}\, \lambda||\boldsymbol{\theta}||_2^2 - \frac{1}{n}\sum_{i=1}^{n} \log(\sigma(y_i x_i^t \boldsymbol{\theta})),$$

avec $\lambda \in \mathbb{R}_+^*$ un paramètre de régularisation et $n = card(\mathcal{T})$ le nombre courant de données dans la base d'apprentissage.

Soit $\boldsymbol{\theta}^*$ les paramètre optimaux telles que $f_{\boldsymbol{\theta}^*}(x) = \mu(x)$. Alors, le risque global quadratique est estimé par

$$\hat{R}_{2,\mathcal{R},\mathcal{T}}(f_{\boldsymbol{\theta}_{\mathcal{T}}}) = \mathbb{E}_{\mathbf{y}_{\mathcal{T}} \sim \mu(\mathbf{X}_{\mathcal{T}})}[||f_{\boldsymbol{\theta}_{\mathcal{T}}}(\mathbf{X}_{\mathcal{T}}) - f_{\boldsymbol{\theta}^*}(\mathbf{X}_{\mathcal{T}})||_2^2],$$

en utilisant les même notations que précédemment pour $||.||_2$, $\mathbf{X}_{\mathcal{T}}$, $\mathbf{y}_{\mathcal{T}}$ et $\mathbf{R}$.

Dans [35], il est montré que ce risque peut être borné par

$$\hat{R}_{2,\mathcal{R},\mathcal{T}}(f_{\boldsymbol{\theta}_\mathcal{T}}) \leq Ctr(\boldsymbol{\Sigma}_\mathcal{T}^{\frac{1}{2}}\mathbf{R}_\mathcal{T}^t(\lambda\mathbf{I}_n + \frac{1}{n}\mathbf{X}_\mathcal{T}\mathbf{D}_\mathcal{T}\mathbf{X}_\mathcal{T}^t)^{-1}\mathbf{R}_\mathcal{T}\boldsymbol{\Sigma}_\mathcal{T}^{\frac{1}{2}}),$$

avec $tr(.)$ la trace, C une constante, $\mathbf{D}_\mathcal{T}$ et $\boldsymbol{\Sigma}_\mathcal{T}$ les matrice diagonale contenant respectivement la variance réelle et estimée de la distribution de l'oracle évaluée sur les données contenues dans la base d'apprentissage.

Étant donné que $\mathbf{D}_\mathcal{T}$ est inconnue, elle est remplacée par son estimation $\boldsymbol{\Sigma}_\mathcal{T}$. Puis, la stratégie de sélection consiste à choisir le paquet de données minimisant cette borne sur le risque :

$$x_{t+1} = \underset{(x_i)_{i\in[\![1,b]\!]}\in X^b}{\arg\min}\ tr(\boldsymbol{\Sigma}_{\mathcal{T}^+}^{\frac{1}{2}}\mathbf{R}^t(\lambda\mathbf{I}_{n+b} + \mathbf{X}_{\mathcal{T}^+}\boldsymbol{\Sigma}_{\mathcal{T}^+}\mathbf{X}_{\mathcal{T}^+}^t)^{-1}\mathbf{R}\boldsymbol{\Sigma}_{\mathcal{T}^+}^{\frac{1}{2}}),$$

avec $\mathcal{T}^+ = \mathcal{T}\cup\{x_i\}_{i\in[\![1,b]\!]}$ l'ensemble de données étiquetées après ajout du paquet considéré. Notez que la matrice $\boldsymbol{\Sigma}_{\mathcal{T}^+}$ ne dépend pas du choix du paquet, uniquement l'estimation courante de la variance est utilisée, l'indexation par $\mathcal{T}^+$ ne sert qu'à représenter les données d'entrée sur lesquels elle est évaluée. De même que précédemment, trouver l'ensemble de données minimisant ce critère est un problème d'optimisation combinatoire, pour plus de détail sur la solution considérée nous invitons le lecteur à lire la publication originale [35].

Finalement, cette méthode permet d'utiliser la l'idée développée dans la méthode précédente en prenant en compte à chaque pas de temps l'ensemble des étiquettes reçues à travers l'estimation de la variance contenue dans la matrice $\boldsymbol{\Sigma}_\mathcal{T}$. Notez que cette matrice joue la même rôle que le paramètre σ^2 de la méthode précédente, sauf que celle-ci varie avec la donnée d'entrée et s'adapte au cours du processus. Rappelons qu'ici, l'objectif est d'obtenir le plus rapidement possible la meilleure régression du paramètre de la distribution de l'oracle, ce qui est différent d'obtenir la meilleur prédiction. Cette méthode est adaptable aux scénarios d'apprentissage entièrement séquentiel et d'une traite en faisant varier b, la taille du paquet.

Les deux méthodes que nous venons de voir basent leur stratégie de sélection sur une borne supérieure du risque global résultant de la simulation. En tentant d'atteindre la borne minimum avec le budget de données étiquetées considéré, le risque global réel est garanti d'être inférieur à ce minimum. De plus, il est important que la borne possède le même comportement que le risque réel, afin qu'un effet sur la borne implique un effet sur le risque. C'est le cas ici puisque cette borne est étroite car pouvant être atteinte. D'autre part, notez que le risque réel devant être minimisé, utiliser une borne supérieure revient à considérer le pire cas vis-à-vis de l'incertitude liée au paramètre de l'oracle.

En conclusion, dans cette section nous avons présenté plusieurs stratégies de sélection des données d'entrée utilisant le principe de la réduction de l'erreur. Toutes partagent l'objectif commun de minimiser un critère idéal : le risque réel global. Cependant, ce critère n'est pas utilisable en pratique car il dépend de paramètre inconnus, à savoir les paramètres de la distribution de l'oracle. Ceux-ci peuvent toutefois être estimés à travers des observations provenant de l'échantillonnage de cette distribution que sont les étiquettes reçues. Le nombre d'observations étant fini, cette estimation n'est pas parfaite et il subsiste une incertitude quant à ces paramètres. Les méthodes se différencient donc par la façon de gérer cette incertitude. Nous avons vu plusieurs solutions, dont le remplacement direct par l'estimation, la confiance dans le meilleur cas, la prudence quant au pire cas (min-max, borne supérieure) et l'espérance sur la distribution *a posteriori* dans le cas Bayésien. En outre, la confiance dans le meilleur cas n'offre pas une solution viable lorsqu'elle est utilisée seule et doit être modifiée pour inclure une certaine prudence par exemple en alternant avec une autre stratégie. Une partie de nos travaux viendrons s'inscrire dans ce contexte et offriront un

nouvelle solution pour gérer l'incertitude sur les paramètres qui permettra de définir le niveau de confiance dans le meilleur cas à l'aide d'un unique paramètre.

3.6 Réduction de l'espace des versions

Dans les sections précédentes, nous avons vu qu'une façon de traiter le problème d'apprentissage actif était de prendre en considération la performance atteinte par le classifieur pour tenter de la maximiser. Ainsi, à chaque pas de temps, ou lors de chaque simulation, un seul jeu de paramètres du classifieur était utilisé, celui-ci étant appris pour correspondre à la performance maximale atteignable avec les données disponibles. Le choix de la prochaine données d'entrée à soumettre à l'oracle était alors effectué pour que ces paramètres appris sur la base d'entrainement résultante mènent à la performance maximale. Dans cette section, nous présentons une autre approche usuelle de l'apprentissage actif qui consiste à pouvoir déterminer le plus rapidement possible les paramètres idéaux. Peu importe la performance absolue de ces paramètres puisqu'aucune autre version du classifieur ne peut avoir une performance supérieure. Pour cela, l'idée est de considérer l'ensemble des classifieurs idéaux potentiels, appelé espace des versions, et de réduire le plus rapidement possible son volume, analogue à un nombre de classifieurs pour des paramètres continus, afin que la révélation du classifieur idéal puisse être effectuée le plus tôt possible. Le choix des données d'entrée doit donc se faire dans le but de discréditer un maximum de candidats au jeu de paramètres idéal. Étant donné que l'étiquette attribuée à la donnée d'entrée sélectionnée n'est pas connue par avance, l'estimation du nombre de données ainsi discréditées doit prendre en compte toutes les issues possibles.

Dans la littérature existante, cette approche a été étudiée sous deux aspects, selon que le problème de classification considéré soit bruité ou non. En effet, le choix des classifieurs à inclure dans l'espace des version dépend fortement de cette considération. Cette section se divise donc en deux parties, nous présentons premièrement les méthodes développées pour le problème non bruité, ce qui nous permettra d'appréhender plus facilement les concepts et les solutions mis en place. Dans un deuxième temps, nous décrirons différentes manières dont cette approche a pu être adaptée aux problèmes bruités.

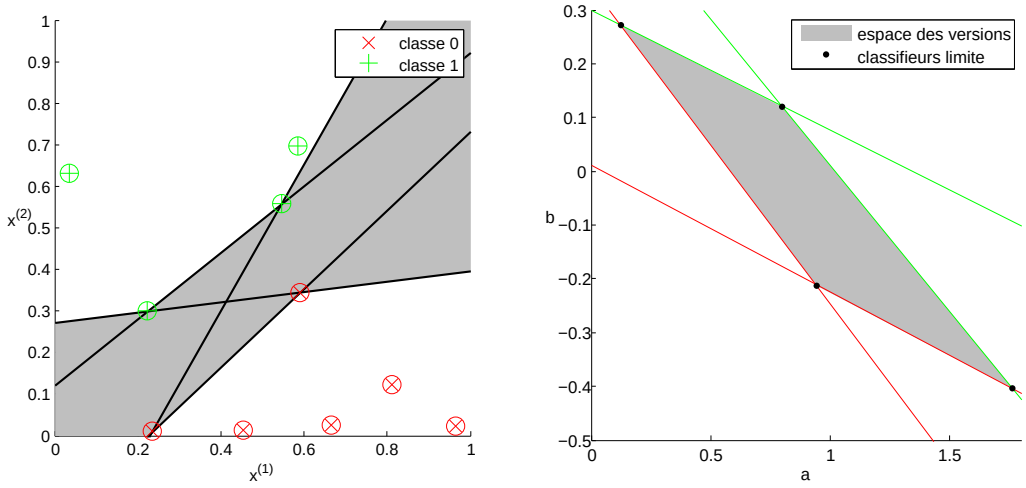
Avant de décrire plus en détails les stratégies de sélections existantes, nous commençons par donner des généralités sur le problème de réduction de l'espace des versions dans le cas non-bruité. Ceci nous permet définir le cadre commun sur lequel sont construits les différents algorithmes, qui partagent le même principe de fonctionnement, pour ensuite lister les points sur lesquels s'appuient leurs variations. Tout d'abord, soit $\mathcal{T} = \{(x_t, y_t)\}_{t\in[\![1,n]\!]}$ la base d'apprentissage courante contenant l'ensemble des données d'entrée soumises à l'oracle et leur étiquette associée. Soit $\mathcal{H}$ l'espace d'hypothèse, c'est à dire l'ensemble des classifieurs initialement disponibles, on suppose ici qu'il contient le classifieur idéal f^*, c'est à dire le classifieur qui prédit parfaitement la réponse de l'oracle :

$$\forall x \in X, f^*(x) = P(y = 1|x),$$

en rappelant que dans le cas de la classification non-bruitée $P(y = 1|x) \in \{0, 1\}$. Dans la suite, si le classifieur est paramétré par θ, on utilisera abusivement $\mathcal{H}$ pour parler de l'espace des paramètres. Le classifieur idéal ayant une prédiction parfaite sur n'importe quelle donnée d'entrée, cela inclus les données de la base d'apprentissage dont on connait le résultat. Un classifieur est dit cohérent avec la base d'apprentissage si sa prédiction sur les données annotées est juste. Tout classifieur n'étant pas cohérent avec la base d'apprentissage n'est obligatoirement pas le classifieur idéal. Ainsi, l'espace des version $\mathcal{V}$ est dans le cas non-bruité, l'ensemble des classifieurs cohérents avec la base d'apprentissage

$$\mathcal{V} = \{f \in \mathcal{H}, \forall t \in [\![1, n]\!], f(x_t) = y_t\}.$$

De même qu'avec $\mathcal{H}$, on emploiera abusivement la notion d'espace des versions pour parler de



(a) Espace d'entrée (b) Espace des hypothèses

FIGURE 3.1 – Dualité espace d'entrée / espace des hypothèses

l'espace des paramètres définissant un classifieur cohérent avec la base d'apprentissage.

Le but est maintenant de sélectionner la prochaine donnée d'entrée à annoter dans le but de minimiser le plus rapidement possible le volume de l'espace des versions. Pour cela, il s'agit de définir une mesure du désaccord entre les classifieurs contenus dans l'espace des versions sur l'étiquette à prédire pour une donnée d'entrée fixée. L'annotation d'une donnée d'entrée permettra de déterminer quels sont les classifieurs cohérents et donc de rejeter tous les autres de l'espace des versions. En fonction de l'étiquette reçue le nombre de classifieurs ainsi rejetés peut varier. Celle-ci étant initialement inconnue, la solution est de sélectionner une donnée d'entrée étant sujette à un grand désaccord, c'est à dire qu'un grand nombre de classifieurs font des prédictions divergentes. Ainsi, cela permet de maximiser le nombre minimum de classifieurs rejetés selon l'issue de l'annotation. Les différents algorithmes présentés ci-après se distinguent par la façon dont le désaccord est mesuré.

La Figure 3.1 illustre la notion d'espace des versions en utilisant l'exemple d'un classifieur binaire linéaire dans un espace en deux dimensions. Le classifieur est alors tel que $\forall x \in X, f_{a,b}(x) = x^{(2)} - ax^{(1)} - b$ avec $(a, b) \in \mathbb{R}^2$ ses deux paramètres et $x^{(i)}$ le $i^{\text{ème}}$ attribut de x. La Figure 3.1(a) représente l'espace d'entrée où apparaissent les données étiquetées, les données appartenant à la classe 1 sont représentées par un $+$ et de couleur verte, celles appartenant à la classe 0 sont représentées par un $\times$ et de couleur rouge. Ces données contraignent l'ensemble des classifieurs possibles dont le séparateur doit passer uniquement par la zone grise. Les séparateurs des classifieurs limite sont représentés en noir sur la figure. Notez que dans le cas étudié ici, seul un certain nombre de données d'entrée définissent réellement l'ensemble des classifieurs cohérents. La Figure 3.1(b) affiche une représentation de l'espace des hypothèses où les coordonnées sont les paramètres des classifieurs. Chaque classifieur est donc représenté par un point dans cet espace et l'ensemble des classifieurs cohérents définit une zone ici représentée en gris. Il est intéressant de noter qu'un séparateur dans l'espace d'entrée correspond à un point dans l'espace des hypothèses et qu'inversement un point dans l'espace d'entrée correspond à une droite dans l'espace des hypothèses. En effet, une seule donnée d'entrée contraint à elle seule un ensemble de classifieurs. Ici, les données d'entrée délimitant l'espace des versions sont représentées dans la couleur correspondant à leur étiquette. Ceci permet de comprendre qu'annoter une nouvelle donnée d'entrée permet de couper en deux

l'espace des versions. Afin de réduire le plus vite possible le volume de l'espace des versions sans connaitre l'étiquette d'une donnée d'entrée, il faut choisir celle qui le coupe en deux parties égales.

Pour pouvoir choisir la prochaine donnée d'entrée à soumettre à l'oracle il est nécessaire de pourvoir mesurer le volume de l'espace des versions ainsi que les volumes de chacune des parties que sépare la donnée d'entrée considérée. Cependant, pour la plupart des classifieurs, la géométrie de l'espace des versions est complexe, et le calcul de ces volumes n'est pas trivial. La solution introduite dans [68] est la *requête par comité*, elle consiste à entrainer un nombre fini de classifieurs appelé comité qui sont disséminés sur l'ensemble de l'espace des versions. De cette façon, le volume de chaque partie est estimé en comptant simplement le nombre de classifieurs du comité prédisant chacune des étiquettes pour la donnée d'entrée considérée. Ceci revient à estimer le volume par *Monte-Carlo*. Ceci possède de plus l'avantage d'inclure naturellement dans la mesure du volume la distribution *a posteriori* des paramètres sur l'espace des versions. En effet, réduire le volume de l'espace des versions ne consiste pas simplement à délimiter la plus petite zone contenant des classifieurs cohérents. Même parmi l'ensemble des classifieurs cohérents, il est plus bénéfique d'éliminer ceux que l'on pense plus probable d'être le classifieur idéal plutôt que ceux qui n'auraient de toute façon pas été envisagés. Ainsi, dans [68], les classifieurs sont générés en utilisant l'algorithme de Gibbs, dans [60], les paramètres des classifieurs du comité sont tirés selon une distribution de Dirichlet, et dans [25], les classifieurs sont générés à partir d'une chaine de Markov cachée. D'autre part, dans [61], un unique comité de classifieurs est généré au début du processus, lequel contient le classifieur idéal, puis à chaque pas de temps les classifieurs incohérents sont éliminés. En conséquence, uniquement la distribution *a priori* des classifieurs est utilisée, sans qu'un classifieur puisse être plus probable qu'un autre. Ceci est utile lorsque les classifieurs sont non-probabilistes ou sont de nature différentes. De cette façon, le nombre de données à sélectionner pour déterminer le meilleur classifieur varie en $O(\log_2(card(\mathcal{H})))$. Ce dernier papier fait le parallèle entre la réduction de l'espace des versions et une dichotomie généralisée à un espace de dimensions supérieure à deux.

Une fois qu'un ensemble de classifieurs permettant une mesure du volume de l'espace des versions a été sélectionné, il reste à définir comment combiner les prédiction de chaque classifieur dans le but de choisir la donnée d'entrée la plus adaptée. Ceci prend usuellement la forme d'une mesure de désaccord, étant donné que le but est de sélectionner la donnée d'entrée permettant au maximum de séparer les classifieurs en deux groupes distincts. Soit $V_{\mathcal{H},\mathcal{T}}$ le volume de l'espace des versions courant, et $V^{+}_{\mathcal{H},\mathcal{T}}(x)$, resp. $V^{-}_{\mathcal{H},\mathcal{T}}(x)$ le volume de l'espace des versions après que la donnée d'entrée $x \in X$ ait été annotée avec l'étiquette 1, resp. 0. Dans [68] et [25], l'idée est de sélectionner la donnée d'entrée pour laquelle le gain d'entropie de Shannon est le plus élevé. Ainsi,

$$x_{t+1} = \arg\max_{\mathcal{U}} -\frac{V^{+}_{\mathcal{H},\mathcal{T}}(x)}{V_{\mathcal{H},\mathcal{T}}} \log(\frac{V^{+}_{\mathcal{H},\mathcal{T}}(x)}{V_{\mathcal{H},\mathcal{T}}}) - \frac{V^{-}_{\mathcal{H},\mathcal{T}}(x)}{V_{\mathcal{H},\mathcal{T}}} \log(\frac{V^{-}_{\mathcal{H},\mathcal{T}}(x)}{V_{\mathcal{H},\mathcal{T}}}).$$

Dans [61], la mesure du désaccord est simplement le nombre de classifieurs ayant des prédictions différentes.

$$x_{t+1} = \arg\min_{\mathcal{U}} |V^{+}_{\mathcal{H},\mathcal{T}}(x) - V^{-}_{\mathcal{H},\mathcal{T}}(x)|,$$

avec $|.|$ la valeur absolue. Ces méthodes sont à analogues à l'échantillonnage d'incertitude dans le cas où le paramètre de la distribution de l'oracle est estimé avec la proportion des classifieurs du comité prédisant l'étiquette 1. Ici, seule la prédiction finale de chaque classifieur est prise en compte, cependant, il est possible que les classifieurs du comité soit probabilistes et que les scores puissent fournir une mesure de certitude sur l'étiquette a attribuer. Dans [60], la mesure du désaccord prend en compte cette certitude à travers la divergence de Kullback-Leibler. Soit $\hat{P}_f(y|x)$ les

scores donnés par le classifieur f pour la donnée d'entrée x. La divergence de Kullback-Leibler entre deux classifieurs f_1 et f_2 est alors :

$$D(\hat{P}_{f_1}(y|x)||\hat{P}_{f_2}(y|x)) = \sum_{i \in Y} \hat{P}_{f_1}(y=i|x) \log(\frac{\hat{P}_{f_1}(y=i|x)}{\hat{P}_{f_2}(y=i|x)}).$$

Soit $\hat{P}_{avg}(y|x) = \frac{1}{card(\mathcal{C})} \sum_{f \in \mathcal{C}} \hat{P}_f(y|x)$ le score moyen sur le comité $\mathcal{C}$. Alors la sélection de la donnée d'entrée se base sur la divergence moyenne sur le comité :

$$x_{t+1} = \arg\max_{\mathcal{U}} \frac{1}{card(\mathcal{C})} \sum_{f \in \mathcal{C}} D(\hat{P}_f(y|x)||\hat{P}_{avg}(y|x)).$$

Une autre façon de mesurer le volume de l'espace des versions à été développée dans [74] où le classifieur étudié est une machine à vecteurs supports. Les auteurs montrent que, pour les machines à vecteurs support linéaires, la marge peut être représentée par une sphère dans l'espace des versions. Ainsi, le problème de trouver le séparateur linéaire qui possède la plus grande marge revient à trouver la plus grande sphère totalement contenue dans l'espace des versions. De cette façon, ils choisissent d'estimer le volume de l'espace des versions en utilisant la marge. De plus, ils montrent que l'échantillonnage d'incertitude, qui sélectionne une donnée d'entrée au plus proche du séparateur, ne permet pas de couper l'espace des versions en deux parties égales. En effet, le centre de la plus grande sphère n'est pas nécessairement proche du barycentre de l'espace des versions, et la contrainte sur la donnée d'entrée à sélectionner, qui revient à faire passer la droite correspondante par le centre, n'a aucune garantie de couper l'espace des versions en deux. De ce fait, une nouvelle stratégie est introduite, qui consiste, pour chaque donnée d'entrée, à calculer les deux plus grandes sphères, de chaque côté de la droite, simplement en entrainant le classifieur pour les deux issues de l'annotation pour en récupérer la marge, puis à sélectionner la donnée d'entrée pour laquelle la plus petite des deux sphères est la plus grande.

Nous venons de voir un ensemble de méthodes qui traitent le problème d'apprentissage actif par la réduction de l'espace des version. En divisant à chaque pas de temps sa taille en deux, ces méthodes parviennent à sélectionner le classifieur idéal avec un nombre d'annotations optimal. Cependant, elles ne s'appliquent que sur un problème de classification non bruité, en particulier car elles s'appuient sur la notion de classifieur cohérent. Une telle hypothèse limite fortement l'utilisation de ces méthodes car les problèmes rencontrés en pratique sont souvent bruités. Le problème est alors de savoir si cette approche peut être adaptée à des problèmes de classification bruitée. Dans ce cas, la notion de classifieur cohérent comme définie précédemment n'a pas de sens car un échantillon apportant une étiquette contraire à la prédiction du classifieur ne le discrédite pas pour autant, étant donné qu'il peut s'agir uniquement d'un mauvais tirage. Toutefois, il est possible de redéfinir la notion de classifieur cohérent et donc d'espace des versions en utilisant la distribution *a posteriori* sur les classifieurs. Le but de la stratégie de sélection étant alors de centrer la distribution *a posteriori* sur le classifieur idéal. Il est aussi nécessaire de définir une nouvelle mesure du volume afin de savoir quel est l'apport d'un échantillon, de même que la mesure du désaccord sur laquelle se base la stratégie de sélection.

La première méthode, développée dans [62], reprend les travaux de [61] et les étend au cas bruité. Alors que dans cette précédente méthode, tout les classifieurs étaient également considérés sous réserve d'être contenus dans l'espace des versions qui, lui, évoluait à chaque pas de temps, ici l'espace des versions contient en permanence l'ensemble des classifieurs initiaux. Cependant, il intervient une distribution *a posteriori* sur les classifieurs qui permet de sélectionner le classifieur supposé idéal. La stratégie de sélection tente donc de réduire cette distribution à un Dirac en sélectionnant une donnée d'entrée qui permet de différencier les classifieurs.

Ainsi, soit $\mathcal{H}$ un espace d'hypothèses contenant un nombre fini de classifieurs. Soit une mesure de probabilité $p \in [0,1]^{\mathcal{H}}$ sur $\mathcal{H}$ calculée de façon Bayésienne dont la mise à jour se fait après chaque nouvelle annotation de la façon suivante. Soit, au pas de temps t, l'annotation d'une donnée d'entrée $x_t \in X$ par l'étiquette $y_t \in Y$. L'équation de mise à jour Bayésienne de la mesure probabilité associée au classifieur $f \in \mathcal{H}$ est ici

$$p_{t+1}(f) \propto p_t(f)\beta^{\mathbb{1}_{f(x_t)\neq y_t}}(1-\beta)^{\mathbb{1}_{f(x_t)=y_t}},$$

où $\beta \in [0, \frac{1}{2}]$ est un paramètre.

A chaque pas de temps, le classifieur possédant la plus grande probabilité *a posteriori* est sélectionné comme meilleur candidat pour le classifieur idéal. Soit $\hat{f}_t = \arg\max_{f\in\mathcal{H}} p_t(f)$ le classifieur ainsi sélectionné. Le but est alors de réduire le plus vite possible la probabilité que celui-ci ne soit en réalité pas idéal $P(\hat{f}_t \neq f^*)$. Pour cela, l'idée est de faire en sorte que toute la probabilité *a posteriori* soit regroupée sur un seul classifieur. La mesure du désaccord utilisée pour une donnée d'entrée x est alors la distance à $\frac{1}{2}$ de la prédiction moyenne des classifieurs pondérés par leur probabilité *a posteriori*. Ce qui apparait dans la stratégie de sélection suivante :

$$x_{t+1} = \arg\min_{x\in\mathcal{U}} |\frac{1}{2} - \sum_{f\in\mathcal{H}} p_t(f)f(x)|.$$

En effet, plus les prédiction sont différentes, plus l'annotation d'une donnée pourra différencier les classifieurs, et d'autre part, plus un classifieur est probable, plus il est important de le différencier.

Une autre solution est envisagée dans [34] avec l'algorithme EffECXtive, l'idée est de considérer que l'ensemble des données d'entrée qu'il est possible d'annoter est en nombre fini, c'est par exemple le cas lorsque le scénario envisagé est l'échantillonnage dans un réservoir. Le classifieur considéré idéal est alors celui que l'on aurait choisi si toutes les données étaient annotées. Étant donné que plusieurs jeux de données peuvent mener à un même classifieur, ils sont regroupés dans ce qu'on appelle des classes d'équivalence. Ici, soit $\mathcal{T}^{all}$ une des bases d'entrainement qui pourrait résulter de l'annotation de l'ensemble des données du réservoir, et $f_{\mathcal{T}^{all}}$ le classifieur appris sur celle-ci, la classe d'équivalence relative à un classifieur f est

$$\mathcal{H}_f = \{\mathcal{T}^{all}, f_{\mathcal{T}^{all}} = f\}.$$

Ainsi, identifier le classifieur idéal revient à identifier la classe d'équivalence à laquelle il appartient. De cette façon, le problème de classification bruité à été réduit à un problème non-bruité. L'espace des versions correspondant ici aux bases d'entrainement cohérentes avec les données reçues.

L'annotation d'une donnée d'entrée peut être vue comme permettant de casser les liens existants entre les classes d'équivalences dans le but d'identifier la classe d'équivalence idéale, il faut alors définir le poids de ces liens :

$$w(\mathcal{H}_{f_1}, \mathcal{H}_{f_2}) = \sum_{\mathcal{T}_1^{all}\in\mathcal{H}_{f_1}, \mathcal{T}_2^{all}\in\mathcal{H}_{f_2}} P(\mathcal{T}_*^{all} = \mathcal{T}_1^{all})P(\mathcal{T}_*^{all} = \mathcal{T}_2^{all}) = P(\mathcal{T}_*^{all} \in \mathcal{H}_{f_1})P(\mathcal{T}_*^{all} \in \mathcal{H}_{f_2}),$$

avec $\mathcal{T}_*^{all}$ la vraie base d'entrainement finale. Puis utiliser comme fonction objectif la somme des poids :

$$\sum_{\mathcal{H}_{f_1}\neq\mathcal{H}_{f_2}} w(\mathcal{H}_{f_1}, \mathcal{H}_{f_2}) = 1 - \sum_{f\in\mathcal{H}} P(\mathcal{T}_*^{all} \in \mathcal{H}_f)^2.$$

En effet, le but est d'affaiblir les liens entre les classes d'équivalence pour finalement pouvoir en discréditer et déterminer la classe d'équivalence idéale. Notez que nous pouvons aussi interpréter cela comme une mesure du volume de l'espace des versions.

Finalement, la stratégie de sélection adoptée est de choisir la donnée d'entrée minimisant de façon myope cette fonction objectif :

$$x_{t+1} = \arg\max_{x \in \mathcal{U}} \sum_{y_{t+1} \in Y} \hat{P}_{\mathcal{T}}(y = y_{t+1}|x)(\sum_{f \in \mathcal{H}} \hat{P}_{\mathcal{T} \cup \{(x,y_{t+1})\}}(\mathcal{T}_*^{all} \in \mathcal{H}_f)^2) - \hat{P}_{\mathcal{T}}(\mathcal{T}_*^{all} \in \mathcal{H}_f)^2.$$

Dans [6], les auteurs partent du principe que dans le cas bruité, une seule donnée d'entrée étiquetées ne permet pas de discréditer un classfieur, mais que lorsqu'un certain nombre de ces données viennent contredire ce classifieur, il est peu probable que ce soit le classifieur idéal et il peut donc être retiré de l'espace des versions. Ce qui revient à être prudent sur la réjection d'un classifieur, en effet, il serait fortement pénalisant de rejeter le classifieur idéal et il s'agit donc d'être confiant sur la décision prise. Ainsi, l'espace des versions peut se redéfinir comme l'ensemble des classifieurs probablement idéaux.

Plus formellement, un intervalle de confiance $[\epsilon_l(f, \mathcal{T}, \delta), \epsilon_u(f, \mathcal{T}, \delta)]$ peut être construit, contenant le risque réel du classifieur $f \in \mathcal{H}$ avec une certaine probabilité $1 - \delta$. De sorte que

$$\forall f \in \mathcal{H}, P(\epsilon_l(f, \mathcal{T}, \delta) \leq R_m(f) \leq \epsilon_u(f, \mathcal{T}, \delta)) \geq 1 - \delta.$$

Si pour deux classifieurs différents f_1 et f_2, les intervalles de confiances sont disjoints ($\epsilon_u(f_1, \mathcal{T}, \delta) \geq \epsilon_l(f_2, \mathcal{T}, \delta)$), cela veut dire qu'il y a une faible probabilité pour que les vrais risques soient inversés $R_m(f_1) \leq R_m(f_2)$. Dans [6], la perte binaire est utilisée mais n'importe quelle mesure m pourrait la remplacer.

L'espace des versions $\mathcal{V}$ est alors défini comme l'ensemble des classifieurs tels que la probabilité qu'il existe un autre classifieur avec un meilleur risque n'est pas faible, c'est à dire qu'il n'existe pas de classifieur avec un intervalle de confiance disjoint. Ainsi,

$$\mathcal{V} = \{f \in \mathcal{H}, \epsilon_l(f, \mathcal{T}, \delta) \leq \min_{f' \in \mathcal{H}} \epsilon_u(f', \mathcal{T}, \delta)\}. \tag{3.1}$$

A partir de cet espace des versions peut être définie une zone de désaccord dans l'espace des entrées, telle que n'importe quelle donnée d'entrée prise dans cette zone soit sujette à un désaccord pour au moins deux classifieurs de l'espace des versions.

$$\mathcal{D}_{\mathcal{V}} = \{x \in X, \exists (f_1, f_2) \in \mathcal{V}^2, f_1(x) \neq f_2(x)\}. \tag{3.2}$$

Le désaccord $Desaccord_{\mathcal{D}}(\mathcal{V})$ relatif à un espace des versions $\mathcal{V}$ mesuré sur un ensemble D de l'espace d'entrée est défini par la probabilité qu'il existe une paire de classifieurs dans l'espace des versions qui soient en désaccord sur une donnée d'entrée aléatoire tirée selon la distribution naturelle des données d'entrée restreinte à $\mathcal{D}$. Ainsi,

$$Desaccord_{\mathcal{D}}(\mathcal{V}) = \mathbb{P}_{x \sim P(x)}(\exists (f_1, f_2) \in \mathcal{V}^2, f_1(x) \neq f_2(x) | x \in \mathcal{D}).$$

Ainsi cet algorithme fonctionne de la manière suivante, il est divisé en plusieurs rounds, à chaque round, l'espace des versions est reconsidéré avec la zone de désaccord qui s'en suit. Un round consiste à tirer des données d'entrée dans la zone de désaccord, puis de calculer les intervalles de confiances à partir de ces données, jusqu'à ce qu'il soit possible de modifier l'espace des versions de telle sorte que son volume soit divisé par deux. Si c'est le cas, cela est effectué et l'algorithme passe au round suivant. L'algorithme est résumé sur la Figure 2. Le critère d'arrêt pour l'algorithme est qu'il existe un classifieur tel que la taille de l'intervalle de confiance soit inversement inférieur au désaccord de l'espace des versions considéré. Le paramètre correspondant est noté η.

$\mathcal{V} \leftarrow \mathcal{H}$
$\mathcal{D} \leftarrow \mathcal{D}_{\mathcal{V}}$ selon 3.2
$\mathcal{T} \leftarrow \emptyset$
$t \leftarrow 0$
tant que $Desaccord_{\mathcal{D}}(\mathcal{V})(\min_{f \in \mathcal{V}} \epsilon_u(\mathcal{T}, f, \delta_t) - \min_{f \in \mathcal{V}} \epsilon_l(\mathcal{T}, f, \delta_t)) > \eta$ **faire**
  $\mathcal{T} = \emptyset$
  tant que $vol_{\mathcal{D}}(\mathcal{V}_{test}) \geq \frac{1}{2} vol_{\mathcal{D}}(\mathcal{V})$ **faire**
    si $Desaccord_{\mathcal{D}}(\mathcal{V}_{test})(\min_{f \in \mathcal{V}_{test}} \epsilon_u(\mathcal{T}, f, \delta_t) - \min_{f \in \mathcal{V}_{test}} \epsilon_l(\mathcal{T}, f, \delta_t)) > \eta$ **alors**
      retourner $f^* = \arg\min_{f \in \mathcal{V}_{test}} \epsilon_u(\mathcal{T}, f, \delta_t)$
    sinon
      Tirer $x \sim P(x), x \in \mathcal{D}$
      si x *n'est pas déjà étiqueté* **alors**
        Soumettre x à l'oracle, recevoir y
      fin
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{(x, y)\}$
      $\forall f \in \mathcal{H}$, calculer $[\epsilon_l(f, \mathcal{T}, \delta_t), \epsilon_u(f, \mathcal{T}, \delta_t)]$
      Calculer $\mathcal{V}_{test}$ selon 3.1
    fin
  fin
  $\mathcal{V} \leftarrow \mathcal{V}_{test}$
  $\mathcal{D} \leftarrow \mathcal{D}_{\mathcal{V}}$ selon 3.2
  $t \leftarrow t + 1$
fin
retourner $f^* = \arg\min_{f \in \mathcal{V}} \epsilon_u(\mathcal{T}, f, \delta_t)$

Algorithme 2 : MM+M

Dans cette section, nous avons introduit le concept d'espace des version et vu que le problème d'apprentissage actif pouvait être abordé par la volonté de réduire le volume de celui-ci. Ainsi, contrairement aux sections précédentes où un unique jeu de paramètres était mis à jour à chaque pas de temps dans le but de converger vers le classifieur idéal, ici, tous les jeux de paramètres sont considérés en même temps, et chaque étape permet de filtrer les classifieurs dans le but de ne récupérer que le classifieur idéal. Nous avons vu que les premières méthodes ont été développées dans le cas non-bruité, où l'observation d'une unique donnée permet de discréditer directement tous les classifieurs non-cohérents avec elle. Cela permettant de diviser à chaque pas de temps le volume de l'espace des versions par deux en sélectionnant une donnée d'entrée adaptée, et donc de déterminer le classifieur idéal sous un nombre d'annotations optimal. Cependant, cela nécessite que le classifieur idéal soit inclus dans l'espace d'hypothèse initial, un classifieur proche de l'idéal n'ayant aucune garantie d'être retenu plus longtemps qu'un autre, et peut être éliminé tôt si une donnée sur laquelle il est en désaccord avec l'oracle est annotée. D'autre part, l'hypothèse d'un problème non-bruité limite énormément l'utilisation de ces méthodes. Nous avons aussi vu des méthodes adaptant cette approche au cas bruité, où la cohérence avec les données ne peut plus être binaire, et il est plus approprié de parler de niveau de cohérence avec les données, qui peut prendre la forme d'une probabilité *a posteriori* ou encore d'un intervalle de confiance.

Cette section est destinée uniquement à présenter l'approche consistant en la réduction de l'espace des versions, et à en comprendre le fonctionnement. Les méthodes présentées ici sont alors celles étant à l'origine du domaine. Cependant, cette approche fait encore l'objet de nombreuses études comme nous pouvons le voir avec [42], [7], [76], [41] et [47].

Chapitre 4

De la sélection de données en Optimisation

4.1 Introduction

L'apprentissage actif n'est pas le seul paradigme à faire intervenir de la sélection de données. En effet, celle-ci intervient également dans un autre domaine qui est celui de l'optimisation. Ce dernier se définit comme le domaine visant à déterminer l'optimum d'une fonction initialement inconnue, bruitée ou non. Pour cela un algorithme d'optimisation pourra évaluer la fonction en un certain nombre de points avant de renvoyer une approximation de l'optimum. L'enjeu d'un tel algorithme est de définir quelles sont les évaluations à effectuer pour que l'approximation converge le plus rapidement possible vers l'optimum réel. Ceci s'apparente donc à de la sélection de données dans lequel un oracle fournit le résultat des évaluations. Cependant, l'apprentissage actif et l'optimisation différent dans leur intentions. En effet, dans un cas le but est de connaitre la fonction dans son ensemble avec une précision équitable quelle que soit la donnée fournie, dans l'autre le but est de connaitre un endroit unique (initialement inconnu) de la fonction avec une précision maximum.

L'objet de cette thèse concerne seulement l'apprentissage actif, cependant nous souhaitons nous intéresser aux méthodes de sélection des données développées pour l'optimisation dans le but d'en extraire des solutions pouvant y être adaptées. De plus, dans le chapitre précédent, nous avons présenté le problème d'apprentissage actif ainsi que les diverses approches abordées pour le résoudre. Nous avons vu que le choix des données était toujours guidé par un critère idéal à maximiser : le risque local dans le cas de l'échantillonnage d'incertitude, la décroissance du risque global dans le cas de la réduction de l'erreur et le volume de classifieurs allant être éliminés dans le cas de la réduction de l'espace des versions. Ce critère idéal est inconnu puisqu'il est fonction des paramètres de la distribution de l'oracle qui sont eux-même initialement inconnus. Le but est alors, pour permettre de sélectionner la donnée adéquate, de rechercher le maximum du critère idéal qui n'est que partiellement connu à travers les données déjà étiquetées. Il existe donc un lien fort entre apprentissage actif et optimisation. Une différence réside toutefois dans le fait que le critère idéal se trouve modifié après chaque acquisition d'une nouvelle étiquette.

De plus, un des aspects du problème d'apprentissage actif se rapproche d'un problème usuel traité en optimisation. Plus précisément, parmi les différentes solutions qui ont été présentées pour chaque approche, chacune se distinguait sur la façon de gérer les paramètres inconnus dans le critère idéal. L'approche naïve consiste à les remplacer directement par leur estimation. Or, dans ce cas, on peut observer un phénomène de concentration des données étiquetées dans certaines zones, notamment lorsque le séparateur idéal est disjoint. Par exemple, dans le cas de l'échantillonnage

d'incertitude, le but est d'affiner le séparateur en sélectionnant des données se situant à proximité de ce dernier. Si l'on ne se fie qu'à l'estimation courante du séparateur, les données sélectionnées vont donc converger vers la première partie du séparateur idéal découverte. Ceci peut entrainer une chute des performance voire que le classifieur n'atteigne jamais la prédiction optimale. Pour palier à cela, il est nécessaire de forcer l'exploration de l'espace d'entrée afin d'être sûr de ne pas passer à côté d'une partie du séparateur. Cela est équivalent à chercher à améliorer la qualité de l'estimation des paramètres sur l'ensemble de l'espace d'entrée en se basant sur l'incertitude liée à celle-ci. Cependant, cela requiert de sélectionner des données dans ce but plutôt qu'elles servent à améliorer la performance du classifieur. Il est donc nécessaire de faire un choix entre améliorer directement la performance du classifieur et améliorer l'estimation des paramètres pouvant mener à un meilleur choix des données suivantes. Ceci peut donc être vu comme un compromis entre amélioration à court terme ou à long terme. Le but est alors de savoir comment gérer le compromis entre ces deux attitudes afin de mener à une sélection optimale des données d'entrée. Notez que l'amélioration des performances du classifieur est conduite par l'estimation brute des paramètres tandis que l'exploration est conduite par l'incertitude.

Ce problème est plus connu sous le nom de dilemme entre exploration et exploitation. Celui-ci a d'abord été introduit puis largement étudié dans le cadre de l'optimisation en budget fini. En optimisation standard, le but est de connaitre l'optimum d'une fonction avec précision, l'unique objectif est donc qu'à la fin du processus, l'algorithme renvoie l'optimum global. Dans le cas du budget fini, la donnée d'entrée sélectionnée à chaque pas de temps doit également fournir la meilleure valeur possible. Le but étant alors de maximiser la somme cumulée des résultats des évaluations. Celui-ci est toujours vu comme un problème d'optimisation, car sur le long terme, seul l'optimum doit être sélectionné. Cependant, il permet de mêler les phases de recherche de l'optimum et d'exploitation de ce dernier. L'utilité d'un tel problème peut être illustrée par le cas de l'exploitation minière dans lequel la connaissance des filons passe par l'extraction même de ces derniers. On désire donc dans le même temps trouver le filon le plus riche et rentabiliser la recherche en extrayant un maximum de ressource durant le processus.

Le but est alors d'adapter les méthodes de déjà développées pour le dilemme entre exploration et exploitation au problème d'apprentissage actif afin de fournir une meilleure façon de gérer l'incertitude sur les paramètres de la distribution de l'oracle. En effet, nous avons déjà mentionné que dans le problème d'apprentissage actif, le critère idéal est une fonction à maximiser dont la connaissance est seulement partielle, ceci peut donc se voir comme un problème d'optimisation. D'autre part, le risque global final peut être vu comme le risque global initial moins la somme des décroissances, minimiser le risque global final revient donc à maximiser la somme cumulée des décroissances. Finalement, dans un problème d'optimisation en budget fini, le but est que les meilleures récompenses intervienne le plus tôt possible, c'est aussi le cas en apprentissage actif où l'on souhaite obtenir la meilleure performance avec un minimum d'échantillons.

Le dilemme entre exploration et exploitation a tout d'abord été posé dans le cadre des bandits à bras multiples [73]. Ce problème, dont le nom provient de l'étude d'un ensemble de machines à sous ayant des distributions de gains différentes et initialement inconnues, permet d'étudier ce dilemme sur des données d'entrée discrètes et indépendantes ; en effet, les tirages d'une machine ne permettent pas de déduire la distribution d'une autre. Le but est alors de récolter un maximum de gains avec un nombre de tirages limité, il faut donc apprendre chaque distribution indépendamment tout en jouant la meilleure machine plus souvent. Ce problème a ensuite été dérivé pour le cas d'une fonction à maximiser. Ainsi, les données d'entrée appartiennent à un espace continu, l'apprentissage étant rendu possible par une hypothèse de régularité sur la fonction étudiée. L'incertitude sur l'estimation de la fonction est alors fournie par l'utilisation de Processus Gaussiens, une approche

Bayésienne pour l'apprentissage de fonction avec un *a priori* Gaussien sur la valeur de sortie en tout point.

Dans ce chapitre, nous présenterons donc les principales méthodes permettant de résoudre le dilemme entre exploration et exploitation afin de s'en inspirer pour développer de nouvelles méthodes d'apprentissage actif. Nous commencerons d'abord par poser le cadre des bandits à bras multiples. Ensuite, nous étudieront les principales méthodes développées pour ce problème en portant une attention particulière à la méthode dit de l'optimisme face à l'incertitude. Finalement, nous présenterons les travaux existants se plaçant à la frontière entre l'apprentissage actif et les approches d'optimisation en budget fini telles qu'étudiées à travers les bandits à bras multiples.

4.2 Les bandits à bras multiples

Le problème des bandits à bras multiple tire son nom des machines à sous appelées bandits manchot. Celui-ci ne se limite évidement pas à ce contexte et peut être utilisé pour représenter de nombreux problèmes dans divers domaines. Cependant, cette représentation illustre de façon simple le type de problèmes étudié et permet de bien comprendre son principe de fonctionnement, c'est pourquoi nous le reprenons dans notre description. Ainsi, supposons qu'un joueur soit confronté à un ensemble de machines à sous. Contrairement à la pratique usuelle dans les casinos, toutes ne sont pas identiques et certaines donnent de meilleurs gains que les autres en moyenne, mais cette information est inconnue du client. Ce dernier va donc jouer sur ces machines et tenter de gagner le plus d'argent possible. Pour cela, il dispose d'un certain nombre de crédits. Chaque crédit lui donne la possibilité de jouer une machine une fois, ce qui lui renvoie un gain en fonction de la machine jouée. Le problème est alors de savoir quelles machines jouer pour récupérer le gain total maximal. Il n'est toutefois pas obligatoire de définir la répartition des crédits à l'avance, le joueur a la possibilité de choisir successivement les machine sur lesquels il joue en fonction des résultats obtenus auparavant. Notez que lorsque le joueur joue une machine, il ne connait pas la récompense qu'il aurait obtenu si il avait joué les autres. Ainsi, pour apprendre le gain moyen d'une machine, il faut la jouer. De cette façon, dépenser un crédit sur une machine peut servir deux objectifs : récupérer un récompense ou apprendre le gain moyen. Le problème que nous venons de décrire est celui des bandits cumulatifs, il existe d'autres problèmes comme les bandits adversariaux [4], les bandits linéaires [5] ou les bandits à simple regret [11], mais nous laissons ceux-ci de côté pour nous concentrer sur le problème le plus courant et celui que nous avons utilisé dans nos travaux.

4.2.1 Définition du problème

Le problème des bandits à bras multiple a été formalisé dans [64], c'est le processus de décision séquentiel le plus simple faisant intervenir le dilemme entre exploration et exploitation. Soit un ensemble de bras (d'indices) $\{1, ..., K\}$ correspondant à chacune des machine à sous. A chaque bras $k \in \{1, ..., K\}$ est associée une distribution ν_k initialement inconnue mais fixe, soit sa moyenne μ_k. Soit un budget de tirages n, à chaque pas de temps $t \in [\![1, n]\!]$, l'utilisateur peut tirer un bras $k_t \in \{1, ..., K\}$, il reçoit ainsi une récompense $y_t \sim \nu_{k_t}$. L'objectif est de maximiser la somme cumulée des récompenses : $\sum_{t=1}^{n} y_t$.

Afin d'étudier la performance d'une stratégie de tirages, on définit le regret cumulatif R_n comme la différence entre le gain maximal, atteint si l'on avait joué à chaque pas de temps la meilleure machine, et le gain effectivement atteint. Le pseudo-regret $\bar{R}_n$ est le regret espéré sur les tirages des récompenses selon les distributions des bras joués. Ainsi,

$$\bar{R}_n = \mathbb{E}_{y_t \sim \nu_{k_t}}[n \max_{k \in \{1,...,K\}} \mu_k - \sum_{t=1}^{n} y_t].$$

Le regret est le manque à gagner de la stratégie évaluée, par rapport à la stratégie connaissant initialement toutes les distributions. Le but est alors de définir une stratégie telle que le regret soit minimum. Cependant, il est impossible de définir une stratégie qui donne un regret nul. En effet, les distributions étant inconnues, il est nécessaire d'apprendre les récompenses moyennes de chaque bras pour savoir lequel est optimal. Surtout, pour pouvoir écarter un bras, il est nécessaire de l'avoir joué. Il y a donc nécessairement des bras sous optimaux qui doivent être joués, ce qui impacte le regret.

Pour savoir si une stratégie peut encore être améliorée ou non, il est nécessaire de définir le regret minimum atteignable par une stratégie quelconque. Dans [53] les auteurs montrent que si la stratégie vérifie pour n'importe quelle distribution de Bernoulli $\{\nu_k = \mathcal{B}er(\mu_k)\}_{k\in\{1,\dots,K\}}$ et n'importe quel $a > 0$, que quand $n \to +\infty$, $R_n = o(n^a)$, alors pour n'importe quel jeu de distributions de Bernoulli le regret minimum est logarithmique et vérifie

$$\lim_{n\to+\infty} \inf \frac{\bar{R}_n}{\log n} \geq \sum_{k=1}^{K} \frac{\mu_{k^*} - \mu_k}{KL(\mathcal{B}er(\mu_k), \mathcal{B}er(\mu_{k^*}))},$$

avec $k^* = \arg\max_{k\in\{1,\dots,K\}} \mu_k$, $KL(.,.)$ la divergence de Kulback-Leibler et $\mathcal{B}er(p)$ une distribution de Bernoulli de paramètre p.

4.2.2 Quelques algorithmes simples

Afin de sélectionner le meilleur bras en comparaison des autres, il est nécessaire d'avoir connaissance des gains moyens de l'ensemble des bras, et donc de les avoir tirés, ce qui se traduit par de l'exploration. Mais dans le même temps, pour maximiser la récompense, il faut tirer le meilleur bras selon la connaissance actuelle, ce qui est considéré comme de l'exploitation. Concevoir une stratégie de sélection des bras consiste à équilibrer l'exploration et l'exploitation dans le but de minimiser le regret. Si le regret atteint sa borne inférieure, alors on dit que la stratégie est optimale. Dans la suite nous présentons quelques-unes des méthodes les plus connues

Tout d'abord, nous pouvons trouver la stratégie $\epsilon - greedy$ [72]. Soit deux stratégies de sélection, une purement exploratrice : la stratégie aléatoire qui sélectionne un bras au hasard uniformément dans l'ensemble des bras, et une purement exploitante : la stratégie gloutonne qui sélectionne le bras qui a reçu la meilleur récompense moyenne jusqu'ici. L'idée derrière $\epsilon - greedy$ est de sélectionner à chaque pas de temps une des deux stratégies de manière aléatoire. La nombre de fois que la stratégie gloutonne est sélectionnée par rapport à la stratégie aléatoire est gérée par un paramètre $\epsilon \in [0, 1]$ qui donne son nom à la stratégie. Seulement, lorsque peu d'information est disponible, la stratégie aléatoire devrait être prépondérante étant donné qu'il faut gagner de l'information plus vite et que l'estimation courante est peu fiable. Tandis qu'au fur et à mesure que l'algorithme progresse, la stratégie gloutonne devrait prendre le pas sur l'autre, de sorte qu'à terme, uniquement le meilleur bras soit sélectionné. Ainsi, le paramètre ϵ devrait évoluer avec le temps. Dans [3], il est prouvé que si $\epsilon_t = \frac{K}{d^2 t}$, avec d inférieur à l'écart entre la moyenne espérée du bras optimal et celle du deuxième meilleur, alors le regret croit de manière logarithmique comme $\frac{K \log n}{d^2}$.

L'échantillonnage de Thompson [73] est une des méthode les plus ancienne, elle se base sur un bandit à bras multiple dont toutes les distributions sont Bernoulli. Soit $\{\nu_k = \mathcal{B}er(\mu_k)\}_{k\in\{1,\dots,K\}}$ ces distributions. En suivant une approche Bayésienne, une distribution de probabilité *a posteriori* est ensuite calculée sur les paramètres de Bernoulli. Soit $\pi_{k,t}$ la distribution *a posteriori* du paramètre μ_k au temps t. L'idée est de tirer un paramètre pour chaque bras selon les distributions *a posteriori*, puis de sélectionner le bras en faisant comme si ce paramètre était le vrai. Soit $\tilde{\mu}_{k,t} \sim \pi_{k,t}$, alors la stratégie revient à $k_{t+1} = \arg\max_{k\in\{1,\dots,K\}} \tilde{\mu}_{k,t}$.

4.2.3 Optimisme face à l'incertitude

Les prochaines stratégies que nous présentons reposent sur le principe dit de l'optimisme face à l'incertitude. Sous cette appellation repose le fait, lorsqu'un système décidant est confronté à une incertitude sur les paramètres d'un problème, de faire la liste des paramètres cohérents avec la connaissance courante, provenant de l'ensemble des observations, puis de déterminer le plus

favorable ou pouvant donner les meilleurs résultats. Ensuite, la décision est effectuée en considérant ces paramètres les plus favorables. En faisant cela, deux cas peuvent se présenter. Soit les vrais paramètres sont effectivement ceux considérés, et les utiliser dans la décision est alors profitable. Soit ce n'est pas le cas, et la décision effectuée engendre une observation qui vient les contredire et permet de les enlever de la liste des paramètres cohérents. Notez qu'une autre décision n'aurait pas obligatoirement contredit les paramètres considérés. Ainsi, cette stratégie est profitable dans tout les cas.

Dans le cas des bandits à bras multiples, ce principe a été utilisé dans l'algorithme des bornes de confiance supérieures (UCB). Ici, les paramètres inconnus sont les récompenses espérées de chaque bras. Soit μ_k le paramètre du bras k. En pratique, la liste des paramètres cohérents est représentée par des intervalles de confiance qui sont fournis par des inégalités de concentration. Soit $\hat{\mu}_{k,t}$ la moyenne des récompenses reçues jusqu'au temps t par la bras k. Soit $T_{k,t}$ le nombre de fois que le bras k a été tiré jusqu'au temps t. Soit une inégalité de concentration donnant f une fonction telle que $\forall \epsilon \in \mathbb{R}$,

$$\mathbb{P}(\mu_k - \hat{\mu}_{k,t} > \epsilon) \leq f(T_{k,t}, \epsilon),$$

alors, si f admet un inverse par rapport à ϵ, avec une probabilité d'au moins $1 - \delta$,

$$\mu_k < \hat{\mu}_{k,t} + f^{-1}(T_{k,t}, \delta).$$

Ainsi, en utilisant un intervalle de confiance dont la borne supérieure est $\hat{\mu}_{k,t} + f^{-1}(\delta)$, un paramètre est considéré cohérent si la probabilité qu'il renvoie une récompense moyenne supérieure à celle observée est faible. Un exemple d'inégalité de concentration pouvant être utilisée est l'inégalité de Hoeffding [43] [3] dans laquelle $f(T_{k,t}, \epsilon) = e^{-2T_{k,t}\epsilon^2}$.

L'algorithme UCB consiste donc à sélectionner à chaque pas de temps le bras k_{t+1} tel que

$$k_{t+1} = \underset{k \in \{1,\dots,K\}}{\arg\max}\ \hat{\mu}_{k,t} + f^{-1}(\delta).$$

Notez que de cette façon se trouve combiné les deux aspects du dilemme entre exploration et exploitation en un seul critère. En effet, la première partie $\hat{\mu}_{k,t}$ est l'estimation naïve de la récompense espérée et est strictement le critère utilisé par une stratégie d'exploitation pure. La deuxième partie, $f^{-1}(\delta)$, quant à elle ne dépend que du nombre d'observations considéré, elle est donc liée à l'incertitude de l'estimation et, utilisée seule comme critère, correspondrait à une stratégie purement exploratrice. Ainsi, tandis que la stratégie $\epsilon - greedy$ présentée précédemment tentait de combiner deux stratégies possédant chacune un critère différent en les alternant, ici ce sont les critères qui sont combinés pour obtenir une unique stratégie.

Dans le cas où l'inégalité de Hoeffding est utilisée, cela donne la borne sur le pseudo-regret suivante :

$$\bar{R}_n \leq \sum_{k:\mu_{k^*} > \mu_k} (\frac{8}{\Delta_k} \log n + 2).$$

Plusieurs améliorations de cet algorithme ont été proposées. Parmi elles, l'algorithme UCB-V [2] propose d'utiliser la variance de la distribution. En effet, l'inégalité de concentration peut être rendue plus étroite si elle peut dépendre de la variance de la distribution, en utilisant l'inégalité de Bernstein. Étant donné que cette variance est elle même inconnue, l'utilisation d'une autre inégalité de concentration permet de construire un intervalle de confiance sur la variance. En mêlant ces deux intervalles, il est possible de créer un intervalle beaucoup plus fin.

Une deuxième variante appelée KL-UCB [33] [59] propose de déterminer la cohérence des paramètres non pas par une borne sur leur distance à la moyenne estimée, mais par une borne sur

leur divergence de Kulback-Leibler. Ce qui est intéressant avec cette approche est qu'elle possède une borne sur le pseudo-regret, à la fois en échantillons finis et asymptotiquement optimale pour les distributions de Bernoulli.

Un autre algorithme qu'il est intéressant de citer dans le cadre de cette thèse est Bayes-UCB [51]. Les auteurs proposent d'adapter le principe de l'optimisme face à l'incertitude dans une approche Bayésienne. En effet, l'algorithme UCB se place d'un point de vue fréquentiste, où les paramètres sont déterministes mais inconnus. Du point de vue Bayésien, les paramètres sont des variables aléatoires tirées dans une distribution *a priori*. De cette façon, ils sont capables d'utiliser la distribution *a posteriori* qui est entièrement connue. Ici, plutôt que d'utiliser des intervalles de confiance, qui listent les paramètres cohérents en se basant sur la probabilité qu'ils renvoient les récompenses observées, une approche Bayésienne privilégiera l'utilisation d'intervalles de crédibilité, qui sont définis tels que le paramètre se trouve à l'intérieur avec une certaine probabilité. Ainsi, si $\pi_{k,t}$ est la distribution *a posteriori* du paramètre μ_k du bras k au temps t, alors on définit un quartile $Q(\delta, \pi_{k,t})$ tel que

$$\mathbb{P}_{\pi_{k,t}}(\mu_k \leq Q(\delta, \pi_{k,t})) = \delta.$$

Ce quartile agit comme la borne supérieure de l'intervalle de crédibilité. Notez qu'alors que précédemment les intervalles de confiance étaient définis grâce à des inégalité de concentration, ici, la probabilité est soumise à une égalité. En effet, la forme de la distribution *a posteriori* étant entièrement connue, la probabilité peut être calculée exactement. En appliquant le principe de l'optimisme face à l'incertitude, la stratégie de sélection est

$$k_{t+1} = \underset{k \in \{1,\dots,K\}}{\arg\max}\ Q(\delta_t, \pi_{k,t}).$$

Notez que les auteurs choisissent de faire dépendre la probabilité de t en fixant $\delta_t = 1 - \frac{1}{t(\log n)^c}$, où c est un paramètre de l'algorithme. Dans les expériences, $c = 0$. Cet algorithme satisfait une borne sur le pseudo-regret en échantillon fini qui implique une optimalité asymptotique. De plus les évaluations montrent qu'il surpasse $KL - UCB$. L'utilisation d'une approche Bayésienne est particulièrement efficace lorsque la famille de distributions rencontrée est connue. En effet, la famille de distributions *a priori* à utiliser est alors dictée par cet connaissance. Puisque nos travaux se placent dans le cadre de la classification, dans lequel les étiquettes sont tirées selon des distributions de Bernoulli, alors ces résultats sont extrêmement utiles.

Dans le problème des bandits à bras multiple, les bras sont en nombre fini et complètement séparés. En effet, les observations d'un bras n'interviennent pas dans l'estimation de la récompense espérée des autres bras. Ceci ne permet de représenter que les problèmes d'optimisation sur un espace discret et fini. Pourtant, dans le cas général, un problème d'optimisation consiste à trouver le maximum d'une fonction ayant un espace continu X en entrée. En faisant l'hypothèse que la fonction appartient à un certain ensemble ou possède une certaine régularité, il est possible d'estimer la valeur de sortie sur tout l'espace avec seulement un nombre fini d'observations. Il y a donc une influence d'une observation acquise pour une donnée d'entrée sur l'estimation d'une autre donnée d'entrée. La question est alors de savoir si le principe de l'optimisme face à l'incertitude peut être utilisé dans un tel contexte. Dans [71], les auteurs introduisent un algorithme appelé $GP - UCB$ qui utilise ce principe avec des Processus Gaussiens. Ce qui consiste à considérer que la fonction suit une distribution de probabilité et à utiliser une approche Bayésienne avec une distribution *a priori* Gaussienne. De cette façon, la distribution *a posteriori* au temps t, elle aussi Gaussienne, n'est caractérisée que par deux fonctions : la moyenne μ_t et la variance σ_t^2. De plus, dans ce cadre, un intervalle de crédibilité sur la valeur de sortie en $x \in X$ est $[\mu_t(x) - \alpha\sigma_t(x), \mu_t(x) + \alpha\sigma_t(x)]$,

avec α un paramètre dépendant de la probabilité souhaitée pour l'intervalle de crédibilité. La stratégie de sélection est alors

$$x_{t+1} = \arg\max_{x \in X} \mu_t(x) + \alpha\sigma_t(x).$$

En faisant cela sur n pas de temps, ils obtiennent une borne sur le regret cumulé de l'ordre de $\sqrt{n}$ avec une forte probabilité.

4.3 Liens entre la théorie des bandits à bras multiples et l'apprentissage actif

Jusqu'ici, nous avons vu que l'apprentissage actif impliquait le dilemme entre exploration et exploitation et qu'à chaque pas de temps la donnée à annoter est sélectionnée sur la base d'un critère à maximiser. Ensuite, nous avons vu que ce dilemme avait été très largement étudié à travers le problème des bandits à bras multiples et que de nombreuses méthodes avaient été proposées. Nos travaux s'orientent donc vers l'adaptation d'une des approches, en l'occurrence l'optimisme face à l'incertitude, à l'apprentissage actif pour la classification. Plusieurs méthodes développées pour les bandits à bras multiples ont déjà fait l'objet d'adaptation à l'apprentissage actif. Dans cette section, nous recensons et décrivons ces algorithmes pour ensuite être capable d'appréhender la spécificité de nos travaux.

Tout d'abord, la mise en évidence du dilemme entre exploration et exploitation au sein de l'apprentissage actif n'est pas nouvelle puisque c'est l'objet de [63]. Dans cet article, ces deux aspects sont abordés dans le cadre de l'échantillonnage selon l'incertitude et prenne la forme de stratégie de sélection distinctes. La première est une stratégie d'exploitation pure qui consiste à affiner la frontière de décision en sélectionnant une donnée d'entrée proche de celle-ci. La deuxième est purement exploratoire et consiste à repartir les données de façon uniforme sur l'espace d'entrée. Ainsi, dans celle-ci, la donnée sélectionnée est celle qui s'éloigne le plus de n'importe quelle donnée déjà étiquetée. L'idée est alors de combiner ces deux stratégies de la même manière que dans l'algorithme $\epsilon - greedy$, c'est à dire en définissant à chaque pas de temps une probabilité de jouer l'une ou l'autre stratégie. Ici, cette probabilité est guidée par l'impact de l'étiquette nouvellement acquise sur le classifieur. Ainsi, si le classifieur se trouve beaucoup modifié, c'est que l'exploration a été fructueuse et il est souhaité de continuer à explorer. L'idée est similaire à l'algorithme $MM + M$ que nous avons déjà présenté, sans pour autant que le dilemme entre exploration et exploitation n'y soit mentionné. Rappelons que dans cet algorithme, la stratégie utilisée était dictée par la réussite de la prédiction de l'étiquette acquise.

Nous avons aussi déjà parlé de l'algorithme $UPAL$ [31], dans lequel la stratégie de sélection est stochastique. Cela permettait de laisser une part à la distribution naturelle des données d'entrée dans la probabilité de sélection des données étiquetées afin de corriger l'influence de ces dernières dans l'apprentissage du classifieur. Notez que ceci permet, en outre, d'explorer en sélectionnant de temps en temps une donnée *a priori* non favorable. D'autant plus qu'une probabilité minimum est mise en place, sa valeur permettant de gérer le compromis entre exploration et exploitation. Notez aussi la forte similarité entre cette stratégie et un algorithme du type $Softmax$ [15] dans laquelle le critère de sélection est aussi stochastique. Dans [32], les auteurs améliorent l'algorithme $UPAL$ en utilisant le classifieur qui minimise une borne inférieure sur le risque, plutôt que le risque estimé. Le but étant de trouver le classifieur pour lequel le risque est minimal, il est possible de considérer les classifieurs possibles comme des bras d'un bandit, puis de gérer l'exploration en sélectionnant un classifieur en fonction de sa borne inférieure sur le risque. Une fois que le classifieur a été choisi,

le but est d'améliorer l'estimation de son risque en sélectionnant la donnée d'entrée appropriée. Nous pouvons donc interpréter cet algorithme comme utilisant le principe de l'optimisme face à l'incertitude dans une approche basée sur l'espace des versions.

Il est aussi important de citer la thèse française [12] qui fournit un travail assez conséquent sur l'utilisation du principe de l'optimisme face à l'incertitude dans l'apprentissage actif basé sur la minimisation de l'erreur de prédiction, qu'elle soit locale ou globale, dans le contexte de la régression d'histogramme. Ainsi, soit l'espace d'entrée divisé en un certain nombre de parties, le but est de prédire pour chacune d'entre elles la valeur correspondant à l'espérance du résultat pour une donnée d'entrée tirée dans cette partie de l'espace. Ceci est alors vu comme un problème d'allocation du budget. Selon la forme de l'erreur que l'on souhaite minimiser, l'allocation optimale est proportionnelle à la variance [13] ou à l'écart-type [14] de la distribution sous-jacente. Ces paramètres étant inconnus, l'idée est de voir ceci comme un bandit à bras multiple où l'on cherche à minimiser le regret simple (non-cumulé). Un algorithme utilisant les bornes supérieures d'intervalles de confiance définis pour ces paramètre est alors développé. Le cas d'une partition adaptative a aussi été envisagé dans laquelle les parties de l'espace peuvent être redécoupées lorsqu'elles contiennent assez de données et que la variance estimée est assez forte.

En apprentissage actif, le but est que la fonction de prédiction obtenue soit performante. En optimisation, le but est de trouver l'emplacement dans l'espace d'entrée pour lequel la valeur de la sortie est maximale. La recherche active est un domaine à cheval entre les deux où, dans un contexte de classification binaire, le but est de trouver les échantillons positifs, mais qu'une fois que ceux-ci ont été trouvés ils sont retirés de l'ensemble et il faut aller cherche ailleurs. Dans ce contexte, [58] introduit un algorithme utilisant le principe de l'optimisme face à l'incertitude basé sur $GP-UCB$. Dans ce dernier, le terme d'exploration cherche à sélectionner les données les plus éloignées des données déjà étiquetées. Ici, celui-ci est modifié pour que les données puissent tout de même avoir de l'influence sur les autres.

4.4 Conclusion intermédiaire

Au cours de ces deux derniers chapitres, nous avons pu décrire différents problèmes issus de la littérature ainsi qu'entrevoir les approches envisagées pour les résoudre au sein des méthodes existantes. A travers une vision unifiée de ces méthodes, nous avons mis en évidence les points clés de divergence entre elles. Dans cette section, nous tachons de rassembler et synthétiser l'ensemble des remarques apportées précédemment afin de voir comment elles s'articulent entre elles et d'appréhender la nature de nos contributions.

Le premier chapitre a permis de montrer que dans le problème d'apprentissage actif, quel que soit l'approche utilisée, la difficulté majeure est de gérer l'incertitude liée aux paramètres de la distribution de l'oracle. Nous avons montré à travers un problème simple qu'ignorer cette incertitude dégradait drastiquement les performances jusqu'à empêcher le classifieur d'atteindre sa performance optimal même avec un nombre infini d'échantillons. Les solutions existantes pour considérer l'incertitude pouvant être, par exemple, sous une approche Bayésienne, d'effectuer l'espérance sur la distribution *a posteriori* [49] [50], ou bien d'employer le pire [36] [46] ou le meilleur cas [39]. Une autre solution est de traiter cela comme un dilemme entre exploration et exploitation et d'appliquer à l'apprentissage actif les méthodes développées pour ce problème [63] [13] [14]. Pour cela, il faut reposer le problème d'apprentissage actif comme un problème d'optimisation en budget fini. En effet, nous avons vu qu'avant que les paramètres ne soient estimés, les stratégies de sélection prennent la forme d'un critère optimal à maximiser. Ce dernier n'étant que partiellement connu à travers les observations passées.

Dans le deuxième chapitre, nous avons étudié le problème des bandits à bras multiples, celui-ci étant le cas le plus simple d'optimisation en budget fini. Dans sa version la plus classique, le but est de minimiser le regret cumulé, et donc de maximiser la somme des récompenses. Ceci se prête bien à l'apprentissage actif, et particulièrement à l'approche appelée réduction de l'erreur, car le critère idéal se définit par la décroissance du risque réel. Le but final étant de minimiser le risque, cela revient maximiser la somme cumulée des décroissances, et donc des évaluations du critère idéal, à la différence qu'il n'y a pas de feedback direct, celles-ci n'étant pas directement observées mais voient leur connaissance affinée grâce aux observations acquises. Notez que dans [13] et [14], seul le regret simple est minimisé puisque l'objectif uniquement que l'allocation optimale soit atteinte à la fin du processus.

Nous avons entrevu un certain nombre de méthodes pour résoudre le problème des bandits à bras multiples, et vu que les plus efficaces utilisaient le principe dit de l'optimisme face à l'incertitude. Parmi elles, les meilleures sont celles qui s'adaptent à la famille de distributions rencontrées, sous l'hypothèse que celle ci soit connue. Ainsi, dans le cas où la famille de distribution est entièrement connue, la meilleure performance est obtenue en utilisant une approche Bayésienne pour définir les intervalles de confiances utilisés. Puisque nous étudions l'apprentissage actif dans le contexte de la classification, nous savons que la distribution de l'oracle est Bernoulli, il est alors possible d'utiliser une approche Bayésienne. Un autre avantage à utiliser cette approche est dû au fait que la distribution de récompense n'est pas stationnaire. En effet, d'un pas de temps sur l'autre, la donnée à échantillonner idéalement n'est pas la même, et donc le critère idéal change. Ceci ne permet donc pas de l'estimer à l'aide d'une méthode fréquentiste. Cependant, les paramètres de la distribution de l'oracle restent constant et peuvent donc être estimés grâce aux observations passées. Ainsi, étant donné que le critère idéal est une fonction des paramètres, une distribution *a posteriori*, et donc un intervalle de confiance, peut être calculé sur celui-ci.

Dans la suite, nous allons voir comment construire des méthodes d'apprentissage actif basées sur la minimisation du risque local ou global en faisant usage du principe de l'optimisme face à l'incertitude. Nous essaierons de tirer pleinement parti de toutes les spécificités du problème de classification. Ainsi, cela consiste tout d'abord à ce que le critère de sélection idéal soit spécifiquement conçu pour la classification en le faisant se reporter à la performance de prédiction calculée à partir de la perte binaire. Ensuite, nous ferons en sorte qu'une approche Bayésienne soit utilisée pour définir les intervalles de confiance utilisés. Ces méthodes seront d'abord développées pour le problème simple des bandits à bras multiples puis seront progressivement complexifiées pour finalement être capable considérer de un espace d'entrée continu. Bien que l'idée d'utiliser le principe de l'optimisme face à l'incertitude en apprentissage actif soit déjà présente dans [13] [14] de laquelle nous nous sommes inspiré, nos travaux se distinguent par plusieurs aspects. Dans cette dernière, le contexte est celui de la régression, ce qui en plus de changer le critère idéal ne permet pas d'utiliser une approche Bayésienne de façon simple étant donné que la distribution des valeurs de sorties n'est pas fixée. De plus, nous verrons que le critère idéal qui y est défini n'exploite pas efficacement toute l'information disponible, à savoir la valeur courante de l'estimation. Ensuite, ces méthodes ne sont étudiées que sur une partition de l'espace d'entrée, potentiellement adaptative, mais ne permettent pas réellement d'être utilisées pour l'apprentissage actif d'une méthode de régression efficace. Dans ce sens, nos travaux vont plus loin dans l'étude de l'emploi de ce principe en apprentissage actif.

Deuxième partie

Contributions

Chapitre 5

Avant propos sur la démarche employée

Comme annoncé dans la partie précédente, nos travaux vont maintenant consister à montrer que le principe de l'optimisme face à l'incertitude peut se révéler efficace en apprentissage actif sur un problème de classification, lorsque employé en tant que moyen de gérer l'incertitude sur le critère idéal. Pour cela, il s'agit de développer un algorithme d'apprentissage actif qui en fait usage sur un classifieur en particulier et d'effectuer une série d'évaluations en le comparant aux autres méthodes évoquée précédemment. Le but va être que le classifieur sur lequel il s'applique soit un classifieur populaire dans la littérature reconnu pour ses performances. Cependant, s'attaquer directement à un tel classifieur peut être une tâche difficile. En effet, on n'en contrôle pas bien les comportements résultants et ceux que l'on observe peuvent être difficile à interpréter. Ce travail va donc se diviser en plusieurs étapes s'appuyant sur une sélection de classifieurs de complexité progressive. Il s'agit ainsi de commencer par étudier un problème simplifié à l'extrême, ce qui permettra de comprendre comment ce principe peut être utilisé et de concevoir les mécanismes nécessaires à son fonctionnement. Puis de complexifier petit à petit le problème en se concentrant sur l'introduction d'un nouvel aspect à chaque étape afin de bien identifier le rôle de chaque élément de l'algorithme ainsi que des approximations effectuées en essayant de les éviter au maximum. Un algorithme différent sera introduit à chacune de ces étapes en étant spécifiquement adapté à chacun de ces problèmes. Toutefois, tous les algorithmes qui seront développés ont tous le même objectif d'utiliser le principe de l'optimisme face à l'incertitude et partagent ainsi un même schéma de construction. Dans ce chapitre, nous commençons par décrire le cheminement dans son ensemble en présentant brièvement chaque étape et en insistant sur la façon dont elles s'articulent entre elles. Ceci permettra, lors de la lecture, de re-situer chaque chapitre vis-à-vis de son rôle dans cheminement global ainsi que de comprendre les raisons qui ont motivés le choix des problèmes étudiés. Ensuite, nous nous attarderons à bien décrire le schéma de construction qui sera repris par l'ensemble des algorithmes suivants afin que celui-ci soit bien compris, et permettra par la suite de relever plus facilement les particularités de chaque algorithme.

Commençons par détailler le cheminement. Le premier problème étudié est celui des partitions fixes avec des zones indépendantes, celui-ci est extrêmement simple puisqu'il n'est régit que par un faible nombre de paramètres. Notez que notre intention n'est ici que de réaliser une étude préliminaire, l'utilisation d'une partition sur un problème réel étant quelque peu limitée. Cependant, ceci permet de se focaliser sur les mécanismes de base de l'apprentissage actif, en les écartant de toutes autres perturbations. Le but est donc d'abord de développer les solutions générales qui pourront ensuite être dérivées pour des cas plus complexes, et ceci dans un environnement contrôlé. De plus, cela nous permettra de mettre en évidence un phénomène particulier et d'en exagérer l'importance et de montrer que certaines des méthodes les plus connues d'apprentissage actif, bien que

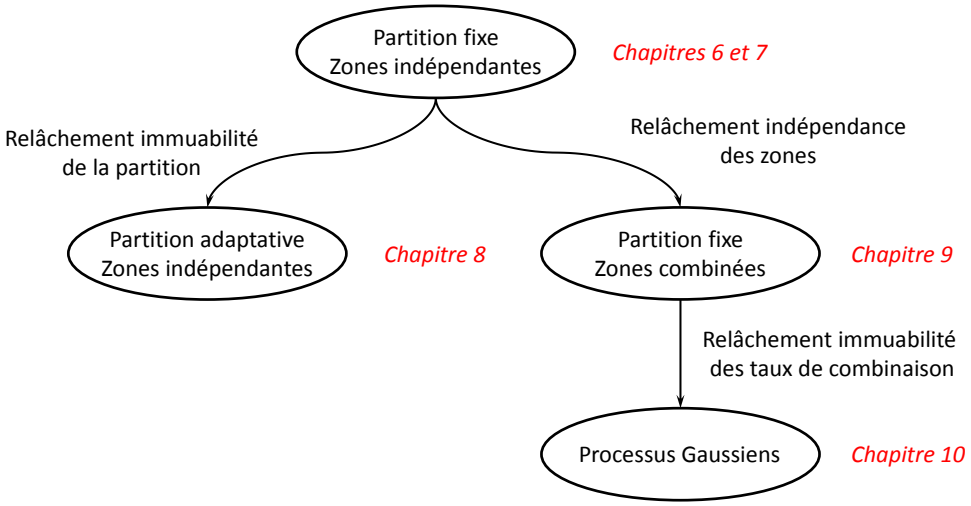


FIGURE 5.1 – Représentation du cheminement effectué dans nos travaux

performantes sur un problème complexe, sont en réalité mal adaptées à la gestion de ce phénomène. Ceci justifie donc notre démarche cherchant à développer une méthode efficace à tous les niveaux du cheminement. L'utilisation partitions fixes permet de ramener un problème continu à un problème discret. L'espace d'entrée est initialement découpé en un certain nombre de zones distinctes et toutes les données d'entrée provenant d'une même zone sont vues et traitées de la même manière. Cette découpe ne peut pas se voir modifiée durant la totalité du processus. La sélection d'une donnée se fait ainsi aléatoirement dans un zone déterminée. Chaque zone est alors vue comme une distribution de Bernoulli sur les étiquettes reçues, et n'est donc conditionnée que par un unique paramètre. L'indépendance entre les zones implique qu'il n'y a potentiellement pas de corrélation entre les étiquettes provenant de différentes zones. Le paramètre de chaque zone ne peut donc être estimée qu'à travers les étiquettes reçues dans sa zone propre. Nous voyons que le problème est dénué de complexité, à la fois par le faible nombre de paramètres (discrets) et par la simplicité avec laquelle les paramètres sont estimés (indépendance), à cela s'ajoute le fait qu'échantillonner une zone n'a pas d'influence sur les autres zones. Dans la suite du cheminement le but va être de relâcher une à une ces contraintes. Une première option est de commencer par relâcher l'immuabilité de la partition. Cependant, cela doit suivre certaines règles. En effet, étant donné qu'il est impossible de différencier les données à l'intérieure d'une même zone, pour que l'estimation des paramètres soit correcte il est nécessaire que les données soient tirées aléatoirement dans chacune des zones. Afin d'être certain que cela soit vérifié à chaque instant du processus, les frontières déjà établies entre les zones ne peuvent pas être déplacées ni supprimées. Il reste toutefois la possibilité de subdiviser les zones existantes. Nous travaillerons donc sur le problème des partition adaptatives qui suivent ce principe. Ceci possède l'avantage par rapport au problème précédent de pouvoir commencer avec un faible nombre de zones et d'augmenter ce nombre au fur et à mesure de l'avancement du processus. Ceci permettant de gérer idéalement le compromis entre vitesse d'apprentissage et limitation des performances. Une deuxième option est de commencer par relâcher l'indépendance des zones tout en conservant l'immuabilité de la partition. En effet, les données que l'on traite proviennent souvent du monde réel et sont donc contraintes par les lois de la physiques, ce qui implique que le paramètre associé à l'oracle ne peut pas varier trop rapidement sur l'espace d'entrée. Ainsi, deux

zones voisines auront tendance à avoir des paramètres proches. Dans ce cas, les étiquettes de chaque zone permettent aussi de donner une indication quant à la valeur des paramètres des zones voisines. En relâchant l'hypothèse d'indépendance, on autorise donc l'estimation des paramètres et donc de la prédiction pour chaque zone à prendre en compte l'ensemble des étiquettes reçues. Le problème devient alors plus complexe car l'échantillonnage dans une zone donnée aura des répercutions sur l'ensemble des prédictions, il faut donc toutes les prendre en compte dans le choix de la zone à échantillonner. Cependant, cela permet d'obtenir une vitesse d'apprentissage plus élevée que sur le problème précédent et ne possède pas la même limitation sur le nombre de zones. En poussant ainsi à l'extrême le nombre de zones, on arrive à un traitement continu des données. Toutefois, la partition n'est toujours pas autorisée à changer de même que les taux de combinaisons, déterminant l'influence des étiquettes provenant d'une zone dans la prédiction d'une autre zone, doivent être fixés avant de débuter le processus. La dernière étape du cheminement consiste alors à relâcher la contrainte d'immuabilité des taux de combinaisons permettant ainsi d'utiliser un classifieur issu de la régression de la fonction de paramètre de l'oracle par un Processus Gaussien. La Figure 5.1 schématise le cheminement en indiquant les contraintes relâchées à chaque étape et les chapitres correspondants.

Voyons maintenant le schéma de construction qui sera toujours suivi par les algorithmes d'apprentissage actif qui seront développés dans les chapitres suivants. Celui-ci s'appuie sur une méthodologie fréquemment employée pour développer un algorithme optimiste qui fonctionne en deux temps, en étudiant d'abord la solution idéale qui serait suivie si l'ensemble des paramètres du problème étaient connus d'avance puis en définissant ensuite une stratégie qui tente de s'en rapprocher avec seulement l'information limitée disponibles sur ces paramètres. En effet, les seules données sur lesquelles l'algorithme peut se baser pour déterminer sa façon d'agir ne sont que des observations partielles des paramètres du problème. Or, construire une stratégie efficace directement à partir de ces données sans prendre en compte cette considération est une tâche ardue. Cette façon de procéder permet de bien comprendre ce que l'on fait en explicitant le comportement recherché ainsi qu'en le distinguant des moyens utilisés pour l'atteindre. Cela permet en outre de se savoir quelles performances maximales peuvent être espérées et donc d'avoir une idée de la marge de manœuvre potentielle de nos algorithmes. Il s'agit donc de procéder en deux temps distincts. Dans un premier temps, nous étudierons et nous définirons un critère idéal, cherchant à répondre à l'objectif fixé, en connaissance des paramètres du problème. Dans notre cas, deux objectifs différents se rapportant au problème d'apprentissage actif pourront être étudiés, ce qui en résultera deux critères différents. Le premier est de minimiser le plus rapidement possible l'erreur espérée maximale associée à la prédiction du classifieur pour n'importe quel donnée d'entrée. Nous verrons que le critère qui permet cela est le risque réel local. Le deuxième est de minimiser le plus rapidement possible le risque réel global du classifieur, qui représente l'erreur de prédiction moyenne sur une donnée d'entrée tirée aléatoirement selon sa distribution naturelle. Nous verrons que le critère qui permet cela est la décroissance du risque réel global. Il s'agira donc d'être capable d'expliciter ces deux critères en fonction de la prédiction courante du classifieur et des paramètres du problème. Dans tous les cas, le critère idéal peut être vu comme la récompense attendue dans le cadre original des bandits à bras multiples. Habituellement, la récompense obtenue est stochastique et l'on cherche à tirer le bras ayant la récompense espérée maximale. Ici, l'erreur réelle, ou bien la décroissance de celle-ci, est déterministe mais une incertitude existe sur les paramètres du problème. Ces deux contextes peuvent cependant être traités de la même manière. Dans un deuxième temps, nous utiliserons le principe de l'optimisme face à l'incertitude pour établir un critère qui tendra à suivre la stratégie idéale. Celui-ci sera donc dérivé du critère idéal défini précédemment, la forme de ce dernier ne pouvant plus se voir modifiée. Rappelons que le principe de l'optimisme face à l'incertitude consiste

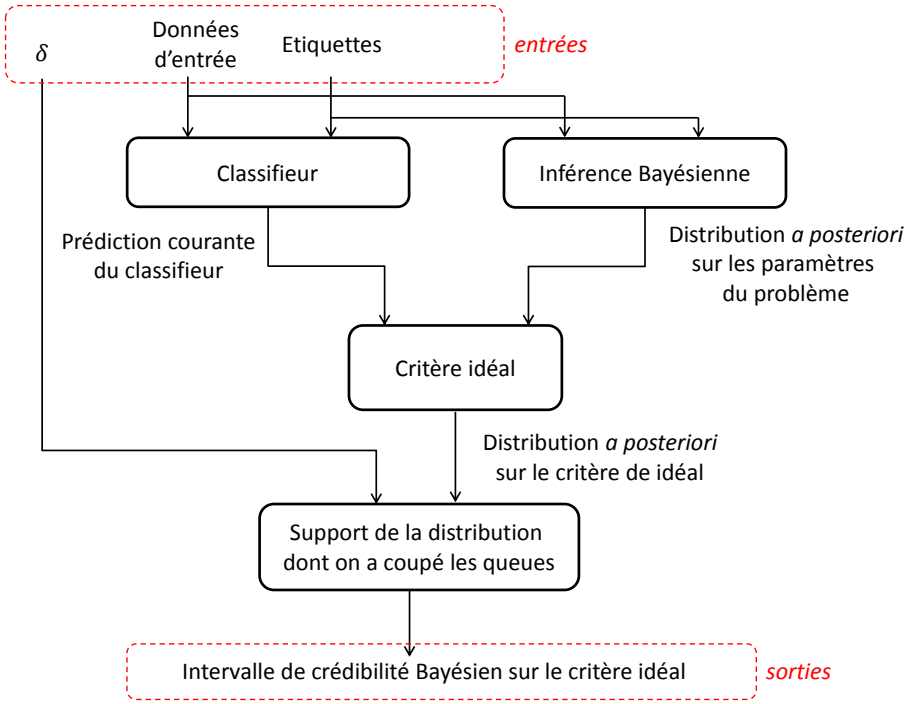


FIGURE 5.2 – Processus de constitution des intervalles de crédibilité Bayésiens sur le critère idéal

à établir une zone d'incertitude, ou intervalle de confiance, autour du critère idéal et d'en ne retenir que la borne supérieure pour former le critère définitif. Dans nos travaux, nous avons préféré opter pour des intervalles de crédibilité Bayésiens. Pour constituer de tels intervalles, il s'agira d'abord de calculer une distribution *a posteriori* sur le critère idéal. Le critère idéal étant une fonction des paramètres du problème, cela consistera à calculer une distribution *a posteriori* sur les paramètres puis de transférer celle-ci au critère idéal. Ensuite, les intervalles de crédibilité Bayésiens doivent contenir le critère idéal avec une certaine probabilité identique pour tous les intervalles. Ainsi, une façon de constituer de tels intervalles est de fixer une certaine probabilité δ puis de couper les queues de la distribution chacune contenant une probabilité $\delta/2$. Le critère de sélection définitif de chacun de nos algorithmes consistera donc en la borne supérieure de ces intervalles de crédibilité Bayésiens. Le processus de constitution de ces intervalles est résumé sur la Figure 5.2. Notez que les intervalles de crédibilité ainsi constituées sont relatives à la probabilité fixée δ. Comme c'est souvent le cas pour les algorithmes optimistes, nos algorithmes prendront donc un paramètre en entrée, celui-ci étant cette probabilité. Nous ne nous sommes pas intéressé à déterminer quelle serait la valeur idéale du paramètre à utiliser. Le but de nos travaux est de montrer que le principe de l'optimisme face à l'incertitude est une solution intéressante pour l'apprentissage actif. Ainsi, nous nous limitons à montrer qu'il est possible d'obtenir des performances intéressantes avec au moins une valeur du paramètre. Lors des évaluations, le paramètre sera donc réglé à l'aide d'une recherche par quadrillage. Une étude théorique des performances de nos algorithmes pourrait révéler la valeur du paramètre idéale à fixer en fonction de la connaissance initiale sur le problème ainsi que du budget d'annotations. Une telle étude fait partie de nos perspectives de recherche proches.

Chapitre 6

Allocation optimale dans une partition fixe

6.1 Introduction

Dans ce chapitre, nous nous intéressons à l'utilisation d'une partition fixe de l'espace d'entrée. Ce problème considère que ce dernier est préalablement découpé en un certain nombre de zones. Toutes les données d'entrée provenant d'une même zone étant confondues. Le classifieur doit alors attribuer à chaque zone une étiquette permettant de prédire au mieux la réponse de l'oracle pour une donnée d'entrée tirée aléatoirement dans cette zone. Notez que cela revient à apprendre les paramètres d'une fonction de prédiction constante par morceau sur les zones. Cette partition possède plusieurs contraintes. D'une part, la partition est donnée initialement comme paramètre de l'algorithme et ne peut pas être redéfinie en cours d'apprentissage, d'où le nom de partition fixe. Ce problème est équivalent à déterminer l'étiquette de données d'entrée discrètes et bruitées, en laissant la possibilité de soumettre une même donnée plusieurs fois à l'oracle dans le but de palier le bruit. D'autre part, les zones sont complètement indépendantes, les étiquettes prédites ne pouvant se baser que sur les observations provenant de leur zone respective. En effet, aucune hypothèse de régularité n'est ici effectuée sur les paramètres de la distribution de l'oracle, il n'y a alors aucun lien entre le paramètre d'une zone et les observations des zones voisines. L'apprentissage actif dans ce contexte consiste à déterminer la zone dans laquelle aller piocher la prochaine donnée à soumettre à l'oracle, ce qui a pour effet d'augmenter la qualité d'estimation du paramètre et donc de la prédiction, uniquement dans la zone correspondante.

Les motivations pour l'utilisation d'une telle partition sont doubles. D'une part, nos travaux s'inspirent et tentent d'étendre ceux effectués dans [13] et [14], lesquels utilisent aussi une partition de l'espace d'entrée. Ce faisant le problème peut être traité comme un problème de bandits à bras multiples dans lequel chaque zone correspond à un bras. D'autre part, cela constitue un problème jouet simple dans lequel les comportements peuvent être facilement interprétés. De plus, les paramètres peuvent être estimés par tirages successifs sans nécessiter d'interpoler à partir des voisins, ce qui permet d'éviter les approximations sur la famille de distributions à considérer lors du calcul de la distribution *a posteriori*.

Ce chapitre constitue une première étape dans l'élaboration d'une méthode efficace d'apprentissage actif permettant de bien cerner les mécanismes utiles pour les ré-exploiter par la suite sur des problèmes plus complexes. Pour l'instant, nous nous limitons à reprendre l'approche adoptée dans [13] et [14], et à l'adapter à la classification. Celle-ci se base sur l'allocation optimale du budget d'annotations, c'est à dire le nombre prédéfini de données à étiqueter dans chaque zone

permettant de maximiser la qualité attendue sur l'ensemble des zones. Ainsi, cela passe par l'évaluation de la qualité de la prédiction attendue pour un certain nombre de données étiquetées en fonction du paramètre de la zone, celle-ci s'exprimant comme le risque moyen obtenu sur tous les tirages possibles. Le but est ensuite de faire en sorte que la répartition des données étiquetées dans les différentes zones se rapproche au maximum de l'allocation optimale tout en apprenant les paramètres au fur et à mesure. Notez que cette allocation optimale ne dépend pas de l'estimation courante des paramètres, ceci fera l'objet d'une amélioration dans le prochain chapitre. Enfin, le principal intérêt de ce chapitre consiste est de montrer qu'il existe un réel intérêt à utiliser un algorithme spécifiquement conçu pour la classification. Notez que les résultats présentés ici ont déjà fait l'objet d'une publication [20].

La différence entre [13]e t [14] provient de la définition de l'objectif. Soit l'allocation optimale a pour objectif de minimiser l'espérance du risque maximale encouru sur les zones. Ou alors elle tente de minimiser l'espérance du risque global, c'est à dire ici une somme pondérée des risques dans chaque zone. Nous reprenons dans nos travaux ces deux types d'objectifs. Les algorithmes qui en découlent étant en tout point identiques hormis le critère idéal utilisé, ces deux approches seront traitées en parallèle.

La suite du chapitre se décompose comme suit. Tout d'abord, nous établirons les allocations optimales pour chaque objectif ainsi que les critères de sélection en-ligne permettant de les atteindre. Ensuite, nous introduirons deux algorithmes permettant de tendre vers ces allocations. Pour cela, nous montrerons comment utiliser les intervalles de crédibilité Bayésiens sur les critères idéaux pour pouvoir utiliser la borne de confiance supérieure en tant que critère de sélection de nos algorithmes. Finalement, nous élaborerons des expériences jouet pouvant représenter n'importe quel problème du monde réel que nous emploierons pour comparer nos algorithmes avec ceux issus de [13] et [14].

6.2 La Partition fixe

Avant tout chose, tachons de poser le problème de la partition fixe. Soit X l'espace d'entrée et Y l'ensemble des étiquettes possibles. Ici, nous nous limitons à la classification binaire, donc $Y = \{0, 1\}$. Soit la partition suivante de l'espace d'entrée en K zones :

$$N = \{X_1, \ldots, X_K\}.$$

Comme toute partition, elle respecte les propriétés suivantes :

- $\forall i \in [\![1, K]\!] : X_i \neq \emptyset$ (aucune zone n'est vide),
- $\cup_{i=1}^{K} X_i = X$ (les zones couvrent l'ensemble de l'espace d'entrée),
- $\forall (i, j) \in [\![1, K]\!]^2 : i \neq j \implies X_i \cap X_j = \emptyset$ (les zones ne se recouvrent pas entre elles).

Les étiquettes reçues en soumettant à l'oracle une donnée d'entrée aléatoirement dans une zone suivent une distribution de Bernoulli. En effet, même si la distribution de l'oracle pour chaque donnée d'entrée dans la zone est différente, les étiquettes prennent toujours obligatoirement la valeur 0 ou 1. Étant donné que la distribution de l'oracle est fixe et que les données d'entrée sont toujours tirées selon la même distribution, alors la distribution de l'oracle associée à la zone entière est fixe aussi. Ainsi, la distribution de l'oracle pour la zone k est caractérisée par un unique paramètre μ_k tel que

$$\mu_k = \mathbb{E}_{x \sim P(x)}[\mu(x) | x \in X_k],$$

on rappelle que $\mu(x)$ correspond au paramètre de l'oracle pour une donnée d'entrée x fixée.

D'autre part, il peut être intéressant d'évaluer l'importance relative de chaque zone dans le risque global. C'est à dire à quel point chaque zone est sollicitée. Celui-ci est noté

$$w_k = \mathbb{P}_{x\sim P(x)}[x \in X_k].$$

La distribution naturelle n'étant pas connue *a priori*, une estimation de celle-ci est utilisée. Puisque l'ensemble des données non-étiquetées peut être utilisé, et que celui-ci ne change pas durant le processus, une estimation de ce dernier est suffisante.

A chaque pas de temps $t \geq 1$, l'algorithme peut demander l'obtention d'une étiquette provenant de la zone $k_t \in \{1, \ldots, K\}$. Alors, une donnée d'entrée est tirée aléatoirement selon la distribution naturelle dans cette zone, puis elle est soumise à l'oracle qui renvoie l'étiquette y_t. Ce procédé n'étant pas pris en compte par l'algorithme, cela revient à tirer une étiquette selon la distribution de l'oracle associée à la zone k_t : $y_t \sim \mathcal{B}er(\mu_{k_t})$. Soit $T_{k,t}$ le nombre de fois que la zone k a été sollicitée par l'algorithme au temps t, ainsi

$$T_{k,t} = \sum_{i=1}^{t} \mathbb{1}_{k_t=k}.$$

C'est donc le nombre d'étiquettes reçues dans cette zone. Soit $\hat{\mu}_{k,t}$ la moyenne des étiquettes reçues dans la zone k au temps t, ainsi

$$\hat{\mu}_{k,t} = \frac{1}{T_{k,t}} \sum_{i=1}^{t} \mathbb{1}_{k_t=k} y_t.$$

Cette moyenne permet donc d'estimer la valeur de μ_k à l'aide des $T_{k,t}$ observations reçues.

Nous venons de définir les notations relatives à la partition, ainsi que les paramètres et les données accessibles par l'algorithme. Dans le problème que nous considérons ici, nous supposons que celle-ci est fixée initialement et est donnée comme argument de l'algorithme. Le but de ce dernier étant alors uniquement d'obtenir la meilleure performance possible sous cette contrainte. Notez que si l'on souhaitait utiliser ce problème en pratique, le choix de la partition ne serait pas triviale. En effet, l'emplacement et la taille de chaque zone est déterminante pour les performances finales. Pourtant, la partition doit être définie alors qu'aucune étiquette n'a encore été observée et ne peut pas changer par la suite, elle ne peut donc se baser que sur la distribution des données non-étiquetées. De plus, le nombre de zones doit être calibré pour obtenir la meilleure vitesse d'apprentissage. Cependant, rappelons qu'il ne s'agit ici que d'une étude préliminaire, cette limitation n'étant donc pas problématique. De plus, cette limitation n'interviendra pas pour les problèmes qui seront étudiés dans la suite puisqu'ils pourront voir leur partition être définie au fil des annotations, correspondant ainsi plus exactement à la répartition des classes dans l'espace d'entrée, ou bien ne pas voir leur vitesse d'apprentissage limitée par le nombre de zones, permettant ainsi de discrétiser l'espace beaucoup plus finement.

6.3 Définition du critère d'échantillonnage en connaissance parfaite

Dans cette section, nous définissons deux sortes d'allocations optimales du budget d'annotations sur les zones en fonction de l'objectif souhaité. Ici, les paramètres du problème seront supposés connus. Bien évidement, cette allocation optimale idéale n'a pas pour but d'être utilisée directement pour résoudre le problème. C'est en effet impossible car les paramètres du problème ne sont, pour un problème réel, pas connus à l'avance. Il s'agit cependant d'une étape intermédiaire

nécessaire pour ensuite servir de cible pour la stratégie de sélection définitive qui se basera, elle, uniquement sur l'estimation de ces paramètres obtenues à l'aide des observations reçues. Nous exprimerons aussi les critères de sélection en-ligne menant à ces allocations optimales afin de pouvoir en dériver les critères définitifs. Ceux-ci ont aussi accès aux vrais paramètres pour la décision de la prochaine zone à échantillonner. Notez que même dans ce contexte, avec une connaissance parfaite des paramètres du problème, le classifieur, quant à lui, n'a pas accès à cette information, auquel cas il n'aurait pas besoin d'apprendre à partir des observations. Il faut voir le classifieur et l'algorithme d'apprentissage actif en connaissance parfaite comme deux entités distinctes. La première n'a accès qu'aux données acquises et effectue la prédiction qui lui semble la meilleure, tandis que la deuxième connais les vrais paramètres et lui fournit les données qui le guideront vers la meilleure prédiction. Dans le cas présent, le critère de sélection en connaissance parfaite n'a pas accès aux données effectivement reçues par le classifieur et peut seulement en avoir une idée en fonction des paramètres.

Les deux allocations que nous allons définir tentent de minimiser une fonction de perte dans chaque zone basée sur l'espérance du risque réel. Nous commençons par définir cette perte, regardons donc d'abord le risque réel encouru par une zone k de paramètre μ_k ayant reçu $T_{k,t}$ données étiquetées. Tout d'abord, supposons que le classifieur attribue toujours l'étiquette majoritaire dans chaque zone. Soit $f_{k,t}$ l'étiquette prédite par le classifieur f_t pour la zone k au temps t. Ainsi, $\forall k \in \{1, \ldots, K\}$,

$$f_{k,t} = \lfloor \hat{\mu}_{k,t} \rceil,$$

où $\lfloor . \rceil$ est l'opérateur d'arrondi.

Le risque réel $R_k(f_t)$ associé à la zone k en utilisant la perte binaire $l_{0/1}$ est alors

$$R_k(f_t) = \mu_k \mathbb{1}_{f_{k,t}=0} + (1 - \mu_k)\mathbb{1}_{f_{k,t}=1}.$$

Ceci est simplement le taux de données mal classifiées, si le classifieur attribue la valeur 0, il se trompe sur toutes les données étant étiquetées 1 par l'oracle, donc en espérance, sur une proportion μ_k de données, et inversement si il attribue la valeur 1. Il est important de noter que la valeur optimale de l'étiquette attribuée par le classifieur est $\lfloor \mu_k \rceil$, et que celle-ci engendre un risque non-nul. En effet, $R_k(f^*) = \frac{1}{2} - |\mu_k - \frac{1}{2}|$, avec f^* le classifieur qui attribue l'étiquette optimale à chaque zone et $|.|$ l'opérateur de valeur absolue. Afin d'éviter que le critère de sélection optimal sélectionne une zone avec un risque fort dans le but de le faire décroitre alors que c'est impossible, le risque optimal est soustrait du risque courant. Ainsi,

$$R_k(f_t) - R_k(f^*) = 2|\mu_k - \frac{1}{2}|\mathbb{1}_{f_{k,t} \neq \lfloor \mu_k \rceil}.$$

Le risque réel n'est pas connu par l'algorithme de sélection en connaissance parfaite, car il n'a pas accès au résultat des tirages. Cependant, il sait que chaque étiquette reçue dans la zone k a été tirée selon $\mathcal{B}er(\mu_k)$, alors $T_{k,t}\hat{\mu}_{k,t}$ suit une loi de Bernoulli de paramètres $T_{k,t}$ et μ_k. Nous définissons la fonction de perte $L_{k,t}$ pour la zone k comme l'espérance de cette différence sur les tirages possibles. Ainsi,

$$\begin{aligned} L_{k,t}(\mu_k, T_{k,t}) &= \mathbb{E}[R_k(f_t) - R_k(f^*)] = 2|\mu_k - \frac{1}{2}|\mathbb{P}(f_{k,t} \neq \lfloor \mu_k \rceil) \\ &= 2|\mu_k - \frac{1}{2}|\mathbb{1}_{\lfloor \mu_k \rceil = 0}(1 - \mathcal{I}_{1-\mu_k}(T_{k,t} - \lfloor \frac{T_{k,t}}{2} \rfloor, \lfloor \frac{T_{k,t}}{2} \rfloor + 1)) \\ &\quad + \mathbb{1}_{\lfloor \mu_k \rceil = 1}\mathcal{I}_{1-\mu_k}(T_{k,t} - \lfloor \frac{T_{k,t}}{2} \rfloor, \lfloor \frac{T_{k,t}}{2} \rfloor + 1), \end{aligned}$$

avec $\mathcal{I}_x(a,b)$ la fonction Beta incomplète de paramètres a et b en $x \in [0,1]$, et $\lfloor . \rfloor$ l'opérateur de partie entière.

Deux sortes d'allocation peuvent alors être définies en fonction de l'objectif désiré et de la manière dont les fonctions de perte sur chaque zone sont combinées dans la définition de la fonction objectif globale. Soit le but est que quelle que soit la zone, la perte soit minimale. C'est à dire que la perte maximale à travers les zones doit être minimum. Cela se traduit par la fonction objectif suivante :

$$L_t^m = \max_{k \in \{1,\dots,K\}} L_{k,t}(\mu_k, T_{k,t}).$$

Soit le but est qu'en tirant une donnée d'entrée selon la distribution naturelle, et donc en tirant une zone avec une probabilité proportionnelle à son importance relative w_k, l'espérance de la perte soit minimum. Ce qui se revient à minimiser la fonction objectif suivante :

$$L_t^s = \sum_{k=1}^{K} w_k L_{k,t}(\mu_k, T_{k,t}).$$

Notez que ces deux fonctions objectifs peuvent être rapprochées des approches d'échantillonnage d'incertitude et de réduction de l'erreur abordées dans l'état de l'art.

Pour minimiser la fonction objectif L_t^m, dans le cas où les zones sont indépendantes et que la fonction de perte $L_{k,t}$ est décroissante avec $T_{k,t}$, la solution est de sélectionner la zone pour laquelle la perte est maximale :

$$k_{t+1}^m = \arg\max_{k \in \{1,\dots,K\}} L_{k,t}(\mu_k, T_{k,t}).$$

En effet, puisque les zones sont indépendantes, sélectionner une autre zone que celle pour laquelle le risque est maximal ne permettra pas de modifier la perte associée à cette dernière, et donc la perte maximale restera la même. D'autre part, si la perte est décroissante, alors il est garanti que ni cette zone ni aucune autre ne possèdera dans le futur une perte plus élevée. Si cela n'avait pas été le cas, il aurait pu être préférable de sélectionner une autre zone qui verra sa perte augmenter dans le futur afin de garantir une perte maximale le plus tôt possible.

Pour minimiser la fonction objectif L_t^s, dans le cas où les zones sont indépendantes et où la fonction de perte est strictement décroissante et convexe avec $T_{k,t}$, la solution est de sélectionner la zone pour laquelle la décroissance de la perte est maximale :

$$k_{t+1}^s = \arg\min_{k \in \{1,\dots,K\}} w_k \Delta L_{k,t}(\mu_k, T_{k,t}),$$

avec $\Delta L_{k,t}(\mu_k, T_{k,t}) = L_{k,t}(\mu_k, T_{k,t}+1) - L_{k,t}(\mu_k, T_{k,t})$. En effet, tout d'abord la fonction objectif est alors décroissante, il s'agit donc de la faire décroitre le plus rapidement possible. Puisque les zones sont indépendantes, la décroissance de la fonction objectif est égale à la décroissance de la zone sélectionnée. Ensuite, puisque la perte est convexe, alors $\forall t_2 > t_1, \quad \Delta L_{k,t_2}(\mu_k, T_{k,t_2}) > \Delta L_{k,t_1}(\mu_k, T_{k,t_1})$. Donc, aucune autre zone n'engendrera une décroissance plus grande dans le futur. Il est donc préférable de dépenser une unité de budget à l'instant courant dans cette zone plutôt que dans n'importe quelle autre zone à n'importe quel moment dans le futur.

Le problème est ici que la fonction de perte définie précédemment n'est pas décroissante, ceci étant dû à la partie entière présente dans l'équation. Pour remédier à cela, nous remplaçons la fonction de perte par une borne sur cette dernière qui est décroissante et convexe. L'allocation optimale aura alors pour objectif de minimiser cette borne plutôt que la fonction objectif précédente.

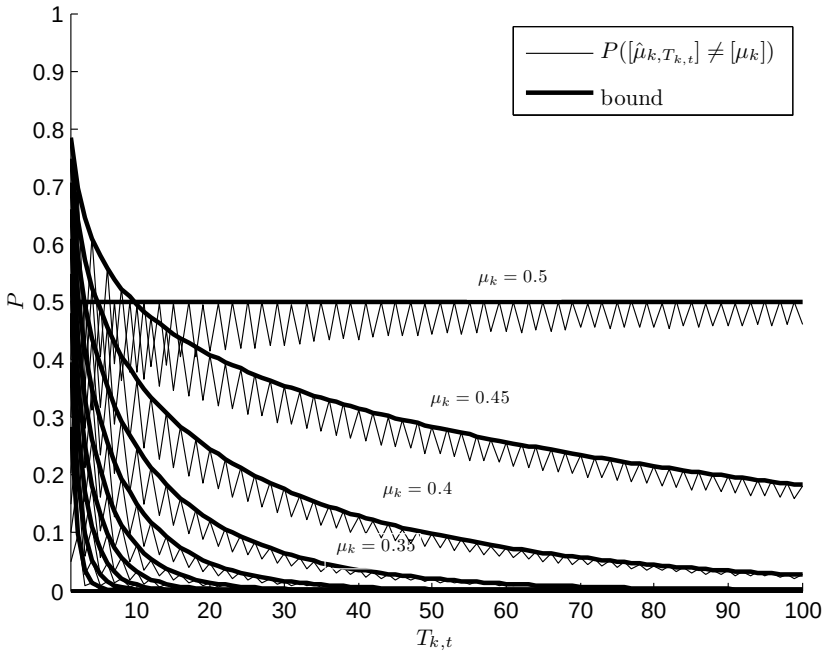


FIGURE 6.1 – $\mathbb{P}(\lfloor\hat{\mu}_{k,t}\rceil \neq \lfloor\mu_k\rceil)$ and its bound defined in (6.1)

L'idée est alors d'encadrer la partie entière, les bornes obtenues de cette façon sont extrêmement étroites et sont égales en un nombre de points infiniment dénombrables. Ainsi,

$$L_{k,t}(\mu_k, T_{k,t}) \leq 2|\mu_k - \frac{1}{2}|\mathbb{1}_{\lfloor\mu_k\rceil=0}(1 - \mathcal{I}_{1-\mu_k}(\frac{T_{k,t}}{2}+1, \frac{T_{k,t}}{2})) + \mathbb{1}_{\lfloor\mu_k\rceil=1}\mathcal{I}_{1-\mu_k}(\frac{T_{k,t}}{2}, \frac{T_{k,t}}{2}+1) \tag{6.1}$$

$$= \tilde{L}_{k,t}(\mu_k, T_{k,t}).$$

Les nouvelles fonctions objectifs sont alors

$$\tilde{L}_t^m = \max_{k\in\{1,\dots,K\}} \tilde{L}_{k,t}(\mu_k, T_{k,t})$$

et

$$\tilde{L}_t^s = \sum_{k=1}^{K} w_k \tilde{L}_{k,t}(\mu_k, T_{k,t}).$$

La figure 6.1 montre la fonction de perte $L_{k,t}$ et sa borne $\tilde{L}_{k,t}$ pour plusieurs valeurs de μ_k. Nous pouvons voir que la borne est extrêmement étroite et préserve le comportement de la fonction de perte tout en respectant les conditions nécessaires.

L'allocation optimale d'un budget de n annotations en utilisant la fonction objectif $\tilde{L}_t^m$ est telle que la perte dans chaque zone soit égale. En effet, si ce n'est pas le cas, la partie du budget ayant servi à la zone avec la perte la plus faible à dépasser les autres aurait pu être redistribuée sur les autres zones pour faire descendre la perte maximale. Notez qu'ici on s'autorise un nombre d'annotations non-entier, ce qui est possible car la fonction de perte est définie $\forall T_{k,t} \in \mathbb{R}^+$. Ceci n'est pas possible en pratique mais tentera d'approcher cette allocation optimale. Ainsi l'allocation

optimale $T^*_{k,n}$ sous le budget n est telle que

$$\begin{cases} \forall k \in \{1, \dots, K\}, \exists c, \tilde{L}_{k,n}(\mu_k, T^*_{k,n}) = c \\ \sum_{k=1}^{K} T^*_{k,n} = n. \end{cases}$$

De même, en s'autorisant un nombre d'annotations non-entier, l'allocation optimale d'un budget de n annotations en utilisant la fonction objectif $\tilde{L}^s_t$ est telle que

$$\begin{cases} \forall k \in \{1, \dots, K\}, \exists c, w_k \frac{\partial \tilde{L}_{k,n}}{\partial T_{k,n}}(\mu_k, T^*_{k,n}) = c \\ \sum_{k=1}^{K} T^*_{k,n} = n. \end{cases}$$

L'allocation optimale peut être difficile à calculer en tant que telle, il peut être préférable de passer par une version en-ligne. De plus, un algorithme basé sur l'optimisme face à l'incertitude nécessite l'expression d'un critère de sélection en-ligne. Dans notre cas, le critère de sélection en connaissance parfaite menant à l'allocation optimale du budget d'annotations s'exprime respectivement comme

$$C^m_{k,t}(\mu_k, T_{k,t}) = \tilde{L}_{k,t}(\mu_k, T_{k,t}) \tag{6.2}$$

et

$$C^s_{k,t}(\mu_k, T_{k,t}) = w_k \Delta \tilde{L}_{k,t}(\mu_k, T_{k,t}). \tag{6.3}$$

Et ainsi, les stratégie de sélection idéales pour les deux objectifs fixés sont les suivantes :

$$k^m_{t+1} = \underset{k \in \{1,\dots,K\}}{\arg\max}\ C^m_{k,t}(\mu_k, T_{k,t})$$

et

$$k^s_{t+1} = \underset{k \in \{1,\dots,K\}}{\arg\max}\ C^s_{k,t}(\mu_k, T_{k,t}).$$

Dans la section suivante, nous allons voir comment cette allocation peut être approchée alors que la valeur des paramètres est inconnue.

6.4 Prise en compte de l'incertitude

Nous venons de définir l'allocation optimale qui détermine comment les données étiquetées doivent être réparties sur les différentes zones de l'espace d'entrée dans le cas où les paramètres de la distribution de l'oracle sont connus afin d'obtenir la meilleure performance du classifieur. Pourtant, ceux-ci ne sont pas connus initialement et doivent être appris en cours d'exécution. L'allocation optimale va donc servir de cible à atteindre, cela demandant une gestion du compromis entre exploration et exploitation, pour savoir à quel moment privilégier l'apprentissage des paramètres ou leur utilisation pour tendre vers l'allocation optimale. Ici, nous montrons que le principe de l'optimisme face à l'incertitude peut être utilisé dans ce but. L'idée est alors d'utiliser un algorithme de borne supérieure de confiance en utilisant des intervalles de crédibilité Bayésiens définis sur les deux critères de sélection en connaissance parfaite de la section précédente.

Les deux algorithmes résultants partagent le même corps, qui est représenté sur la Figure 3. Ceux-ci peut être dérivés en précisant le critère de sélection en connaissance parfaite utilisé. Ils prennent en argument un paramètre δ qui permet de régler la probabilité relative aux intervalles de crédibilité. Plus cette probabilité est grande, plus l'étendue des valeurs crédibles du paramètre est large. Elle permet donc en outre de contrôler le ratio entre exploration et exploitation. Dans la

Données : N, $(w_k)_{k\in\{1,\dots,K\}}$, δ
Initialisation : $\forall k \in \{1,\dots,K\}$, $\hat{\mu}_{k,0} = \frac{1}{2}$, $T_{k,0} = 0$
pour *t = 1, ..., n* **faire**
  Calculer $e^{sup}_{k,t}(\delta)$ à partir de (6.5) et (6.4) avec $C_{k,t}$ le critère issu de (6.2) ou (6.3)
  Échantillonner la zone $k_t = \arg\max_k e^{sup}_{k,t}(\delta)$
  Recevoir $y_t \sim \mathcal{B}er(\mu_{k_t})$
  Mettre à jour $\hat{\mu}_{k_t,t}$ et $T_{k_t,t}$
fin
Résultat : $\lfloor \hat{\mu}_{k,n} \rceil$ pour chaque zone $k \in \{1,\dots,K\}$

Algorithme 3 : Algorithme générique

suite, nous détaillons le fonctionnement de l'algorithme et comment calculer le critère de sélection final. Les intervalles de crédibilité Bayésiens sont l'équivalent Bayésien des intervalles de confiance fréquentistes, ils contiennent la vraie valeur de l'objet considéré avec une certaine probabilité. Pour les construire, il est nécessaire de pouvoir obtenir une distribution *a posteriori* de l'objet considéré. Le critère de sélection en connaissance parfaite pour la zone k est une fonction de μ_k. Il possible de calculer simplement une distribution *a posteriori* sur μ_k à partir des observations reçues jusqu'au temps t puis de transférer celle-ci sur le critère idéal.

Étant donné que les étiquettes reçues dans la zone k sont tirées selon la loi de Bernoulli $\mathcal{B}er(\mu_k)$, la moyenne estimée $\hat{\mu}_{k,t}$ provient d'une loi Binomiale $\mathcal{B}in(T_{k,t}, \mu_k)$. Rappelons que la famille de lois Beta fourni une famille de lois *a priori* conjuguées pour la loi Binomiale. Ainsi, si la loi uniforme $\mathcal{B}eta(1,1)$ est prise comme distribution *a priori* pour le paramètre μ_k, alors par inférence Bayésienne nous obtenons la distribution *a posteriori* suivante :

$$\mathbb{P}(\mu_k = x|\hat{\mu}_{k,t}, T_{k,t}) = \frac{x^{T_{k,t}\hat{\mu}_{k,t}}(1-x)^{T_{k,t}(1-\hat{\mu}_{k,t})}}{Beta(T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1)},$$

avec $Beta(a,b)$ la fonction Beta de paramètres a et b.

Nous avons donc une distribution de probabilité sur la valeur de μ_k, puisque le critère de sélection en connaissance parfaite est une fonction de μ_k, nous pouvons donc en déduire une distribution de probabilité sur la valeur de celui-ci. Dans la suite $C_{k,t}$ représente soit $C^m_{k,t}$ de (6.2) soit $C^s_{k,t}$ de (6.3). Naturellement,

$$\mathbb{P}(C_{k,t}(\mu_k, T_{k,t}) > e_{k,t}|\hat{\mu}_{k,t}, T_{k,t}) = \mathbb{P}(\mu_k, C_{k,t}(\mu_k, T_{k,t}) > e_{k,t}|\hat{\mu}_{k,t}, T_{k,t}).$$

Soit $I_{k,t} = \{\mu_k | C_{k,t}(\mu_k, T_{k,t}) > e_{k,t}\}$, alors

$$\mathbb{P}(C_{k,t}(\mu_k, T_{k,t}) > e_{k,t}|\hat{\mu}_{k,t}, T_{k,t}) = \frac{\int_{x\in I_k} x^{T_{k,t}\hat{\mu}_{k,t}}(1-x)^{T_{k,t}(1-\hat{\mu}_{k,t})}dx}{Beta(T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1)}. \tag{6.4}$$

L'idée est alors de se servir de la distribution de probabilité sur le critère pour construire un intervalle qui le contient avec une certaine probabilité. Pour cela, la solution la plus simple est de couper les queues de la distribution. Ainsi, soit $e^{inf}_{k,t}$ tel que $\forall \delta \in [0, \frac{1}{2}]$,

$$\mathbb{P}(C_{k,t}(\mu_k, T_{k,t}) \leq e^{inf}_{k,t}(\delta)|\hat{\mu}_{k,t}, T_{k,t}) = \delta,$$

et $e^{sup}_{k,t}$ tel que $\forall \delta \in [0, \frac{1}{2}]$,

$$\mathbb{P}(C_{k,t}(\mu_k, T_{k,t}) \geq e^{sup}_{k,t}(\delta)|\hat{\mu}_{k,t}, T_{k,t}) = \delta, \tag{6.5}$$

alors l'intervalle $[e^{inf}_{k,t}(\delta), e^{sup}_{k,t}(\delta)]$ contient $C_{k,t}(\mu_k, T_{k,t})$ avec un probabilité $1 - 2\delta$.

Ainsi, en suivant le procédé d'un algorithme de borne de confiance supérieure, la prochaine zone à échantillonner est celle pour laquelle la borne supérieure de l'intervalle de crédibilité Bayésien est la plus haute. Ainsi,

$$k_{t+1} = \underset{k \in \{1,\dots,K\}}{\arg\max}\ e^{sup}_{k,t}(\delta).$$

Notez que l'équation ci-dessus est générique et peut être dérivée pour les deux critères de sélection en connaissance parfaite possibles.

6.5 Expériences

Dans cette section, nous définissons les expériences qui serviront à évaluer les algorithmes introduits dans la section précédente et à les comparer avec les algorithmes de l'état de l'art. L'intérêt est de savoir si nos algorithmes sont efficaces sur un jeu de données tirées du monde réel. De part la nature du problème que nous nous sommes posé ici, à savoir les partitions fixes avec des zones indépendantes, de nombreux jeux de données peuvent être vus et traités exactement de la même manière par l'algorithme. En effet, puisque la partition est fournie en entrée de l'algorithme et que celle-ci ne peut pas changer au cours du temps, alors chaque problème est caractérisé uniquement par le paramètre de la distribution de l'oracle et l'importance relative dans chaque zone de la partition. La répartition des données dans chaque zone est, elle, indifférente à l'algorithme. Que la zone contienne deux régions où les étiquettes sont différentes mais bien déterminées, ou que chaque donnée d'entrée soit également bruitée, tant que la valeur du paramètre est identique, la zone est perçue de la même manière. Ensuite, des partitions différentes peuvent également être perçues de la même manière. En effet, si deux partitions ont le même nombre de zones, mais disposées différemment dans l'espace d'entrée, et que les zones d'une partitions possèdent les même paramètres et la même importance relative que celles de l'autre (l'ordre n'important pas), alors l'algorithme les traitera exactement de la même manière. En effet, ceci est dû à l'indépendance des zones de la partition qui rend leur positions relatives dans l'espace d'entrée inaccessible par l'algorithme. Finalement, un problème n'est caractérisé que par les valeurs suivantes :

- le nombre de zones K,
- les paramètres de la distribution de Bernoulli associée à chaque zone $(\mu_k)_{k \in \{1,\dots,K\}}$,
- les importances relatives de chaque zone $(w_k)_{k \in \{1,\dots,K\}}$.

Comme expliqué ci-dessus, n'importe quel problème du monde réel qui se verrait attribué une partition rentre dans cette définition. Ainsi, afin d'évaluer nos algorithmes sur un très large ensemble de problèmes, nous créons un problème-jouet en générant aléatoirement un grand nombre de problèmes à partir de la définition ci-dessus. Ainsi, un algorithme performant sur le problème-jouet est probablement performant sur n'importe quel problème, même provenant du monde réel, qui utiliserait ce type de classifieur.

Afin d'évaluer nos algorithmes, nous créons deux bancs de tests. Le premier consiste à générer un grand nombre de problèmes différents en tirant aléatoirement les paramètres de la définition ci-dessus. Ensuite les algorithmes à évaluer sont lancés sur chacun de ces problèmes, et le risque réel est enregistré à chaque pas de temps. Notez qu'ici, le risque réel peut être calculé directement puisque les paramètres sont générés artificiellement et sont donc connus, contrairement au cas où seuls un nombre limité de données sont accessibles et un jeu de test doit être utilisé. De la même

manière, les étiquettes renvoyées par l'oracle sont directement tirées de la distribution correspondante, plutôt que sélectionnées dans un réservoir. La performance globale affichée sur les courbes, qui nous permet de comparer les algorithmes entre eux, est calculée pour chaque pas de temps comme la moyenne des risques réels obtenus sur chaque problème. Dans ce banc de tests, 1000 problèmes ont été générés avec

- K tiré uniformément dans $[\![0, 50]\!]$,
- $\forall k \in \{1, \dots, K\}, \quad \mu_k$ est tirée uniformément dans $[0, 1]$,
- $\forall k \in \{1, \dots, K\}, \quad w_k$ est tirée uniformément dans $[0, 1]$.

Le budget utilisé est de 300 annotations.

Le second banc de tests que nous définissons consiste à n'utiliser qu'un seul problème bien choisi. En effet, dans le premier banc de tests, les performances sur chaque problème sont moyennées, cela ne permet pas d'interpréter le comportement des algorithmes. En particulier, le nombre de zones dans chaque problème est différents, et puisque la vitesse d'apprentissage en dépend, chacun d'eux est donc soumis a des échelles de temps différentes. Moyenner ces différentes échelles de temps ne permet donc pas de rendre compte des phénomènes temporels. De plus, ici, chaque zone possède la même importance relative afin de se concentrer sur la façon dont le critère de sélection gère des zones de paramètres différents. Ainsi, le problème considéré est le suivant :

- $K = 10$,
- $\forall k \in \{1, \dots, K\}, \quad \mu_k = \frac{k}{K}$,
- $\forall k \in \{1, \dots, K\}, \quad w_k = \frac{1}{K}$.

Dans ce problème, les zones ont des paramètres aussi différents que possible, en effet, pour pouvoir juger de la qualité du critère d'allocation, il faut qu'il y ait une différence dans le nombre de données à allouer à chaque zone. De plus, ceci nous permet d'entrevoir comment l'algorithme se comporte sur l'ensemble des paramètres. Les algorithmes sont lancés 1000 fois sur ce problème et les performances sont moyennées. Le budget utilisé est de 300 annotations, ce qui correspond à une moyenne de 30 étiquettes par zone, ce qui est en général suffisant pour avoir une bonne estimation du paramètre.

Le but de ce chapitre est de montrer que les travaux présentés dans [13] et [14] peuvent être adaptés avec succès à la classification. Pour voir l'intérêt que cela porte, nous comparons nos algorithmes à ceux issus de ces travaux dont l'objectif est de l'apprentissage actif pour de la régression dans une partition fixe. Ceux-ci sont tout de même utilisés dans notre problème de classification. Ainsi, l'allocation du budget a pour but de régresser les paramètres de la distribution de l'oracle, dont l'estimation va être utilisée pour prédire les étiquettes. Nous comparons aussi à un algorithme récent d'apprentissage actif pour voir si notre méthode est compétitive. Les algorithmes utilisés pour la comparaison sont les suivants :

- **Échantillonnage aléatoire** : la zone sélectionnée est tirée avec une probabilité proportionnelle à son importance relative, cela revient à tirer une donnée d'entrée selon la distribution naturelle,
- **CH-AS** [13] : algorithme utilisant le principe de l'optimisme face à l'incertitude pour le problème d'apprentissage actif dans le contexte de la régression, la fonction objectif utilisée est la perte maximale sur les zones,

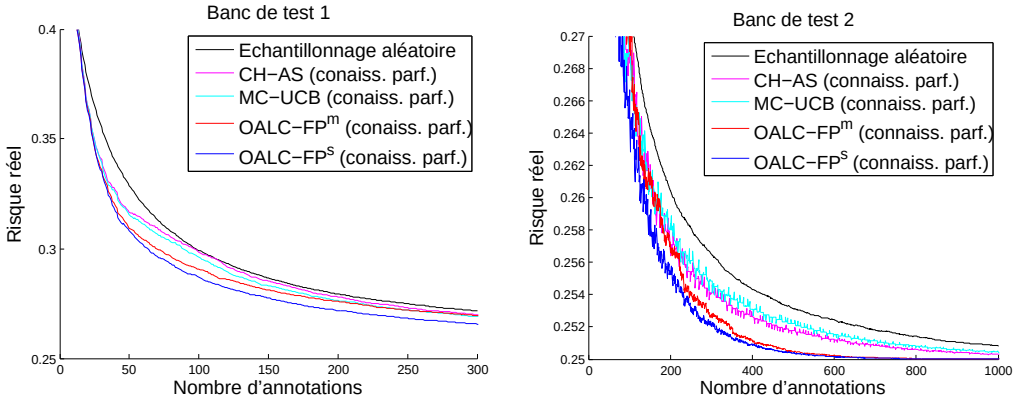


FIGURE 6.2 – Évaluation des critères de sélection en connaissance parfaite

- **MC-UCB** [14] : algorithme utilisant le principe de l'optimisme face à l'incertitude pour le problème d'apprentissage actif dans le contexte de la régression, la fonction objectif utilisée est la somme des perte sur les zones pondérée par leur importance relative,

- **EffECXtive** [34] : algorithme d'apprentissage actif utilisant la notion de classe d'équivalence et définissant des poids entre celles-ci, il tente ensuite de déterminer le plus rapidement la classe d'équivalence contenant le classifieur idéal.

Pour tous les algorithmes optimistes utilisés, la valeur du paramètre δ permettant de contrôler la quantité d'exploration a été réglée en effectuant une recherche par quadrillage.

6.6 Résultats

Voyons maintenant les résultats des expériences. Comme nous avons pu le voir dans les sections précédentes, les algorithmes que nous avons introduits, comme la plupart des algorithmes utilisant le principe d'optimisme face à l'incertitude, ont été construits en deux étapes. Tout d'abord le critère de sélection a été défini en connaissance parfaite, puis est venue s'ajouter l'utilisation d'intervalles de crédibilité pour pouvoir suivre ce critère sans que les paramètres ne soient connus à l'avance. Afin d'évaluer la justesse de nos algorithmes, il est nécessaire de vérifier chacune des étapes séparément. Nous commençons donc par montrer les résultats des expériences menées avec le critère de sélection en connaissance parfaite. Ceux-ci sont disponibles sur la Figure 6.2, le critère idéal de chaque algorithme a été utilisé respectivement, la mention *connaiss. parf.* apparaissant pour les différencier de l'algorithme original. Notez que l'allocation ne dépend pas des étiquettes reçues mais uniquement des paramètres de chaque zone. Ainsi, une seule exécution de l'algorithme est nécessaire, le risque espéré pouvant être calculé directement à partir du paramètre et du nombre d'échantillons alloués à chaque zone. Le bruit dans la courbe correspondant au second banc de test n'est donc pas dû à un faible nombre d'exécutions mais au caractère discret de l'estimateur des paramètres, nous pouvons voir que celui-ci s'atténue progressivement. Ceci n'est pas le cas du premier banc de test car les performances ont été moyennées sur plusieurs problèmes. Pour le second banc de test, le budget à été étendu à 1000 pour pouvoir étudier le comportement à long terme.

Nous pouvons voir que les performances des critères de sélection spécialement conçus pour la classification sont naturellement meilleures que celles de ceux conçus pour la régression. Ceci est flagrant sur le second banc de test où le risque de ces deux premiers chute beaucoup plus rapidement

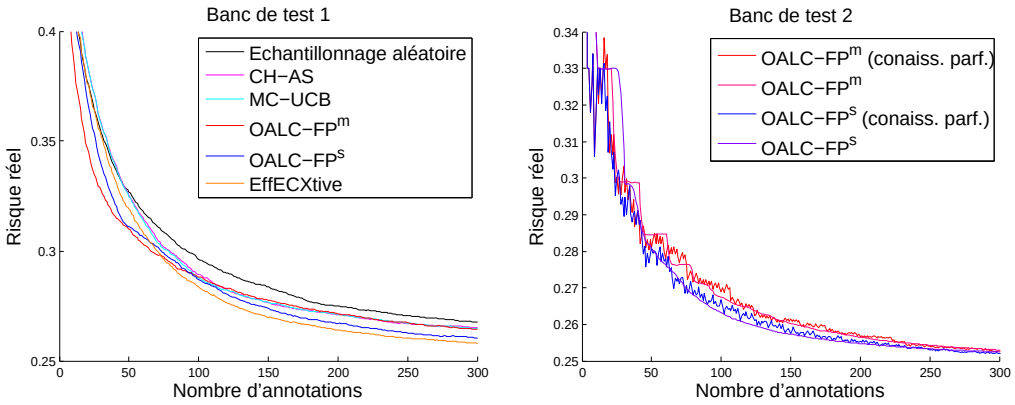


FIGURE 6.3 – Évaluation des algorithmes sans connaissance parfaite

que pour les deux derniers. Ainsi, OALC-FP^s atteint en 490 annotations la performance de MC-UCB en 1000 annotations. De plus, les performances du critère pour lequel la fonction objectif est basée sur la somme des risques espérés dans chaque zone sont toujours meilleures que celles du critère basé sur le risque espéré maximal. En effet, la façon dont la performance est évaluée ici correspond exactement à cette première fonction objectif. Il est donc naturel que le critère adapté possède de meilleures performances.

Maintenant que le critère de sélection en connaissance parfaite a montré qu'il donnait en effet les résultats attendus, et qu'il engendrait donc la meilleure allocation des données en fonction des paramètres idéaux, il faut vérifier que la deuxième étape permet bien d'atteindre cette cible rapidement. Rappelons que pour atteindre cette allocation optimale, il faut prendre en compte l'estimation des paramètres, mais que pour que cette estimation soit juste il faut avoir au moins un certain nombre d'observations, ce qui implique de s'écarter de l'allocation optimale. Il s'agit donc ici de vérifier si le compromis entre exploration et exploitation est bien équilibré. La Figure 6.3 donne les résultats des expériences menées sur les algorithmes finaux. Ceux-ci n'ayant pas accès aux vrais paramètres et devant les estimer à travers les étiquettes reçues. Sur le second banc de tests, ceux-ci sont comparés à leur version en connaissance parfaite. Nous pouvons voir que très rapidement, l'écart entre les performances des deux versions se réduit. Après environ 50 annotations, les deux courbes sont presque superposées. Cela confirme donc que l'utilisation du principe de l'optimisme face à l'incertitude avec des intervalles de crédibilité Bayésiens permet de gérer efficacement l'incertitude sur les paramètres dans le critère de sélection en connaissance parfaite.

Sur le premier banc de tests, nous pouvons voir que nos algorithmes possèdent de meilleures performances que $CH - AS$ et $MC - UCB$ desquels ils sont inspirés. Pourtant, lorsque ceux-ci sont comparés à un algorithme conçu pour l'apprentissage actif pour la classification, EffEXCtive, nous pouvons voir que les performances sont moins bonnes. Ceci pourrait sembler étonnant car l'allocation du budget en connaissance parfaite a été conçue pour être optimale, c'est à dire qu'il n'existe pas de meilleure allocation, et la version sans connaissance parfaite arrive à atteindre cette allocation. Ceci est en réalité logique puisque l'allocation que l'on se fixe comme but dépend uniquement des paramètres de chaque zone et non de l'état courant du classifieur. Ainsi, quelle que soit l'exécution de l'algorithme, et les observations reçues, l'allocation sera toujours la même, l'allocation optimale ne cherchant à minimiser que le risque espéré sur les observations reçues. Dans le cas d'EffEXCtive, le critère de sélection dépend de la valeur des étiquettes reçues, il permet donc de s'adapter à la prédiction courante du classifieur. Bien que notre algorithme final dépende des

observations, il ne les utilise que dans le but d'atteindre l'allocation prédéfinie.

6.7 Conclusion intermédiaire

Dans ce chapitre, il a été question d'effectuer une première approche de l'application du principe de l'optimisme face à l'incertitude dans le problème d'apprentissage actif pour la classification. Pour cela, nous avons étudié le problème simplifié des partitions fixes dans lesquelles les zones sont entièrement indépendantes. En s'inspirant des travaux effectués dans [13] et [14] pour le cas de l'apprentissage actif pour la régression, nous avons développé deux algorithmes répondant à ce problème. En suivant la même approche, nos travaux se sont divisés en deux étapes. Tout d'abord nous avons défini un allocation optimale du budget d'annotations spécifiquement pour le problème de classification en fonction des paramètres de l'oracle dans chaque zone, ceci pour deux fonctions objectifs différentes. Un critère de sélection en connaissance parfaite menant à chacune de ces allocations a aussi été défini. Ensuite, nous avons utilisé le principe de l'optimisme face à l'incertitude pour développer deux algorithmes utilisant les critères définis précédemment qui permettent d'atteindre les différentes allocations optimales tout en nécessitant d'estimer les paramètres de l'oracle grâce aux annotations. Nous avons montré que pour tirer profit du fait que le problème de classification restreignait la possibilité de distribution de l'oracle à la famille de distributions de Bernoulli, les intervalles de crédibilité Bayésiens étaient un choix adéquat.

Nous avons aussi montré que le fait de se ramener à l'allocation optimale, qui ne dépend que des paramètres intrinsèques des zones, ne permet pas d'exploiter pleinement l'ensemble de l'information disponible, à savoir la prédiction courante du classifieur. L'objet du prochain chapitre est de nous libérer de cette restriction et de montrer que le principe de l'optimisme face à l'incertitude est toujours valable dans ce cas.

Chapitre 7

Échantillonnage adaptatif dans une partition fixe

7.1 Introduction

Dans le chapitre précédent, nous étudions le problème cherchant à ce que la répartition des données étiquetées résultant de l'algorithme tende vers l'allocation optimale, ce afin de reprendre l'angle abordé pour traiter le problème dans [13] et [14] et de pouvoir s'y comparer. Le principe d'une allocation est que la répartition du budget d'annotations puisse être définie initialement en fonction des paramètres du problème, et reste ainsi la même quelle que soit l'exécution du processus. Toutefois, cela ne permettait pas de tirer pleinement parti de l'ensemble de l'information rendue disponible par les observations reçues. Notamment, la stratégie de sélection idéale ne pouvait pas s'adapter à la prédiction courante du classifieur. Les observations étaient bien prises en compte dans le critère définitif mais ne servaient qu'à améliorer la connaissance des paramètres afin d'atteindre l'allocation idéale qui, elle, les ignorait. Dans ce chapitre, nous relâchons cette contrainte et autorisons le critère de sélection en connaissance parfaite à dépendre des observations reçues. Nous montrons comment cela se répercute sur l'algorithme final.

Permettre à la stratégie de sélection de constamment s'adapter à la prédiction courante du classifieur permet d'obtenir de meilleures performances. En effet, bien que l'allocation était qualifiée d'optimale, cela n'était vrai qu'en espérance dans le cas où elle ne pouvait pas changer d'une exécution à l'autre. Si le choix de la prochaine zone à échantillonner est autorisé à dépendre de la prédiction courante du classifieur, il est possible de profiter de la situation. Notamment, une prédiction que l'on saurait exacte dans une zone permettrait de redistribuer la partie du budget qui lui était destiné vers les autres zones, et ainsi d'en améliorer la qualité de la prédiction. Par exemple, si deux zones de même paramètre disposent d'un budget de 4 annotations, une allocation optimale voudrait que l'on attribue une part égale du budget à ces deux zones, c'est à dire 2. Dans le cas d'un échantillonnage adaptatif, si après le premier tirage effectué dans une zone l'étiquette reçue correspond à la prédiction optimale, d'une part il est préférable de ne plus échantillonner la zone au risque de modifier la prédiction, et d'autre part, cela permet de tirer 3 échantillons dans l'autre zone, ce qui améliore la qualité de la prédiction mais donne aussi plus de chances d'obtenir la prédiction optimale. Remarquez que la prédiction courante ne pouvant prendre que deux valeurs, une prédiction courante optimale intervient fréquemment, et possiblement tôt dans le processus.

L'idée est alors de permettre au critère de sélection en connaissance parfaite de prendre en compte l'estimation courante des paramètres tout en ayant accès à leurs vraies valeurs dans le but d'estimer le risque courant du classifieur. Ensuite, pour sélectionner au mieux la zone qu'aurait

choisi le critère de sélection en connaissance parfaite alors que la valeur des vrais paramètres ne sont plus accessibles, nous ferons appel au principe de l'optimisme face à l'incertitude. En effet, pour que le critère définitif parvienne à sélectionner les zones adéquates, il faut que l'estimation des paramètres soit d'assez bonne qualité. Il est donc quelques fois plus avantageux de favoriser les zones dont l'estimation manque de précision afin d'acquérir une certitude sur les décisions futures, ce qui se traduit par de l'exploration. Le principe de l'optimisme face à l'incertitude permet donc de gérer le compromis entre l'exploration et l'exploitation de l'estimation du critère idéal. Notez que la différence entre le problème précédent et celui-ci réside dans le fait qu'avant seul l'état final de l'algorithme importait sans tenir compte du chemin suivi, l'allocation réalisée qui devant être au plus proche de l'allocation optimale, tandis qu'ici le but est que le critère idéal correspondant à la zone sélectionnée soit minimal à chaque pas de temps. Nous changeons ainsi l'interprétation du problème qui correspondait auparavant à une minimisation du regret simple, et nous nous plaçons désormais dans le contexte d'une minimisation du regret cumulé. Ce contexte est d'autant plus adapté à l'apprentissage actif puisque ce dernier tente d'obtenir la meilleure vitesse d'apprentissage et que les meilleurs gains de performance interviennent le plus tôt possible. Enfin, notez que les résultats présentés dans ce chapitre ont fait l'objet d'une publication [22].

La démarche suivie reste tout de même similaire à la précédente en suivant le schéma qui a été défini au début de cette partie. Tout d'abord, deux critères de sélection en connaissance parfaite seront définis en Section 7.2, en fonction de l'objectif souhaité. Nous verrons que les fonctions de perte devront satisfaire plusieurs contraintes pour obtenir un comportement adéquat et nous montrerons comment les construire. Ensuite, dans la Section 7.3 nous utiliserons le principe de l'optimisme face à l'incertitude pour définir une stratégie de sélection pouvant gérer l'incertitude vis-à-vis des paramètres. Nous verrons que l'utilisation d'une approche Bayésienne permet de définir des intervalles de crédibilité sur un critère qui change à chaque pas de temps. De plus, nous donnerons une forme close pour ces critères ce qui permet un calcul plus rapide alors que pour les critères du chapitre précédent cela impliquait une inversion non standard, ce qui augmentait la complexité du calcul. En reprenant les expériences du chapitre précédent, nous comparerons dans la Section 7.5 ces nouveaux algorithmes aux précédents pour attester de l'amélioration apportée. Cela sera aussi l'occasion de les comparer à différents algorithmes de l'état de l'art chacun gérant l'incertitude d'une façon différente, et de montrer que nos algorithmes sont compétitifs et surpassent leurs performances.

7.2 Définition du critère d'échantillonnage en connaissance parfaite

Dans cette section, nous définissons les critères d'échantillonnage en connaissance parfaite. Ces derniers sont donc calculés en fonction de la vraie valeur des paramètres et établissent la ligne de conduite que les algorithmes finaux tenterons de suivre. Deux objectifs pourrons être envisagés, le premier est de minimiser le risque réel maximal sur les zones et le deuxième est de minimiser le risque réel global, c'est à dire la somme des risques réels sur les zones pondérés par leur importance relative. Nous verrons que satisfaire ces deux objectifs revient à utiliser deux approches différentes pour l'échantillonnage, la première correspond au scénario d'échantillonnage de l'incertitude et la deuxième correspond au scénario de réduction de l'erreur vus dans la partie précédente.

Comme dans la partie précédente, nous considérons ici une partition fixe avec des zones indépendantes. Nous utiliserons donc les mêmes notations, à savoir qu'une zone $k \in \{1, , \dot{K}\}$ de paramètre μ_k à reçu à l'instant t un nombre $T_{k,t}$ d'étiquettes de moyenne $\hat{\mu}_{k,t}$. Ainsi, le but est de définir un critère de sélection qui, à l'instant t permet de définir la prochaine zone à échantillonner, en fonction des paramètres et des observations dans chaque zone.

Rappelons que l'étiquette prédite par le classifieur au temps t f_t pour la zone k est

$$f_{k,t} = \lfloor \hat{\mu}_{k,t} \rceil,$$

où $\lfloor . \rceil$ est l'opérateur d'arrondi, que le risque réel associé à la zone k en utilisant la perte binaire $l_{0/1}$ est

$$R_k(f_t) = \mu_k \mathbb{1}_{f_{k,t}=0} + (1 - \mu_k)\mathbb{1}_{f_{k,t}=1},$$

et que le regret en termes de risque est

$$R_k(f_t) - R_k(f^*) = 2|\mu_k - \frac{1}{2}|\mathbb{1}_{f_{k,t} \neq \lfloor \mu_k \rceil},$$

avec f^* le classifieur qui attribue l'étiquette optimal à toutes les zones et $|.|$ l'opérateur de valeur absolue. Nous définissons donc la perte relative à une zone comme ce manque à gagner :

$$L^{dd}_{k,t}(\mu_k, T_{k,t}, \hat{\mu}_{k,t}) = 2|\mu_k - \frac{1}{2}|\mathbb{1}_{f_{k,t} \neq \lfloor \mu_k \rceil}.$$

Ici la mention dd indique que la fonction de perte est dépendante des données. Notez que pour une zone de paramètre fixé, cette perte ne peut prendre que deux valeurs en fonction de la valeur de la prédiction. Soit la prédiction est juste et la perte est égale à 0, soit la prédiction est erronée et la perte est égale à deux fois l'écart entre le paramètre et le seuil de prédiction $\frac{1}{2}$.

Les deux fonctions objectifs à minimiser que nous traitons dans cette partie sont : la perte maximale sur les zones

$$L^{dd,m}_t = \max_{k \in \{1,\dots,K\}} L^{dd}_{k,t}(\mu_k, T_{k,t}, \hat{\mu}_{k,t}),$$

et le risque global maximal, qui se traduit par

$$L^{dd,s}_t = \sum_{k=1}^{K} w_k L^{dd}_{k,t}(\mu_k, T_{k,t}, \hat{\mu}_{k,t}).$$

Puisque les zones sont indépendantes, le critère de sélection permettant de minimiser la première fonction objectif s'exprime comme

$$\begin{aligned} C^{dd,m}_{k,t}(\mu_k, \hat{\mu}_{k,t}, T_{k,t}) &= L^{dd}_{k,t}(\mu_k, T_{k,t}, \hat{\mu}_{k,t}) \\ &= 2|\mu_k - \frac{1}{2}|\mathbb{1}_{f_{k,t} \neq \lfloor \mu_k \rceil}. \end{aligned} \tag{7.1}$$

Et la stratégie en connaissance parfaite est alors :

$$k^{dd,m}_{t+1} = \operatorname*{arg\,max}_{k \in \{1,\dots,K\}} C^{dd,m}_{k,t}(\mu_k, \hat{\mu}_{k,t}, T_{k,t}).$$

Notez que ce critère n'est, à première vue, pas assuré de décroitre si l'on échantillonne une nouvelle fois dans une zone quelconque. En effet, cela peut, à la suite d'un mauvais tirage, avoir pour effet de modifier la valeur de la prédiction en mal. Cependant, lors de l'utilisation pratique de cette stratégie, si une zone possède une valeur de la perte nulle elle ne peut pas être sélectionnée. En effet, cela veut dire que la prédiction est correcte et que la perte est optimale, il ne sert donc à rien d'échantillonner une nouvelle fois dans la zone. En pratique cette règle n'est pas explicite, mais toutes les zones ayant une perte non-nulle sont sélectionnées en premier lieu, si une zone de perte nulle devait être sélectionnée, cela voudrait dire que toutes les autres zones ont aussi une perte nulle,

et que le classifieur optimal a été trouvé. Ainsi, si la zone sélectionnée possède une perte non-nulle, puisque seulement deux valeurs sont possibles, alors sa perte ne peut que devenir nulle ou rester la même. Le critère est donc obligatoirement décroissant lors de son utilisation et permet donc bien de minimiser la fonction objectif de façon optimale. Ainsi l'algorithme en connaissance parfaite procède de la façon suivante : il échantillonne les zones dont la prédiction est erronée à tour de rôle en commençant par celles dont le paramètre est le plus écarté de $\frac{1}{2}$, il continue à sélectionner la même zone jusqu'à ce que la prédiction associée soit correcte puis passe à la suivante.

Pour minimiser la deuxième fonction objectif, le critère en connaissance parfaite à utiliser est celui sélectionnant la zone pour laquelle la perte décroit le plus. La décroissance du risque réel s'exprimant comme

$$\begin{aligned}\Delta_y L^{dd}_{k,t}(\mu_k, T_{k,t}, \hat{\mu}_{k,t}) &= L^{dd}_{k,t}(\mu_k, T_{k,t}+1, \frac{T_{k,t}\hat{\mu}_{k,t}+y_{k,t+1}}{T_{k,t}+1}) - L^{dd}_{k,t}(\mu_k, T_{k,t}, \hat{\mu}_{k,t}) \\ &= 2|\mu_k - \frac{1}{2}|(\mathbb{1}_{\lfloor\hat{\mu}_{k,t}\rceil\neq\lfloor\mu_k\rceil} - \mathbb{1}_{\lfloor\frac{T_{k,t}\hat{\mu}_{k,t}+y_{k,t+1}}{T_{k,t}+1}\rceil\neq\lfloor\mu_k\rceil}) \\ &= \pm_{\lfloor\hat{\mu}_{k,t}\rceil=\lfloor\mu_k\rceil} 2|\mu_k - \frac{1}{2}|\mathbb{1}_{\lfloor\hat{\mu}_{k,t}\rceil\neq\lfloor\frac{T_{k,t}\hat{\mu}_{k,t}+y_{k,t+1}}{T_{k,t}+1}\rceil},\end{aligned}$$

et avec $y_{k,t+1}$ l'étiquette issue de l'échantillonnage de la zone k au temps $t+1$ et $\pm_{a=b}$ l'opérateur qui vaut 1 si $a=b$ et -1 autrement. Ceci peut s'interpréter simplement en voyant que la décroissance du risque est nulle si la prédiction ne change pas suite à l'échantillonnage, et est positive si la prédiction était initialement correcte et négative sinon. Seulement, l'étiquette qui sera reçue est pour l'instant inconnue, mais sa distribution de probabilité est connue puisque le paramètre de la zone est connu. Il convient donc de sélectionner la zone pour laquelle la décroissance du risque est la plus forte en espérance.

Cependant, une remarque peut être effectuée, il est possible que la prédiction d'une zone ne puisse pas être modifiée avec l'acquisition d'une seule nouvelle observation mais nécessite de l'échantillonner à plusieurs reprises. Dans ce cas, l'espérance de la décroissance du risque sera nulle puisque ce critère ne reflète que l'impact à court terme. L'échantillonnage d'une telle zone ne sera donc pas prioritaire. Pourtant, si elle possède un haut risque, il est possible que la décroissance finalement obtenue à la suite de ces multiples annotations soit meilleure que la somme des décroissances qui seraient obtenues sur les autres zones avec le même nombre d'annotations. Le problème est alors de calculer le gain à long terme possible sur les autres zones. Puisque les zones sont indépendantes, ceci peut être géré automatiquement par l'utilisation d'une fonction de perte convexe. En effet, la décroissance de la perte à l'instant présent est alors plus forte qu'une quelconque décroissance future. Ainsi, la stratégie sélectionnant à chaque pas de temps la décroissance maximale d'une perte convexe est assurée d'obtenir la perte minimum le plus rapidement possible. Nous tentons alors de définir une pseudo-décroissance permettant de représenter la décroissance du risque réel tout en étant strictement croissante avec le nombre d'observations.

Nous procéderons donc de la manière suivante, nous commencerons par étudier la fonction de décroissance du risque réel pour un nombre d'échantillons quelconque, puis nous la bornerons par une fonction strictement croissante, pour finalement ne garder que sa valeur initiale comme critère de sélection. Soit T^+ le nombre de nouveaux échantillons que l'on prend dans la zone k et $\hat{\mu}^+$ leur moyenne. La décroissance du risque ainsi obtenue est

$$\frac{1}{T^+}\Delta_{T^+,\hat{\mu}^+} L^{dd}_{k,t}(\mu_k, T_{k,t}, \hat{\mu}_{k,t}) = \pm_{\lfloor\hat{\mu}_{k,t}\rceil=\lfloor\mu_k\rceil} \frac{2|\mu_k - \frac{1}{2}|}{T^+}\mathbb{1}_{\lfloor\hat{\mu}_{k,t}\rceil\neq\lfloor\frac{T_{k,t}\hat{\mu}_{k,t}+T^+\hat{\mu}^+}{T_{k,t}+T^+}\rceil}.$$

Et donc

$$\mathbb{E}_{T^+\hat{\mu}^+\sim\mathcal{B}in(T^+,\mu_k)}[\frac{1}{T^+}\Delta_{T^+,\hat{\mu}^+}L^{dd}_{k,t}(\mu_k,T_{k,t},\hat{\mu}_{k,t})]$$
$$= \pm_{\lfloor\hat{\mu}_{k,t}\rceil=\lfloor\mu_k\rceil}\frac{2|\mu_k-\frac{1}{2}|}{T^+}\mathbb{P}_{T^+\hat{\mu}^+\sim\mathcal{B}in(T^+,\mu_k)}(\lfloor\hat{\mu}_{k,t}\rceil\neq\lfloor\frac{T_{k,t}\hat{\mu}_{k,t}+T^+\hat{\mu}^+}{T_{k,t}+T^+}\rceil).$$

Commençons par étudier le cas où $\lfloor\hat{\mu}_{k,t}\rceil=0$ et $\lfloor\mu_k\rceil=1$. Ainsi,

$$\mathbb{P}_{T^+\hat{\mu}^+\sim\mathcal{B}in(T^+,\mu_k)}(\lfloor\hat{\mu}_{k,t}\rceil\neq\lfloor\frac{T_{k,t}\hat{\mu}_{k,t}+T^+\hat{\mu}^+}{T_{k,t}+T^+}\rceil)$$
$$=\mathbb{P}_{T^+\hat{\mu}^+\sim\mathcal{B}in(T^+,\mu_k)}(T_{k,t}\hat{\mu}_{k,t}+T^+\hat{\mu}^+\geq\frac{1}{2}(T_{k,t}+T^+))$$
$$=\mathbb{P}_{T^+\hat{\mu}^+\sim\mathcal{B}in(T^+,\mu_k)}(T^+\hat{\mu}^+\geq\frac{1}{2}T^++T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})).$$

Remarquez que si $T^+<2T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})$ alors la probabilité est nulle. En effet, dans ce cas, même en n'obtenant que des étiquettes de valeur 1 la prédiction ne pourrait pas être modifiée. Afin que le critère inclue une information sur le long terme, il est nécessaire d'utiliser une valeur de T^+ supérieure à ce seuil. Étant donné que $T^+\hat{\mu}^+$ est tirée selon une loi Binomiale de paramètres T^+ et μ_k, alors

$$\mathbb{P}_{T^+\hat{\mu}^+\sim\mathcal{B}in(T^+,\mu_k)}(T^+\hat{\mu}^+\geq\frac{1}{2}T^++T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t}))$$
$$=(1-\mathcal{I}_{1-\mu_k}(T^+-\lceil\frac{1}{2}T^++T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})-1\rceil,\lceil\frac{1}{2}T^++T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})-1\rceil+1)).$$

avec $\mathcal{I}_{1-p}(n-\lfloor.\rfloor,\lfloor.\rfloor+1)$ la fonction de répartition de la loi Binomiale de paramètre n,p et $\lceil.\rceil$ l'opérateur de partie entière par excès. Cette fonction étant en escalier à cause de la partie entière, nous la bornons en prenant la valeur maximale de la partie entière afin d'obtenir une fonction strictement décroissante avec T^+. Notez que $\forall T^+\in\mathbb{N}^+$, $\frac{1}{2}T^++T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})-1\leq\lceil\frac{1}{2}T^++T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})-1\rceil\leq\frac{1}{2}T^++T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})-\frac{1}{2}$. Ainsi,

$$\mathbb{P}_{T^+\hat{\mu}^+\sim\mathcal{B}in(T^+,\mu_k)}(T^+\hat{\mu}^+\geq\frac{1}{2}T^++T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t}))$$
$$\leq 1-\mathcal{I}_{1-\mu_k}(\frac{T^+}{2}-T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})+\frac{1}{2},\frac{T^+}{2}+T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})+\frac{1}{2}).$$

Cette borne est très étroite et atteint la vraie probabilité en un nombre infiniment dénombrable de points. En remplaçant T^+ par la plus petite valeur pour laquelle la prédiction peut être modifiée, c'est à dire $T^+=2T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})$, alors

$$\mathbb{P}_{T^+\hat{\mu}^+\sim\mathcal{B}in(T^+,\mu_k)}(T^+\hat{\mu}^+\geq\frac{1}{2}T^++T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t}))$$
$$\leq 1-\mathcal{I}_{1-\mu_k}(\frac{1}{2},2T_{k,t}(\frac{1}{2}-\hat{\mu}_{k,t})+\frac{1}{2}).$$

En suivant le même raisonnement dans le cas où $\lfloor\hat{\mu}_{k,t}\rceil=1$ et $\lfloor\mu_k\rceil=0$, avec $T^+=2T_{k,t}(\hat{\mu}_{k,t}-\frac{1}{2})+1$, nous obtenons :

$$\mathbb{P}_{T^+\hat{\mu}^+\sim\mathcal{B}in(T^+,\mu_k)}(\lfloor\hat{\mu}_{k,t}\rceil\neq\lfloor\frac{T_{k,t}\hat{\mu}_{k,t}+T^+\hat{\mu}^+}{T_{k,t}+T^+}\rceil)$$
$$\leq\mathcal{I}_{1-\mu_k}(2T_{k,t}(\hat{\mu}_{k,t}-\frac{1}{2})+\frac{3}{2},\frac{1}{2}).$$

Si $\lfloor \hat{\mu}_{k,t} \rceil = \lfloor \mu_k \rceil$ alors

$$\mathbb{E}_{T^+\hat{\mu}^+ \sim \mathcal{B}in(T^+,\mu_k)}[\frac{1}{T^+}\Delta_{T^+,\hat{\mu}^+} L^{dd}_{k,t}(\mu_k, T_{k,t}, \hat{\mu}_{k,t})] \geq 0.$$

Ainsi, on peut simplement remplacer $\pm_{\lfloor \hat{\mu}_{k,t} \rceil = \lfloor \mu_k \rceil}$ par $-\mathbb{1}_{\lfloor \hat{\mu}_{k,t} \rceil \neq \lfloor \mu_k \rceil}$.

Finalement, en combinant les différents cas, nous obtenons une fonction strictement décroissante de T^+ qui borne la décroissance du risque réel encourue par l'échantillonnage d'une nouvelle observation dans une zone. Ainsi, en prenant $T^+ = 2T_{k,t}|\hat{\mu}_{k,t} - \frac{1}{2}| + \mathbb{1}_{\lfloor \hat{\mu}_{k,t} \rceil = 1}$,

$$\begin{aligned}
&\mathbb{E}_{T^+\hat{\mu}^+ \sim \mathcal{B}in(T^+,\mu_k)}[\frac{1}{T^+}\Delta_{T^+,\hat{\mu}^+} L^{dd}_{k,t}(\mu_k, T_{k,t}, \hat{\mu}_{k,t})] \qquad (7.2)\\
&\geq \frac{2|\mu_k - \frac{1}{2}|}{2T_{k,t}|\hat{\mu}_{k,t} - \frac{1}{2}| + \mathbb{1}_{\lfloor \hat{\mu}_{k,t} \rceil = 1}}(\quad \mathbb{1}_{\lfloor \mu_k \rceil = 1}\mathbb{1}_{\lfloor \hat{\mu}_{k,t} \rceil = 0}(1 - \mathcal{I}_{1-\mu_k}(\frac{1}{2}, 2T_{k,t}(\frac{1}{2} - \hat{\mu}_{k,t}) + \frac{1}{2}))\\
&\qquad\qquad + \quad \mathbb{1}_{\lfloor \mu_k \rceil = 0}\mathbb{1}_{\lfloor \hat{\mu}_{k,t} \rceil = 1}\mathcal{I}_{1-\mu_k}(2T_{k,t}(\hat{\mu}_{k,t} - \frac{1}{2}) + \frac{3}{2}, \frac{1}{2}) \quad)\\
&= -C^{dd,s}_{k,t}(\mu_k, \hat{\mu}_{k,t}, T_{k,t}).
\end{aligned}$$

Ainsi, la stratégie en connaissance parfaite définie par

$$k^{dd,s}_{t+1} = \underset{k \in \{1,\dots,K\}}{\arg\max}\ C^{dd,s}_{k,t}(\mu_k, \hat{\mu}_{k,t}, T_{k,t})$$

permet de minimiser la seconde fonction objectif.

7.3 Prise en compte de l'incertitude

Dans la section précédente, nous venons de mettre en place les critères définissant les zones que l'on sélectionnerait, dans le but de réduire le risque réel du classifieur le plus rapidement possible, si les paramètres du problème, notamment ceux de la distribution de l'oracle, étaient connus. En pratique, ces valeurs ne sont pas accessibles directement mais doivent être apprises à travers les observations résultant des annotations effectuées au cours de l'exécution de l'algorithme. Nous allons donc maintenant procéder à la définition du critère de sélection définitif qui emploi le principe de l'optimisme face à l'incertitude pour tenter de se rapprocher du critère idéal en n'ayant accès qu'à ces observations.

La démarche employée est ici basiquement la même que celle du chapitre précédent. Des intervalles de crédibilité sont définis pour contenir la valeur du critère idéal avec une probabilité donnée. Pour cela, une distribution *a posteriori* doit être calculée sur la valeur des paramètres de l'oracle qui sera ensuite transférée au critère idéal. A partir de celle-ci, il est possible de déterminer les valeurs du critère idéal constituant les bornes de l'intervalle de crédibilité. Le critère utilisé sera finalement la borne supérieure de cet intervalle de crédibilité. Notez qu'il est encore une fois possible de profiter de la connaissance de la famille de distributions et de l'indépendance des zones, pour utiliser une approche Bayésienne sans nécessiter de faire des choix arbitraires. Finalement, la seule différence avec le chapitre précédent réside dans la nature des critères idéaux utilisés. Cependant, ces derniers possèdent ici la propriété intéressante d'être des fonctions strictement monotones du paramètre de l'oracle pour la zone considérée. Par conséquent, ceci rend beaucoup plus simple le calcul de l'intervalle de crédibilité associé. En effet, ceci ne nécessite pas d'inversion calculatoire et fait uniquement appel à des fonctions connues.

Les deux algorithmes résultants partagent le même corps, qui est représenté sur la Figure 4. Ceux-ci peuvent être dérivés en précisant le critère de sélection en connaissance parfaite utilisé. Ils prennent en argument un paramètre δ qui permet de régler la probabilité relative aux intervalles de crédibilité. Plus cette probabilité est grande, plus l'étendue des valeurs crédibles du paramètre est large. Elle permet donc en outre de contrôler le ratio entre exploration et exploitation. Dans la suite, nous détaillons le fonctionnement de l'algorithme et comment calculer le critère de sélection final.

Données : N, $(w_k)_{k\in\{1,\dots,K\}}$, δ
Initialisation : $\forall k \in \{1,\dots,K\}$, $\hat{\mu}_{k,0} = \frac{1}{2}$, $T_{k,0} = 0$
pour *t = 1, ..., n* **faire**
  Calculer $e^{sup}_{k,t}(\delta)$ à partir de (7.3) avec $C_{k,t}$ le critère issu de (7.1) ou (7.2)
  Échantillonner la zone $k_t = \arg\max_k e^{sup}_{k,t}(\delta)$
  Recevoir $y_t \sim \mathcal{B}er(\mu_{k_t})$
  Mettre à jour $\hat{\mu}_{k_t,t}$ et $T_{k_t,t}$
fin
Résultat : $\lfloor \hat{\mu}_{k,n} \rceil$ pour chaque zone $k \in \{1,\dots,K\}$

Algorithme 4 : Algorithme générique

Le critère de sélection que nous souhaitons définir doit correspondre à la borne supérieure d'un intervalle de crédibilité. Voyons tout d'abord comment construire un intervalle de crédibilité Bayésien sur le critère idéal. Soit une zone k de paramètre μ_k ayant été échantillonnée $T_{k,t}$ fois à l'instant t, la moyenne des étiquettes reçues étant $\hat{\mu}_{k,t}$. En utilisant une approche Bayésienne, nous pouvons calculer une distribution *a posteriori* sur la valeur du paramètre μ_k. Nous savons que les observations sont tirées d'une loi de Bernoulli et donc que leur moyenne suit une loi Binomiale. Nous savons aussi que la famille de lois Beta forme une famille de loi conjuguées pour la loi Binomiale. Ainsi, la distribution *a posteriori* sur μ_k est

$$\mu_k \sim Beta(T_{k,t}\hat{\mu}_{k,t} + 1, T_{k,t}(1 - \hat{\mu}_{k,t}) + 1).$$

Soit $C^{dd}_{k,t}$ un des deux critères en connaissance parfaite définis dans la partie précédente. Pour $\hat{\mu}_{k,t} \geq \frac{1}{2}$ et $T_{k,t}$ fixés, $C^{dd}_{k,t}$ est une fonction décroissante de μ_k. Ainsi, $\forall \epsilon_{k,t}, \forall \delta \in [0,1]$

$$\begin{aligned}&\mathbb{P}_{\mu_k\sim Beta(T_{k,t}\hat{\mu}_{k,t}+1,T_{k,t}(1-\hat{\mu}_{k,t})+1)}(\mu_k \leq \epsilon_{k,t}) = \delta \\ &\implies \mathbb{P}_{\mu_k\sim Beta(T_{k,t}\hat{\mu}_{k,t}+1,T_{k,t}(1-\hat{\mu}_{k,t})+1)}(C^{dd}_{k,t}(\mu_k,\hat{\mu}_{k,t},T_{k,t}) \geq C^{dd}_{k,t}(\epsilon_{k,t},\hat{\mu}_{k,t},T_{k,t})) = \delta.\end{aligned}$$

Soit $F(.,\alpha,\beta)$ la fonction de répartition de la loi Beta de paramètres α et β. Alors,

$$\begin{aligned}&\mathbb{P}_{\mu_k\sim Beta(T_{k,t}\hat{\mu}_{k,t}+1,T_{k,t}(1-\hat{\mu}_{k,t})+1)}(C^{dd}_{k,t}(\mu_k,\hat{\mu}_{k,t},T_{k,t}) \\ &\geq C^{dd}_{k,t}(F^{-1}(\delta,T_{k,t}\hat{\mu}_{k,t}+1,T_{k,t}(1-\hat{\mu}_{k,t})+1),\hat{\mu}_{k,t},T_{k,t})) = \delta.\end{aligned}$$

De même, si $\hat{\mu}_{k,t} < \frac{1}{2}$, $C^{dd}_{k,t}$ est une fonction croissante de μ_k.

$$\begin{aligned}&\mathbb{P}_{\mu_k\sim Beta(T_{k,t}\hat{\mu}_{k,t}+1,T_{k,t}(1-\hat{\mu}_{k,t})+1)}(C^{dd}_{k,t}(\mu_k,\hat{\mu}_{k,t},T_{k,t}) \\ &\geq C^{dd}_{k,t}(F^{-1}(1-\delta,T_{k,t}\hat{\mu}_{k,t}+1,T_{k,t}(1-\hat{\mu}_{k,t})+1),\hat{\mu}_{k,t},T_{k,t})) = \delta.\end{aligned}$$

Ainsi, soit

$$\epsilon^{sup}_{k,t} = C^{dd}_{k,t}(F^{-1}(\mathbb{1}_{\lfloor\hat{\mu}_{k,t}\rceil=0}(1-\delta) + \mathbb{1}_{\lfloor\hat{\mu}_{k,t}\rceil=1}\delta, T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1), \hat{\mu}_{k,t}, T_{k,t}), \quad (7.3)$$

alors

$$\mathbb{P}_{\mu_k \sim Beta(T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1)}(C^{dd}_{k,t}(\mu_k, \hat{\mu}_{k,t}, T_{k,t}) \geq \epsilon^{sup}_{k,t}) = \delta.$$

Et donc $\epsilon^{sup}_{k,t}$ agit comme la borne supérieur d'un intervalle de crédibilité Bayésien contenant le critère idéal avec une probabilité $1 - 2\delta$. Un algorithme de sélection utilisant le principe de l'optimisme face à l'incertitude échantillonnera donc la zone

$$k_{t+1} = \underset{k \in \{1,\dots,K\}}{\arg\max}\ \epsilon^{sup}_{k,t}.$$

7.4 Expériences

Dans les sections précédentes nous avons introduits deux algorithmes utilisant le principe de l'optimisme face à l'incertitude dans le but de résoudre le problème d'apprentissage actif sur une partition fixe avec des zones indépendantes. Nous nous proposons maintenant d'évaluer ces algorithmes et de les comparer avec d'autres algorithmes issus de l'état de l'art afin de juger de leur efficacité. Les expériences menées pour les évaluer sont les mêmes que celles du chapitre précédent. Nous rappelons brièvement ici leur description et les détails de leur implémentations. Nous présenterons aussi les algorithmes utilisés à titre de comparaison avec les nôtres.

Tout d'abord, nous avions montré que n'importe quel problème, en incluant ceux provenant du monde réel, pouvaient être caractérisés par un nombre réduit de paramètres, à savoir :

- le nombre de zones K,
- les paramètres de la distribution de Bernoulli associée à chaque zone $(\mu_k)_{k \in \{1,\dots,K\}}$,
- les importances relatives de chaque zone $(w_k)_{k \in \{1,\dots,K\}}$.

Ceci étant dû au fait que la partition soit fixe et que la diversité à l'intérieur de chaque zone ne soit jamais accessible.

Nous définissons donc deux bancs de tests, l'un permettant d'évaluer la performance moyenne sur une grande diversité de problèmes tandis que l'autre permet d'étudier un problème spécifique aux caractéristiques bien choisies. Dans le premier banc de tests, 1000 problèmes sont générés aléatoirement de la façon suivante :

- K tiré uniformément dans $[\![1, 50]\!]$,
- $\forall k \in \{1, \dots, K\}, \quad \mu_k$ est tirée uniformément dans $[0, 1]$,
- $\forall k \in \{1, \dots, K\}, \quad w_k$ est tirée uniformément dans $[0, 1]$.

Puis chaque algorithme est exécuté 10 fois sur chaque problème du banc de test avec un budget de 1000 annotations. Le risque réel est calculé de façon exact grâce à la connaissance des vrais paramètres et est enregistré à chaque pas de temps. Ainsi, le risque réel au temps t est

$$R_t = \sum_{k=1}^{K} w_k(\mu_k \mathbb{1}_{\lfloor \hat{\mu}_{k,t} \rceil = 0} + (1 - \mu_k)\mathbb{1}_{\lfloor \hat{\mu}_{k,t} \rceil = 1}).$$

La performance globale d'un algorithme est calculée en effectuant à chaque pas de temps la moyenne du risque réel obtenu sur l'ensemble des problèmes et des exécutions.

Dans le second banc de tests, un seul problème est étudié, il possède les caractéristiques suivantes :

- $K = 10$,
- $\forall k \in \{1, \ldots, K\}, \quad \mu_k = \frac{k}{K}$,
- $\forall k \in \{1, \ldots, K\}, \quad w_k = \frac{1}{K}$.

Notez que les importances relatives de chaque zone sont les mêmes et que les paramètres sont espacés le plus possible. Chaque algorithme est exécuté 10000 fois sur ce problème avec un budget de 300 annotations. La performance globale est calculée de la même manière que précédemment.

Les expériences ont pour but de montrer deux choses. D'une part nous tentons de savoir si la prise en compte des tirages effectifs de la distribution de l'oracle dans le critère en connaissance parfaite engendre une amélioration des algorithmes du chapitre précédent. En outre, il faut savoir si l'utilisation de l'optimisme face à l'incertitude permet d'utiliser efficacement ce critère idéal. D'autre part, nous comparons nos algorithmes à ceux de l'état de l'art afin de savoir si ils sont compétitifs et si le principe de l'optimisme face à l'incertitude fournit une solution efficace pour gérer l'incertitude vis-à-vis de la distribution de l'oracle. Nous devons donc comparer nos algorithmes à d'autres différant dans la manière de gérer cette incertitude. Tous les algorithmes ont été adaptés au cas d'une partition fixe. Voici une description des algorithmes utilisés pour la comparaison :

- **Échantillonnage aléatoire** : la zone sélectionnée est tirée avec une probabilité proportionnelle à son importance relative, cela revient à tirer une donnée d'entrée selon la distribution naturelle,
- **OALC-FP**^{m} et **OALC-FP**^{s} : algorithmes du chapitre précédent, utilisant le principe de l'optimisme face à l'incertitude mais ne s'intéressant uniquement qu'à l'allocation optimale, donc sans prise en compte des tirages effectifs, utilise respectivement la fonction objectif basée sur le risque local maximal et la somme pondérée des risques locaux,
- **Échantillonnage de l'incertitude** [55] : algorithme standard d'apprentissage actif, basé sur la minimisation du risque local, remplaçant directement les paramètres de l'oracle par leur estimation,
- **Réduction de l'erreur** [65] : algorithme basé sur l'espérance de la réduction du risque global en simulant chaque étiquette possible, remplaçant directement les paramètres de l'oracle par leur estimation,
- **Espérance de l'erreur *a posteriori*** : algorithme adapté de [49], dans une approche Bayésienne, minimisation de l'espérance de l'erreur de prédiction *a posteriori*, l'adaptation diffère dans la nature de la distribution *a priori* considérée, ici c'est une loi Beta,
- **Échantillonnage de Thompson** [73] : dans une approche Bayésienne, à chaque pas de temps des paramètres sont tirées de leur distribution *a posteriori* et son utilisé directement dans le critère idéal,
- **EffECXtive** [34] : algorithme d'apprentissage actif utilisant la notion de classe d'équivalence et définissant des poids entre celles-ci, il tente ensuite de déterminer le plus rapidement la classe d'équivalence contenant le classifieur idéal.

Pour tous les algorithmes optimistes utilisés, la valeur du paramètre δ permettant de contrôler la quantité d'exploration a été réglée en effectuant une recherche par quadrillage.

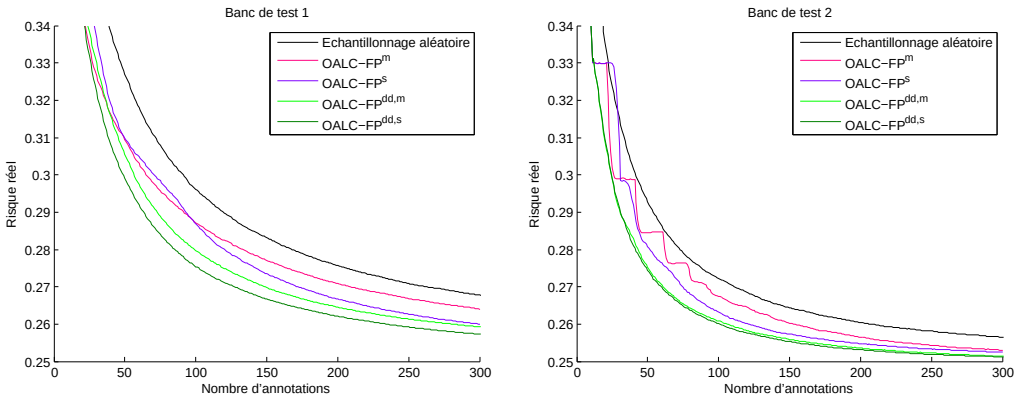


FIGURE 7.1 – Comparaison des algorithmes dépendant des tirages effectifs avec leur équivalent non-dépendant

7.5 Résultats

Dans cette section, nous présentons et commentons les résultats des expériences décrites dans la section précédente. Nous commençons par comparer les performances des algorithmes introduits dans ce chapitre avec ceux du chapitre précédent. Rappelons que la différence se situe dans le fait que la premier tente d'atteindre l'allocation optimale tandis que le second tente à chaque pas de temps de sélectionner la zone qui améliore le plus les performances (locales ou globales) du classifieur. Nous souhaitons ainsi évaluer ce que cela apporte en termes de performances. Les résultats sont affichés sur la Figure 7.1, pour le premier banc de tests, l'affichage se concentre sur les 300 premiers pas de temps afin de distinguer les courbes. Nous pouvons voir que les performances des nouveaux algorithmes sont toujours significativement meilleures que celles de ceux du chapitre précédent. Ceci confirme donc que la prise en compte des tirages effectifs présente un intérêt. De plus, nous pouvons voir que dans le cas du second banc de tests, les deux algorithmes dépendants des données donnent des résultats semblables. En effet, ceci est dû au fait que dans ce banc de tests, les importances relatives de chaque zones sont toutes égales. Notez qu'ici, la version minimisant le risque local maximal sur les zones pondère cependant le critère local par l'importance relative afin de privilégier les zones ayant plus d'influence sur le risque, à la manière de la pondération selon la densité vu dans l'état de l'art. Pourtant, cette solution permet mal d'estimer l'apport de l'échantillonnage d'une zone dans la décroissance du risque global. La décroissance du risque local est un fonction croissante du risque actuellement encouru. Ainsi, si les importances de chaque zone sont égales, sélectionner la zone pour laquelle le risque local est maximal est strictement équivalent à sélectionner celle dont la décroissance est maximale. Cependant, la décroissance du risque n'est pas linéaire avec le risque local. Ainsi, si les importances relatives ne sont pas égales, cela modifie la zone à privilégier.

Jusqu'ici, nous avons construit progressivement un algorithme d'apprentissage actif utilisant le principe de l'optimisme face à l'incertitude. En partant d'un algorithme conçu pour de la régression, il a été montré qu'un critère idéal spécifique à la classification pouvait être utilisé pour améliorer les performances, et que relâcher l'aspect déterministe du critère idéal rendait l'algorithme encore plus performant. Les comparaisons avaient principalement pour but de montrer que les différentes étapes de la création étaient justifiées. Finalement, nous obtenons un algorithme performant utilisant le principe de l'optimisme face à l'incertitude pour l'apprentissage actif en classification. Il reste à savoir comment il se compare avec d'autres méthodes plus classiques d'apprentissage

actif. La Figure 7.2 présente les résultats des expériences pour ces méthodes. Deux méthodes se distinguent par leur inefficacité, il s'agit de l'échantillonnage de l'incertitude et de la réduction de l'erreur. Le point commun entre ces deux méthodes est que le paramètre de la distribution de l'oracle est directement remplacé par son estimation dans le critère idéal. La première utilise un critère idéal tentant de minimiser le risque local tandis que celui de la deuxième tente de minimiser l'erreur globale. Dans les deux cas, ce qui fait défaut est le manque d'exploration. Ceci ressort particulièrement durant l'utilisation d'une partition fixe avec des zones indépendantes. Comme nous l'avons déjà décrit, dans ce cas, il existe un instant à partir duquel une seule zone est échantillonnée, c'est la raison pour laquelle les performances stagnent.

Examinons maintenant la méthode utilisant l'espérance de l'erreur de prédiction *a posteriori* lors d'une approche Bayésienne. Cette méthode est supposée inclure une part d'exploration puisque la variance de la distribution *a posteriori* du paramètre est plus grande lorsque le nombre d'observations est faible. Cependant, nous observons ici de faibles performances pour cette méthode, ce qui montre qu'elle n'est pas efficace pour gérer l'incertitude vis-à-vis des paramètres dans le critère idéal. Parmi les méthodes les plus performantes se trouve EffECXtive. Nous voyons que les performances initiales d'OALC-FP^{dd,s} sont toutefois meilleures que cette dernière. Durant les pas de temps suivants, la différence entre les performances d'OALC-FP^{dd,s} et d'EffECXtive se rétréci. Notez qu'OALC-FP^{dd,m} est meilleure au début mais se fait dépasser à la fin, ceci s'explique par le fait qu'EffECXtive soit une méthode minimisant le risque global avec un critère de sélection regardant un pas de temps en avant.

7.6 Conclusion intermédiaire

Dans ce chapitre, nous avons montré que le principe de l'optimisme face à l'incertitude pouvait être utilisé pour suivre un critère, défini de manière idéal en utilisant la connaissance des paramètres de la distribution de l'oracle, servant à minimiser le risque de classification avec le moins d'annotations possible. Pour cela nous nous sommes basé sur les stratégies introduites dans le chapitre précédent, qui utilisaient le principe de l'optimisme face à l'incertitude pour tendre, à terme, vers une allocation prédéfinie. A la différence de celle-ci, ici l'allocation du budget d'annotations recherchée par le critère idéal change à chaque instant en fonction des observations nouvellement reçues, et le but est de suivre à chaque instant ce critère idéal tout en étant entaché d'une incerti-

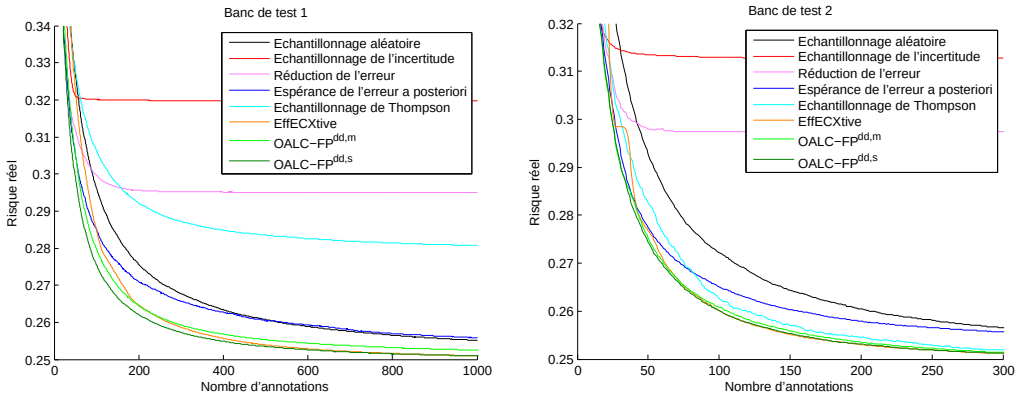


FIGURE 7.2 – Comparaison de nos algorithmes avec des algorithmes standards d'apprentissage actif

tude. Ceci peut être interprété en remarquant que dans le cas où le critère idéal est la décroissance du risque réellement encourue à la suite d'une annotation, alors l'objectif à maximiser pour atteindre le risque minimum est la somme de ces décroissances ; ainsi, ceci revient à maximiser le cumul des récompenses dont la valeur est incertaine mais peut être affinée à chaque pas de temps. Ceci nous ramène donc au cas usuel des bandits stochastiques pour lequel le principe de l'optimisme face à l'incertitude à été initialement défini. Nous avons montré à travers les expériences que cette approche donnait des résultat compétitifs face aux méthodes standards d'apprentissage actif.

Le fait que le critère, et donc la récompense attendue, change à chaque pas de temps ne permet pas de l'estimer de façon fréquentiste. Celui-ci dépend toutefois directement des paramètres de l'oracle, qui eux, peuvent être appris. Cependant, la non-linéarité du critère vis-à-vis de ces derniers n'aurait pas permit de les utiliser de façon brute dans son estimation sans introduire un biais, et donc sans obtenir une concentration erronée. L'utilisation d'une approche Bayésienne possède donc non seulement l'avantage que les intervalles de crédibilité obtenus soient très étroits, mais rend tout simplement possible le calcul de ces dernières.

Nous nous sommes placés dans le problème des partitions fixes avec des zones indépendantes. De cette manière, les comportement des algorithmes étaient exagérés, notamment ceux correspondant à la gestion du dilemme entre exploration et exploitation. Nous avons, par exemple, pu constater que les méthodes se passant de le considérer voyaient leurs performances chuter de façon drastique. Ceci nous a donc permis de développer nos algorithmes dans un environnement contrôlé. Cependant, ce problème est peu utilisable en pratique car les performances dépendent grandement du choix initial de la partition. En effet, plus les zones contiennent des étiquettes variées, plus grande sera l'erreur due à l'attribution d'une étiquette unique. Étant donné que cette partition n'est pas autorisée à changer durant l'exécution de l'algorithme et que, initialement, aucune étiquette n'a été observée, alors le choix de la partition initiale est déterminant, et pourtant celui-ci n'est pas trivial. Dans la suite, nous envisageons de relâcher certaines contraintes, par conséquent, ce choix ne devra plus être problématique. Deux pistes seront envisagées : la partition adaptative qui re-découpe les zones au cours de l'exécution, et la partition fixe avec des zones combinées, dans laquelle la valeur du paramètre peut être estimée à partir des observations dans les autres zones.

Chapitre 8

Utilisation d'un partition adaptative

8.1 Introduction

Jusqu'ici, nous nous étions limité à l'étude d'une partition fixe, c'est à dire qui ne pouvait en aucun cas être modifiée au cours de l'exécution de l'algorithme. Les performances de classification étaient alors limitées par la définition initiale de la partition. En effet, deux cas pouvaient alors se produire. Dans le cas où le nombre de zones était faible, la taille de chacun d'entre elles est nécessairement grande. Il y a donc plus de chance qu'elle intègrent des étiquettes de différentes classes. Cependant, l'hétérogénéité des étiquettes associée à la possibilité de ne prédire qu'une seule étiquette dans chaque zone entraine une erreur de classification inévitable. Cela limite donc la performance maximale atteignable. Au contraire, dans le cas où le nombre de zones était très grand, une discrétisation plus fine de l'espace d'entrée augmente la probabilité que les limites entre les zones coïncident avec les frontières entre classes. Cependant, le nombre de zones limite la vitesse d'apprentissage. En effet, puisque les prédictions dans chaque zones doivent être apprises indépendamment, plus le nombre de zones est grand, plus le nombre d'observations nécessaires pour arriver à une même précision sur la prédiction est grand. Notez qu'une partition adaptée aux données était difficile à obtenir car aucune étiquette n'est encore été reçue au moment de la définition de la partition. Notez aussi que le nombre de zones de la partition pouvait dépendre du budget d'annotations, un faible budget impliquant d'obtenir rapidement une bonne prédiction et donc de limiter le nombre de zones tandis qu'un fort budget permettant une précision convenable sur plusieurs zones simultanément. Dans ce chapitre, nous nous intéressons au cas d'une partition adaptative, qui est capable d'évoluer au cours de l'exécution. Le but va donc être de savoir comment guider la découpe de cette partition pour qu'elle s'adapte à la répartition des classes sur l'espace d'entrée. Nous allons également voir comment faire évoluer le nombre de zones afin de gérer le compromis entre la vitesse d'apprentissage et la performance maximale atteignable et ainsi tirer parti du meilleur des deux mondes.

Le but sera ensuite de définir une stratégie d'apprentissage actif agissant sur cette partition adaptative. Seulement, pour que celle-ci fonctionne, il est nécessaire de régir la façon dont la partition peut évoluer. En effet, une hypothèse importante est que, dans chaque zone, les données d'entrée étiquetées soient tirées selon leur distribution naturelle. Puisque la stratégie d'apprentissage actif a pour but de privilégier certaines zones par rapport à d'autres, en modifiant la partition la distribution dans chaque zone ainsi obtenue pourrait être bouleversée. Ainsi, il est nécessaire d'interdire aux frontières entre les zones d'être déplacées. De même, une frontière existante ne doit pas être supprimée dans le but de fusionner deux zones. Par contre, une zone déjà existante peut être re-découpée afin d'affiner la discrétisation et de donner la possibilité d'attribuer des étiquettes

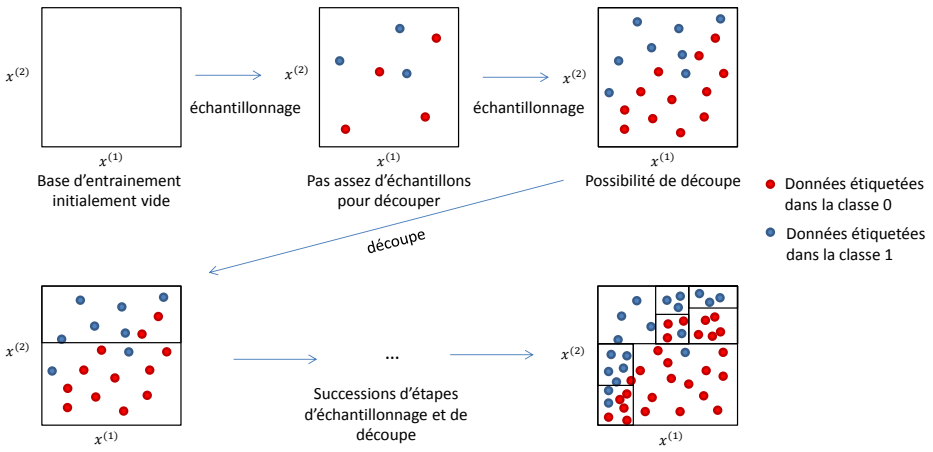


FIGURE 8.1 – Illustration d'une partition adaptative avec un espace d'entrée en 2D

différentes à chaque zone ainsi crée. Ainsi, l'idée est de commencer avec un espace non-partitionné, dont l'unique zone contient tout l'espace d'entrée, puis de re-diviser comme on le souhaite chaque zone au fur et à mesure de l'avancement du processus. La Figure 8.1 permet de visualiser ce que donne une partition adaptative sur un problème de classification binaire en deux dimensions. A chaque pas de temps, l'algorithme d'apprentissage actif choisit la zone dans laquelle échantillonner une donnée d'entrée tirée selon sa distribution naturelle conditionnée à cette zone. Puis, si suffisamment d'étiquettes ont été reçues et que la répartition des classes dans la zone le permet, la zone est divisée en deux zones plus homogènes, et ce de façon récursive. Notez que les données d'entrée à l'intérieur de chaque zone sont toujours tirées selon leur distribution naturelle. En observant la Figure 8.1, nous pouvons voir que certaines zones ont été plus sollicitées que d'autres si bien que les données étiquetées prises dans leur ensemble ne suivent plus leur distribution naturelle. Dans l'approche que nous venons de décrire, les étapes de construction de la partition et de choix de la zone à échantillonner sont complètement indépendantes. En effet, la décision de diviser une zone ou non ne tient pas compte des zones voisines ni des étapes passées ou futures. Ainsi, nous décidons de traiter les deux aspects séparément. D'une part, nous développons un algorithme de construction de la partition adaptative qui fonctionne avec n'importe quelle stratégie de sélection. D'autre part, nous introduisons une méthode d'apprentissage actif destinée à améliorer le plus rapidement possible la performance d'un classifieur basé sur une telle partition. La Figure 8.2 détaille la façon dont s'entremêlent ces deux algorithmes indépendants les uns des autres.

Une partition quelle qu'elle soit peut être vue comme un arbre de décision. Ceux-ci sont des graphes acycliques et unidirectionnels dont les nœuds contiennent un test sur les données d'entrée, et peuvent donc être vus comme une frontière dans l'espace d'entrée et les branches représentent le résultat de ce test. Les feuilles fournissent donc un ensemble de zones disjointes de l'espace d'entrée. Les arbres de classification sont des classifieurs qui associent à chaque feuille de l'arbre une étiquette à utiliser comme prédiction pour une donnée d'entrée guidée jusqu'à elle. En général, le classifieur attribue à chaque feuille l'étiquette majoritairement reçue. Dans le but de maximiser ses performances, un arbre de classification tente de partitionner l'espace en zones d'étiquettes homogènes. Ces arbres de classification sont appris de façon hors ligne en minimisant une mesure

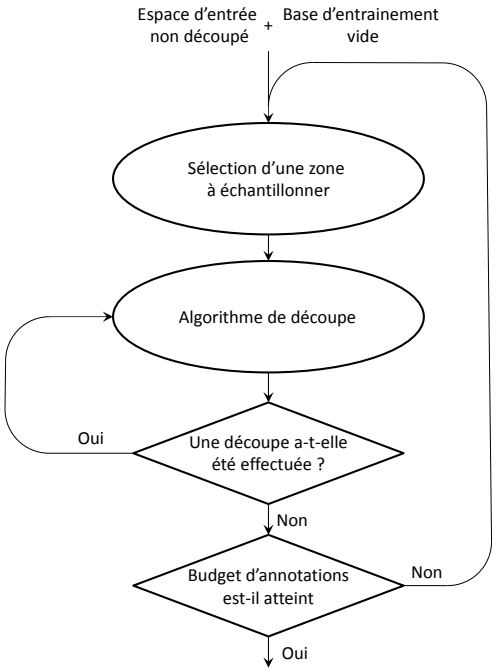


FIGURE 8.2 – Schéma de fonctionnement de l'algorithme d'apprentissage actif sur une partition adaptative

de l'hétérogénéité sur les étiquettes reçues. Et ce, en installant des tests binaires de façon récursive. Des exemples de mesures d'hétérogénéité sont l'entropie d'information ou la variance. Ces arbres de classification ont été adaptés au cas en-ligne sous l'appellation d'arbres incrémentaux. Dans ceux-ci, un arbre a la possibilité d'être étendu en ajoutant, à partir d'une feuille, un nouveau sous-arbre lorsque de nouvelles observations sont reçues. Ces arbres incrémentaux fournissent donc un cadre existant pour notre partition adaptative. Nous ne nous intéressons ici qu'au cas d'arbres incrémentaux dans lesquels les sous-arbres ne peuvent pas être supprimés. La difficulté majeure de cette approche est que cette dernière contrainte impose d'être confiant dans l'utilité d'une extension de l'arbre avant de l'effectuer. En effet, une extension trop hâtive impacterai une grosse pénalité sur les performances puisque le nombre d'annotations nécessaires pour pouvoir corriger cette erreur serait doublé. Il s'agit donc d'inclure une notion de prudence quant à l'inclusion d'un nouveau sous-arbre.

A la lumière des méthodes développées jusqu'ici pour le problème d'apprentissage actif, nous proposons l'utilisation d'une approche similaire dans le cadre des arbres incrémentaux. En effet, la nécessité d'agir prudemment est due à une incertitude sur la marche à suivre idéal, qui est fonction des paramètres du problème. Plus précisément, soit l'hétérogénéité réelle, qui dépend des paramètres de l'oracle dans chaque zone, qui est différente de l'hétérogénéité estimée qui dépend des étiquettes reçues. L'objectif est alors de diviser une zone en deux uniquement si cela fait décroitre l'hétérogénéité réelle. Cependant, cette hétérogénéité réelle n'est pas connue par avance et doit être estimée. Mais cette estimation est associée à une incertitude en suivant le nombre d'observations utilisées. Ainsi, l'idée est de diviser une zone uniquement si l'on croit avec certitude que l'hétérogénéité réelle va décroitre. Pour cela, des intervalles de confiances sont établis autours des hétérogé-

néités réelles précédent et suivant la découpe, puis le pire cas est analysé en considérant les bornes intérieures. Si les intervalles sont disjoints, la découpe a lieu. Une approche similaire a été utilisée dans [26] ou [30], dans lequel des intervalles de confiance étaient construits en utilisant l'inégalité de Hoeffding. Cette méthode donne de bons résultats avec un très grand budget d'annotations. Pourtant, lorsque celui-ci devient plus limité, cet algorithme affiche une inefficacité flagrante. Ceci s'explique par le fait que les inégalité de Hoeffding sont connues pour n'être étroites qu'à partir d'un certain nombre d'échantillons. Ceci implique d'être trop prudent initialement et donc parfois d'attendre trop longtemps avant de décider de découper alors que cela aurait été possible avant. Or, un petit budget ne permet pas de gaspiller des annotations de cette façon. Afin de contrecarrer cela, nous proposons d'utiliser des intervalles de crédibilité Bayésien, ce qui est rendu possible et aisé par la nature du problème de classification. De cette façon, les bornes obtenues sont étroites quel que soit le nombre d'observations car dérivant d'égalité de concentration. Ceci permet donc de découper une zone dès que possible et ainsi d'espérer atteindre des performances quasi-optimales.

L'autre partie de ce chapitre consiste à définir une stratégie de sélection fonctionnant sur cette partition adaptative. L'algorithme que nous proposons est très similaire à ceux de la partie précédente. A chaque instant, la partition adaptative peut être vue comme une partition fixe dans laquelle sélectionner une zone pour échantillonnage. La différence notable est que le risque réel inévitable dans une zone n'est pas limitant car une zone très hétérogène peut être re-découpée en deux zones très homogènes. Ainsi, le critère de sélection en connaissance parfaite ne tiendra pas compte de ce risque inévitable. De plus, le critère de sélection se basant sur la décroissance du risque ne sera pas justifié car celle-ci ne tient pas compte de la possibilité de découpe. Mise à part ces remarques, la démarche suivie reste identique à celle des chapitres précédents. Enfin, notez que les résultats présentés ici ont fait l'objet d'une publication [21].

Nous réaliserons ensuite des expériences qui viendront confirmer des algorithmes proposés. Ainsi, dans un premier temps, nous évaluerons la construction de la partition adaptative seule en la comparant d'abord à la méthode utilisant les inégalités de Hoeffding, puis à un algorithme hors-ligne, pouvant reconstruire entièrement l'arbre de décision pour chaque valeur du budget, et nous montrerons que notre méthode atteint des performances comparables. Dans un deuxième temps, nous inclurons la méthode d'apprentissage actif et la comparerons à celles utilisant une partition fixe afin de montrer l'évolution engendrée. Pour cela plusieurs bancs de tests seront utilisés, incluant des bases de données tirées du monde réel ainsi que des problèmes créés de toutes pièces.

8.2 Notations

Nous commençons avec cette section, dont le but est d'introduire les notations relatives aux arbres de classification dont la construction sera l'objet des algorithmes suivants ainsi qu'aux différents éléments accessibles pouvant être utilisés dans ce but.

Un arbre de décision, dont l'état au temps t est noté A_t, est un graphe constitué d'un ensemble de S_t nœuds indicés par $s \in \{1, \dots, S_t\}$. Ici, nous ne nous intéressons qu'au cas des arbres binaires dont chaque nœud est relié soit à deux autres appelés enfants, soit à aucun auquel cas ce nœud est appelé feuille. Chaque enfant ne peut avoir qu'un seul parent. La structure de l'arbre est donc définie par la fonction $parent : \{1, \dots, S_t\} \to \{0, \dots, S_t\}$ qui associe à chaque nœud son parent. Un nœud particulier, appelé racine ne possède pas de parent, la fonction renvoie 0, il n'en existe qu'un. Chaque nœud est associé à un test binaire permettant de guider une donnée d'entrée navigant dans l'arbre vers l'une ou l'autre branche. En théorie, ce test peut prendre n'importe quelle forme et les algorithmes développés dans la suite sont valables pour une partition possédant des zones de forme quelconque. Cependant, pour des raisons de simplicité, nous ne considérerons ici que le cas

où le test prend la forme d'un seuil sur un unique attribut de la donnée d'entrée. Soit a_s l'attribut considéré par le test relatif au nœud s, et c_s la valeur du seuil utilisée. L'arbre de décision permet donc de renvoyer, pour une donnée d'entrée $x \in X$ quelconque, l'indice correspondant à la feuille correspondante $A_t(x)$.

L'arbre de décision permet de définir une partition, où chaque zone correspond à une feuille de l'arbre. Soit N_t la partition relative à l'état de l'arbre au temps t. Le nombre de zones dans la partition correspond au nombre de feuilles, qui, dans un arbre binaire, est égal à $K_t = \frac{S_t - 1}{2} + 1$. Alors,

$$N_t = \{X_{1,t}, \dots, X_{K_t,t}\} \quad \text{t.q.} \quad \forall k \in \{1, \dots, K_t\}, \forall x \in X_{k,t}, A_t(x) = l_{k,t},$$

avec $l_{k,t}$ l'indice de la feuille k dans l'arbre A_t.

Soit, au temps t, l'ensemble des données d'entrée ayant été annotées $(x_i)_{i\in\{1,\dots,t\}}$ et leur étiquette respective $(y_i)_{i\in\{1,\dots,t\}}$, le tout formant la base d'apprentissage $\mathcal{T}_t = \{(x_i, y_i)\}_{i\in\{1,\dots,t\}}$. Soit $T_{k,t}$ le nombre d'observations reçues appartenant à la zone $k \in \{1, \dots, K_t\}$,

$$T_{k,t} = \sum_{i=1}^{t} \mathbb{1}_{x_i \in X_{k,t}},$$

et $\hat{\mu}_{k,t}$ la moyenne des étiquettes reçues dans cette zone,

$$\hat{\mu}_{k,t} = \frac{1}{T_{k,t}} \sum_{i=1}^{t} \mathbb{1}_{x_i \in X_{k,t}} y_i.$$

Soit $\mu_{k,t}$ le paramètre de la distribution de Bernoulli associée à la zone k au temps t. Nous nous plaçons ici dans le cadre de l'échantillonnage dans un réservoir. Ainsi, en plus des données d'entrée déjà annotées, nous avons accès à un grand nombre de données d'entrée non-étiquetées. Soit $W_{k,t}$ le nombre de données d'entrée non-étiquetées dans la zone k au temps t,

$$W_{k,t} = \sum_{x \in \mathcal{R}} \mathbb{1}_{x \in X_{k,t}}.$$

Cette élément nous permettra d'estimer et de remplacer la distribution naturelle des données d'entrée.

L'arbre de classification est le classifieur qui, pour une donnée d'entrée $x \in X$ quelconque, détermine la feuille à laquelle elle appartient et prédit pour elle l'étiquette majoritairement reçue dans celle-ci.

$$f(x) = \lfloor \hat{\mu}_{l_{A_t(x),t},t} \rceil,$$

où $\lfloor . \rceil$ est l'opérateur d'arrondi.

8.3 Un algorithme d'arbre incrémental utilisant les Intervalles Crédibles Bayésiens

L'apprentissage d'un arbre de classification consiste à choisir, en fonction des données contenues dans la base d'apprentissage, quels sont les tests les plus pertinents à mettre en place. Ainsi, pour chaque zone se pose la question de savoir si elle mérite d'être subdivisée et, si c'est le cas, selon quel attribut et avec quelle valeur du seuil. Ce processus est récursif, c'est à dire qu'il est

répété sur chaque zone nouvellement crée par l'ajout d'un nouveau test. Deux cas se présentent alors. En mode hors-ligne, l'algorithme d'entrainement est exécuté une seule fois, lorsque l'ensemble des données étiquetées est accessible. En mode en-ligne, un arbre initial est mis en place avant de n'avoir encore reçu aucune observation, puis l'algorithme d'entrainement est exécuté par dessus celui-ci après la réception de chaque nouvelle donnée étiquetée, sans possibilité de retirer ou de modifier des branches de l'arbre existant, on parle alors de mise-à-jour. Dans les deux cas, le corps de l'algorithme d'entrainement est le même, la différence provient uniquement de l'expression du critère qui permet de décider du test à mettre en place, et des données auxquelles il a accès. Cet algorithme est décrit sur la Figure 5, le critère utilisé C est générique. La paramètre d est le nombre d'attributs possibles. Notez que les seuls seuils considérés sont ceux équidistants

Données : $noeud$, $\mathcal{T}_t$,$(x_i)_{i\in[\![1,n]\!]}$
Calculer T_{noeud}, $\hat{\mu}_{noeud}$ et W_{noeud}
pour $a = 1 \dots d$ **faire**
  $(x^{tri}_{i,a})_{i\in[\![1,t]\!]} = tri((x_{i,a})_{i\in[\![1,t]\!]})$
  pour $i = 1$ *to* $t-1$ **faire**
    si $x^{tri}_{i,a} \neq x^{tri}_{i+1,a}$ **alors**
      $c_{i,a} = \frac{x^{tri}_{i,a}+x^{tri}_{i+1,a}}{2}$
      Soit X_{c_1} et X_{c_2} deux zones résultant de la découpe de X_{noeud} suivant l'attribut a par le seuil $c_{i,a}$
      Calculer T_{c_1}, $\hat{\mu}_{c_1}$, W_{c_1}, T_{c_2}, $\hat{\mu}_{c_2}$, W_{c_2}
      Calculer $C_{i,a} = C(T_{noeud}, \hat{\mu}_{noeud}, W_{noeud}, T_{c_1}, \hat{\mu}_{c_1}, W_{c_1}, T_{c_2}, \hat{\mu}_{c_2}, W_{c_2})$
    fin
  fin
fin
$\{i^{best}, a^{best}\} = \arg\min_{i,a} C_{i,a}$
si $C_{i^{best},a^{best}} < 0$ **alors**
  Créer deux branches partant de $noeud$ contenant un test suivant l'attribut a^{best} avec le seuil $c_{i^{best},a^{best}}$ et menant aux nœuds $enfant_1$ and $enfant_2$
  $enfant_1 = Entrainement(enfant_1, \mathcal{T}_t, (x_i)_{i\in[\![1,n]\!]})$
  $enfant_2 = Entrainement(enfant_2, \mathcal{T}_t, (x_i)_{i\in[\![1,n]\!]})$
fin
Résultat : $Tree$

Algorithme 5 : Algorithme d'entrainement d'un arbre de classification

des données les plus proches selon l'attribut considéré. En effet, puisque le critère se base sur les étiquettes associées aux données d'entrée dans chaque zone, tous les seuils séparant de la même manière les données d'entrée sont égaux du point de vue de l'algorithme.

L'Algorithme 6, décrit le processus de construction d'un arbre de classification en-ligne. A chaque pas de temps, une nouvelle donnée d'entrée est échantillonnée et incluse dans la base d'entrainement. L'arbre est ensuite mis à jour. Notez que la seule zone qui voit ses valeurs associées changer est celle d'où est tirée la donnée d'entrée. Ainsi, seule cette dernière est soumise à l'algorithme d'entrainement.

Précisons que le critère est nécessairement différent pour le cas en-ligne que pour le cas hors-ligne car il doit prendre en compte le fait qu'il est impossible de revenir sur une décision de découpe. Dans les sous-sections suivantes, le but va donc être de définir un critère C faisant preuve de

Données : $X = \mathbb{R}^d$, $(x_i)_{i\in[\![1,n]\!]}$
Initialiser A_1 n'incluant qu'une seule feuille associée à l'ensemble de l'espace d'entrée X.
$\mathcal{T}_0 = \emptyset$ **pour** $t = 1 \dots n$ **faire**
  Recevoir une étiquette y_t pour la donnée d'entrée x_t
  $\mathcal{T}_t = \mathcal{T}_{t-1} \cup \{(x_t, y_t)\}$ Soit $noeud = A_t(x_t)$
  $SousArbre = Entrainement(noeud, \mathcal{T}_t, (x_i)_{i\in[\![1,n]\!]})$
  Remplacer $noeud$ par $SousArbre$
fin
Résultat : A_n

Algorithme 6 : Core algorithm

prudence lorsqu'il s'agit de découper une zone dans le cas en-ligne.

8.3.1 Critère de découpe en connaissance parfaite

La prudence dont doit faire preuve l'algorithme d'entrainement vis-à-vis de la décision de découpe d'une zone provient de l'incertitude quant à l'utilité de cette dernière. En effet, comme nous l'avons expliqué précédemment, une mauvaise découpe imposerait une forte pénalité sur la performance maximale du classifieur pouvant être atteinte en budget fini. Le critère utilisé par un algorithme hors-ligne se base sur l'hétérogénéité des étiquettes reçues dans les zones précédent et suivant la découpe. Pourtant, lorsque peu d'étiquette ont été reçues, cette hétérogénéité n'est pas significative vis-à-vis de l'utilité de la découpe. Ainsi, il peut être plus judicieux d'attendre d'avoir reçu quelques étiquettes supplémentaires afin d'affirmer avec certitude que la découpe est utile ainsi que d'affiner la sélection du test à mettre en place. Dans le but de gérer ce dilemme entre exploitation (découpe en fonction de l'hétérogénéité mesurée) et exploration (attente d'une meilleure estimation), l'idée est ici de considérer qu'il existe une mesure de l'utilité idéale que l'on cherche à suivre. Celle-ci pouvant dépendre de paramètres inconnus initialement, elle pourra ensuite être estimée à travers les observations reçues en même temps qu'associée à une incertitude. Dans cette sous-section, nous montrons quelle est la mesure de l'utilité utilisée dans notre algorithme de construction d'arbre en-ligne.

Tout d'abord, tachons de mentionner l'existence d'une hétérogénéité réelle. En effet, une zone k de l'arbre au temps t étant associée à un paramètre de Bernoulli $\mu_{k,t}$, les étiquettes reçues suivrons cette proportion. L'hétérogénéité d'une zone sert à définir à quel point les étiquettes reçues sont disparates. C'est ainsi une fonction h du paramètre $\mu_{k,t}$, symétrique par rapport à $\frac{1}{2}$ et croissante sur l'intervalle $[0, \frac{1}{2}]$, et nulle en 0 et 1. Les algorithmes d'arbre de classification existants tentent partitionner l'espace de telle sorte que l'hétérogénéité dans chaque zone soit minimale. En effet, l'erreur inévitable, et donc l'opposé de la performance maximale atteinte par le classifieur, est une fonction croissante de l'hétérogénéité. Dans ceux-ci, l'hétérogénéité peut prendre les formes suivantes [28] :

- entropie d'information : $h(\mu_{k,t}) = -\mu_{k,t}\log(\mu_{k,t}) - (1-\mu_{k,t})\log(1-\mu_{k,t})$,
- variance : $h(\mu_{k,t}) = \mu_{k,t}(1-\mu_{k,t})$,
- écart type : $h(\mu_{k,t}) = \sqrt{\mu_{k,t}(1-\mu_{k,t})}$,
- risque réel optimal : $h(\mu_{k,t}) = \frac{1}{2} - |\mu_{k,t} - \frac{1}{2}|$.

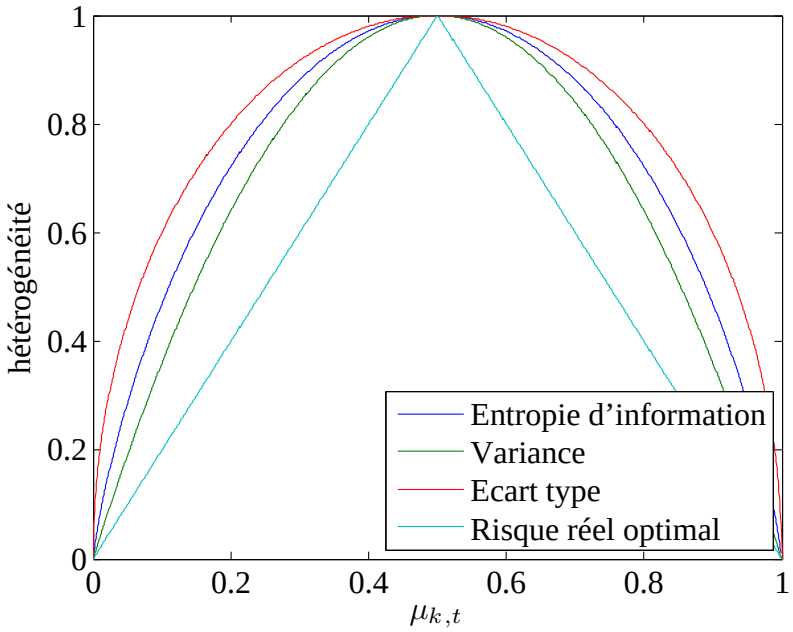


FIGURE 8.3 – Différentes mesures d'hétérogénéité (mises à l'échelle)

Ces différentes mesures d'hétérogénéités sont représentées sur la Figure 8.3. A première vue, le risque réel optimal semble le plus adapté, puisqu'il est directement lié à l'objectif visé, c'est à dire l'augmentation des performances maximales. Rappelons qu'on ne s'intéresse ici qu'au risque optimal, c'est à dire celui atteint lorsque le classifieur prédit la meilleure étiquette. En effet, le but est ici de déterminer la localisation de chaque zone de façon indépendante de l'apprentissage de l'étiquette à prédire. Cependant, l'utilisation de celui-ci engendre un comportement non-désiré. Dans certains cas, une zone à l'intérieur de laquelle les données sont regroupées en deux classes bien séparées dans l'espace ne se voit jamais divisée car peu importe l'endroit de la première découpe, les deux zones ainsi crée possèdent toujours la même prédiction optimale. Alors que plusieurs découpes successives permettraient de partitionner la zone en zones complètement homogènes. Nous pensons que cet aspect myope est un conséquence de la linéarité de la fonction d'hétérogénéité utilisée. Ce phénomène n'apparait pas lorsque l'hétérogénéité utilisée est concave. Dans les expériences, la variance sera privilégiée. Toutefois, dans la suite, les algorithmes seront définis pour une mesure d'hétérogénéité h quelconque.

Le critère en connaissance parfaite permettant de définir si un test doit être réalisé ou non, ainsi que de sélectionner le test le plus utile est alors la décroissance de l'hétérogénéité. Ainsi,

$$C(W_p, \mu_{p,t}, W_{e_1}, \mu_{e_1,t}, W_{e_2}, \mu_{e_2,t}) = W_{e_1} h(\mu_{e_1,t}) + W_{e_2} h(\mu_{e_2,t}) - W_p h(\mu_{p,t}).$$

L'hétérogénéité est pondérée par l'importance relative de chaque zone pour privilégier les découpes en parts égales. En effet, dans le cas contraire, l'algorithme créerait de très petites zones. De plus, dans le cas du risque optimal, ceci est totalement justifié car l'erreur commise dans une zone doit être pondérée par le taux d'utilisation de celle-ci.

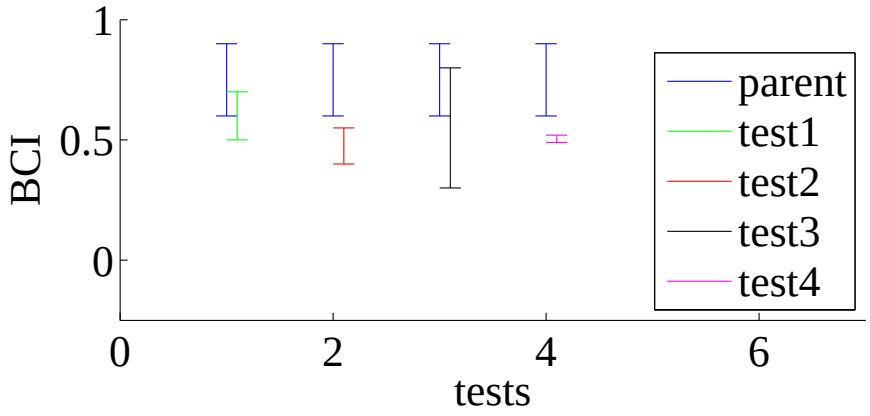


FIGURE 8.4 – Intervalles de crédibilité relatifs aux hétérogénéités des zones issues de différents tests et de la zone initiale

8.3.2 Gestion de l'incertitude sur le critère de découpe

Le critère défini dans la partie précédente dépend de la valeur des paramètres de l'oracle dans chaque zone. En pratique, celles-ci ne sont pas connues par avance. Il s'agit donc d'estimer la valeur du critère à travers les observations acquises jusqu'ici. De façon à gérer l'incertitude relative à ce critère, des intervalles de crédibilité peuvent être construits en fonction de l'estimation des paramètres de l'oracle et du nombre d'observations qui ont été utilisées. La prudence dont doit faire preuve l'algorithme vis-à-vis de la découpe peut alors être gérée en travaillant avec la borne supérieure de cet intervalle. Ainsi, cela garantit que le vrai critère est inférieur à cette borne avec une forte probabilité.

Une autre interprétation peut être effectuée en traitant indépendamment les deux parties du critère correspondant à l'état précédent et suivant la découpe. Un intervalle de crédibilité peut alors être construit pour chacune de ces parties. L'idée est alors d'autoriser la découpe uniquement si ces intervalles sont disjoints. En effet, dans ce cas, il existe une forte probabilité pour que l'hétérogénéité avant la découpe soit supérieure à l'hétérogénéité après la découpe. Ceci est illustré sur la Figure 8.4 dans laquelle les intervalles de crédibilité sont représentés pour différents tests. Dans le cas présenté ici, seuls les tests 2 et 4 seront envisagés car les intervalles sont disjoints, et le test 4 sera finalement choisi car il possède la plus grande marge entre les deux intervalles. Dans la suite, nous montrons comment obtenir de tels intervalles.

Soit une zone quelconque k de l'arbre au temps t. L'hétérogénéité pour cette zone est $h(\mu_{k,t})$. Par inférence Bayésienne, la distribution *a posteriori* sur $\mu_{k,t}$ suit une loi Beta de paramètre $T_{k,t}\hat{\mu}_{k,t}+1$ et $T_{k,t}(1-\hat{\mu}_{k,t})+1$. Soit μ_{basse} et μ_{haute} les deux fonctions inverses de h sur $[0,\frac{1}{2}]$ et $[\frac{1}{2},1]$. Ainsi,

$$\forall \epsilon_{k,t} \in [0, h(\frac{1}{2})], \mu_{basse}(\epsilon_{k,t}) \in [0,\frac{1}{2}] \text{ t.q. } h(\mu_{basse}(\epsilon_{k,t})) = \epsilon_{k,t}) \text{ et}$$
$$\forall \epsilon_{k,t} \in [0, h(\frac{1}{2})], \mu_{haute}(\epsilon_{k,t}) \in [\frac{1}{2},1] \text{ t.q. } h(\mu_{haute}(\epsilon_{k,t})) = \epsilon_{k,t}).$$

Soit la fonction g telle que

$$\forall \epsilon_{k,t} \in [0, h(\frac{1}{2})], g(\epsilon_{k,t}) = \mathbb{P}_{\mu_{k,t} \sim Beta(T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1)}(h(\mu_{k,t}) \geq \epsilon_{k,t}).$$

Ainsi,

$$\forall \epsilon_{k,t} \in [0, h(\frac{1}{2})], g(\epsilon_{k,t}) = \mathcal{I}_{\mu_{haute}(\epsilon_{k,t})}(T_{k,t}\hat{\mu}_{k,t} + 1, T_{k,t}(1 - \hat{\mu}_{k,t}) + 1) \\ - \mathcal{I}_{\mu_{basse}(\epsilon_{k,t})}(T_{k,t}\hat{\mu}_{k,t} + 1, T_{k,t}(1 - \hat{\mu}_{k,t}) + 1),$$

avec $\mathcal{I}_{\cdot}(\alpha, \beta)$ la fonction de répartition d'une loi Beta de paramètres α et β. La fonction g est strictement décroissante, elle admet donc un inverse g^{-1}. Nous pouvons donc définir $\forall \delta \in [0, \frac{1}{2}]$,

$$\epsilon^{sup}_{k,t}(\delta) = g^{-1}(\delta) \text{ et}$$
$$\epsilon^{inf}_{k,t}(\delta) = g^{-1}(1 - \delta).$$

Ainsi, l'intervalle de crédibilité $[\epsilon^{inf}_{k,t}(\delta), \epsilon^{sup}_{k,t}(\delta)]$ contient $h(\mu_{k,t})$ avec une probabilité $1 - 2\delta$.

Soit le test considéré s'appliquant sur une zone parent p menant aux deux zones enfants e_1 et e_2. En combinant les deux équations suivantes :

$$\mathbb{P}(h(\mu_{c_1,t}) \geq \epsilon^{sup}_{c_1,t}(\delta)) = \delta \text{ et}$$
$$\mathbb{P}(h(\mu_{c_2,t}) \geq \epsilon^{sup}_{c_2,t}(\delta)) = \delta,$$

nous obtenons

$$\mathbb{P}(W_{c_1,t}h(\mu_{c_1,t} + W_{c_2,t}h(\mu_{c_2,t}) \geq W_{c_1,t}\epsilon^{sup}_{c_1,t}(\delta) + W_{c_2,t}\epsilon^{sup}_{c_2,t}(\delta)) \leq 2\delta.$$

En combinant cette dernière avec

$$\mathbb{P}(h(\mu_{p,t} \leq \epsilon^{inf}_{p,t}(\delta)) = \delta,$$

nous obtenons

$$\mathbb{P}(W_{c_1,t}h(\mu_{c_1,t} + W_{c_2,t}h(\mu_{c_2,t}) - W_{p,t}h(\mu_{p,t} \geq W_{c_1,t}\epsilon^{sup}_{c_1,t}(\delta) + W_{c_2,t}\epsilon^{sup}_{c_2,t}(\delta) - W_{p,t}\epsilon^{inf}_{p,t}(\delta)) \leq 3\delta.$$

Ainsi, nous définissons le critère de découpe comme

$$C(T_{p,t}, \hat{\mu}_{p,t}, W_{p,t}, T_{c_1,t}, \hat{\mu}_{c_1,t}, W_{c_1,t}, T_{c_2,t}, \hat{\mu}_{c_2,t}, W_{c_2,t}) = W_{c_1,t}\epsilon^{sup}_{c_1,t}(\delta) + W_{c_2,t}\epsilon^{sup}_{c_2,t}(\delta) - W_{p,t}\epsilon^{inf}_{p,t}(\delta).$$

Ainsi, si δ est petit, il y a une forte probabilité que le critère en connaissance parfaite soit inférieur à celui-ci.

Dans cette sous-section, nous venons de mettre en place le critère qui peut être utilisé dans l'algorithme d'entrainement d'un arbre de classification en-ligne. Ce critère est soumis à un paramètre : δ, qui permet de régler la confiance que l'on donne à l'estimation courante et à quel point il est nécessaire d'attendre.

8.4 Apprentissage Actif en partition adaptative

Le résultat de la section précédente fournit déjà une contribution en soit. Le classifieur ainsi défini peut être utilisé comme tel en tant qu'alternative à un algorithme d'arbre incrémental. Une autre perspective envisageable serait d'utiliser ce classifieur dans le contexte de l'apprentissage actif, et de voir comment ses performances peuvent être maximisées sous un budget d'annotations fini. C'est ce que nous allons voir dans cette section. Nous nous intéressons ici à la construction d'une stratégie de sélection qui va permettre de choisir l'ordre dans lequel échantillonner les zones dans le but d'améliorer la prédiction du classifieur le plus rapidement possible. Pour cela, nous utiliserons une approche basée sur le principe d'optimisme face à l'incertitude, suivant la même démarche que pour les algorithmes des chapitres précédents.

8.4.1 Critère de sélection en connaissance parfaite

Il s'agit donc tout d'abord de définir le critère de sélection en connaissance parfaite qui nous indiquera la marche à suivre et qui sera tenté d'être respecté alors que les paramètres seront mêlés d'incertitude. A chaque instant, la partition adaptative peut être vue comme une partition fixe. Il peut alors être envisagé d'utiliser directement un algorithme efficace dans ce contexte. Pourtant, une subtilité entre en jeu dans le cas des partitions adaptatives. En effet, rappelons que dans le cas des partitions fixes, une zone possédant un haut risque courant mais dont le risque optimal était aussi très haut, à cause d'un paramètre proche de $\frac{1}{2}$, n'était pas considéré intéressant à échantillonner car le risque ne pouvait en aucun cas décroitre dans une telle zone. Ici, une zone avec un critère proche de $\frac{1}{2}$ peut potentiellement contenir deux sous zones très homogènes, l'échantillonnage d'une telle zone peut donc mener à sa découpe et donc à un risque optimal très faible. Étant donné qu'il est impossible de faire la différence entre une zone complètement hétérogène ou bien localement homogène uniquement grâce à la connaissance du paramètre, une forte valeur de ce dernier ne doit pas pénaliser le critère d'échantillonnage. De même, le critère basé sur la décroissance du risque implique une évaluation de l'effet sur le long terme qui n'est plus valable lorsque la partition peut être modifiée. Le critère de sélection en connaissance parfaite est donc défini par le risque maximal affecté à la zone et pondéré par son importance relative afin de privilégier les zones les plus utilisées.

De cette façon, un zone k de l'arbre au temps t possède un paramètre $\mu_{k,t}$ et une importance relative $W_{k,t}$. Si l'étiquette prédite pour cette zone est $f_{k,t}$, alors le risque réel engendré est

$$R_{k,t} = \mu_{k,t}\mathbb{1}_{f_{k,t}=0} + (1-\mu_{k,t})\mathbb{1}_{f_{k,t}=1}.$$

Ainsi, le critère idéal est

$$C_{k,t} = R_{k,t},$$

ce qui mène à la stratégie de sélection suivante :

$$k_{t+1} = \underset{k\in\{1,\dots,K_t\}}{\arg\max}\ C_{k,t}.$$

8.4.2 Prise en compte de l'incertitude

Le critère qui vient d'être défini repose sur la connaissance des paramètres de la distribution de l'oracle qui sont en réalité inconnus initialement. Cette stratégie peut toutefois être approchée en utilisant l'information apportée par les étiquettes reçues. L'idée est alors d'utiliser le principe de l'optimisme face à l'incertitude. Nous montrons ici comment construire les intervalles de confiance dont la borne supérieure sera utilisée par l'algorithme de sélection de la zone à échantillonner.

Soit une zone k de l'arbre au temps t. Son paramètre $\mu_{k,t}$ est inconnu, mais elle a reçu $T_{k,t}$ étiquettes de moyenne $\hat{\mu}_{k,t}$. En utilisant l'inférence Bayésienne, une distribution *a posteriori* peut être considérée pour $\mu_{k,t}$. Ainsi, $\mu_{k,t}$ suit une loi Beta de paramètres $T_{k,t}\hat{\mu}_{k,t}+1$ et $T_{k,t}(1-\hat{\mu}_{k,t})+1$.

De cette façon, la distribution *a posteriori* peut être étendue au risque réel, ce qui donne $\forall \epsilon_{k,t} \in [0,1]$

$$\mathbb{P}_{\mu_{k,t}\sim Beta(T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1)}(R_{k,t} \leq \epsilon_{k,t}) = \begin{cases} \mathcal{I}_{\frac{\epsilon_{k,t}}{W_{k,t}}}(T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1) \text{ si } f_{k,t}=0 \\ 1-\mathcal{I}_{\frac{1-\epsilon_{k,t}}{W_{k,t}}}(T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1) \text{ si } f_{k,t}=1, \end{cases}$$

Données : X, $(x_i)_{i\in[\![1,n]\!]}$, δ_1, δ_2
Initialize A_1 n'incluant qu'une unique feuille associée à l'ensemble de l'espace d'entrée X.
pour $t = 1, \dots, n$ **faire**
  pour $k = 1, \dots, K_t$ **faire**
    Calculer $\hat{\mu}_{k,t}$, $T_{k,t}$ and $W_{k,t}$
    Calculer $\epsilon_{k,t}(\delta_2)$ (8.1)
  fin
  $k_{t+1} = \arg\max_k \epsilon_{k,t}$
  Recevoir une étiquette y_t pour la donnée d'entrée $x_t \in X_{k_{t+1}}$
  $noeud = A_t(x_t)$
  $SousArbre = Entrainement(noeud, (x_i)_{i\in[\![1,n]\!]}, (y_i)_{i\in[\![1,t]\!]}, \delta_1)$
  Remplacer $noeud$ par $SousArbre$
fin
Résultat : A_n

Algorithme 7 : Algorithme d'apprentissage actif utilisant une partition adaptative

avec $\mathcal{I}_{\cdot}\alpha, \beta$ la fonction de répartition d'une loi Beta de paramètres α et β. Nous pouvons ainsi définir une borne supérieure sur le risque réel en fonction d'un paramètre δ_2

$$\mathbb{P}(R_{k,t} \geq \epsilon_{k,t}(\delta_2)) = \delta_2 \iff \epsilon_{k,t}(\delta_2) = W_{k,t} \times \dots \tag{8.1}$$
$$\begin{cases} \mathcal{I}^{-1}(1-\delta_2, T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1) & \text{if } f_{k,t}=0 \\ (1-\mathcal{I}^{-1}(\delta_2, T_{k,t}\hat{\mu}_{k,t}+1, T_{k,t}(1-\hat{\mu}_{k,t})+1)) & \text{if } f_{k,t}=1, \end{cases}$$

avec $\mathcal{I}_{\cdot}^{-1}\alpha, \beta$ la fonction de répartition inverse d'une loi Beta de paramètres α et β. La stratégie de sélection consiste donc à échantillonner la zone

$$k_{t+1} = \underset{k\in\{1,\dots,K_t\}}{\arg\max}\ \epsilon_{k,t}(\delta_2).$$

L'algorithme résultant est décrit dans la Figure 7. Il prend en entrée deux paramètres δ_1 et δ_2 qui servent respectivement à gérer l'incertitude dans la décision de découpe et le choix de la zone à échantillonner.

8.5 Expériences

Dans cette section, nous présentons les deux bancs de tests qui seront utilisés pour évaluer et comparer les algorithmes définis dans cette section. Le premier est un problème jouet, c'est à dire crée de toutes pièces afin de pouvoir contrôler avec précision ses paramètres. Ceci permettant en outre d'examiner et d'interpréter les comportement des algorithmes au plus bas niveau. Le deuxième est constitué de plusieurs problèmes tirés du monde réel, les paramètres réels sont donc inconnus, mais cela donne un exemple de ce qui pourrait se passer en pratique lors de l'utilisation de ces algorithmes.

Problème jouet : Dans ce problème, la structure de l'arbre maximal (le plus développé) pouvant être obtenu est prédéfinie. Alors que dans une problème réel, lorsqu'une zone est considérée pour être découpée, le choix de la façon dont cela est effectué est laissé libre à l'algorithme, le but étant de trouver la meilleure paire attribut-seuil, ici, une seule découpe est possible, le rôle de l'algorithme

étant uniquement de décider si celle-ci doit être effectuée ou non. Pour construire ce problème, les paramètres et les importances relatives de chaque feuille de l'arbre prédéfini sont fixés. Cette information est ensuite propagée dans le reste de l'arbre pour connaitre les paramètres associés à chaque nœud.

Dans nos expériences, l'arbre prédéfini est un arbre binaire de profondeur 6. Ainsi, le nombre de feuilles maximales est de $2^6 = 64$. Les paramètres utilisés correspondent à un problème dans un espace en deux dimensions $[0, 1]^2$ découpé alternativement selon celles-ci. Les étiquettes de la zone $k \in \{1, \ldots, 64\}$ sont tirées selon

$$\mu_k = \sigma(5(x_{2,k} - \sqrt{x_{1,k}})),$$

où σ est la fonction sigmoïde : $\forall x \in \mathbb{R}, \sigma(x) = \frac{1}{1+e^{-x}}$, et $x_{1,k}$ et $x_{2,k}$ représentent les coordonnées du centre de la zone k. L'importance relative de chacune de ces zones est égale à $\frac{1}{64}$.

A chaque pas de temps, l'arbre de classification utilisé sera nécessairement un sous-arbre de l'arbre prédéfini. De plus, cet arbre ne peut être qu'étendu, tous les nœuds du sous-arbre à l'instant t doivent obligatoirement faire partie du sous-arbre à l'instant $t + 1$. L'algorithme aura donc pour rôle de décider de débloquer ou non certains nœuds de l'arbre prédéfini pour être inclus dans l'arbre de classification. A chaque pas de temps une étiquette est tirée. Dans le cas où le contexte n'est pas celui de l'apprentissage actif (ou si une stratégie aléatoire est utilisée), la zone de provenance de cette étiquette est une feuille de l'arbre prédéfini tirée aléatoirement selon un probabilité proportionnelle à son importance relative. Dans l'autre cas, cette zone est déterminée par la stratégie de sélection. Notez que pour localiser les données d'entrée étiquetées dans les différentes zones résultant d'une découpe, il est nécessaire de les tirer directement dans les feuilles de l'arbre prédéfini, celle-ci est alors choisie aléatoirement parmi celles descendant de la zone souhaitée.

Notez que le fait de connaitre les paramètres réels du problème donne la possibilité de tirer les étiquettes directement selon une distribution de Bernoulli avec le paramètre correspondant à la zone considérée. Cela possède l'avantage que le nombre de tirages possibles ne soit pas limité comme cela peut être le cas lors de l'échantillonnage dans un réservoir. De plus, le risque réel peut être directement calculé sans nécessiter de passer par une base de test.

Pour ce banc de test, le nombre d'exécutions de l'algorithme a été fixé à 100 avec un budget de 1000 annotations. Le risque réel provenant de chaque exécution de l'algorithme a été enregistré à chaque pas de temps et la performance globale est calculée comme la moyenne du risque réel sur les exécutions.

Le but de ce banc de tests est de comparer les performance relatives à l'utilisation d'une partition adaptative à celles relatives à une partition fixe. En effet, dans le cas d'un problème réel, le choix de la partition fixe utilisée aurait été déterminant pour les performances observées, de plus qu'il n'aurait pas été évident. Ici, les partitions choisies correspondent à des arbres de profondeur fixe. Les algorithmes utilisés pour la comparaison sont donc :

- **OALC-FP**$_8^{dd,m}$: Un algorithme du chapitre précédent, utilise le principe d'optimisme face à l'incertitude pour définir la stratégie de sélection, le critère de sélection en connaissance parfaite est basé sur le risque local maximal, la partition est fixe et comporte 8 zones chacunes issues de trois découpes successives.

- **OALC-FP**$_{64}^{dd,m}$: Un algorithme du chapitre précédent, utilise le principe d'optimisme face à l'incertitude pour définir la stratégie de sélection, le critère de sélection en connaissance parfaite est basé sur le risque local maximal, la partition est fixe et comporte 64 zones chacunes issues de six découpes successives.

- **BOCT** : L'algorithme de construction d'un arbre de classification défini dans ce chapitre, utilisant des intervalles de crédibilité Bayésien, sans sélection active des zones à échantillonner.

- **BOCT-AL** : L'algorithme de construction d'un arbre de classification défini dans ce chapitre, utilisant des intervalles de crédibilité Bayésien, avec sélection active des zones à échantillonner, utilisant le principe de l'optimisme face à l'incertitude pour la définition du critère.

Bases de données tirées du monde réel : Ce second banc de tests utilise des bases de données provenant du UCI Machine Learning Repository [56]. Ces bases sont issues du monde réel et montrent un exemple de ce que peut donner l'utilisation de ces algorithmes en pratique. Lors de chaque execution d'un algorithme sur l'une de ces bases, celle-ci est divisée en deux partie égales. L'une sert à l'apprentissage et constitue, lorsque le contexte est celui de l'apprentissage actif, le réservoir de données dans lequel piocher des données pour être étiquetées. Les données non-étiquetées présentes dans le réservoir servent tout de même à estimer la distribution naturelle des données d'entrée et à calculer l'importance relative des zones qui seront créées. L'autre est réservée uniquement à l'évaluation des performances des classifieurs ainsi appris. Aucune de ces données ne peut être accédée par l'algorithme dans la phase d'apprentissage. A chaque pas de temps, le classifieur est appris avec les données étiquetées et celui-ci effectue une prédiction sur les données d'entrée de la base de test. La performance est ensuite évaluée et enregistrée à chaque pas de temps à travers le nombre de données mal classifiées sur cette base.

Pour ce banc de test, le nombre d'exécution de l'algorithme a été fixé à 1000 avec un budget d'annotations équivalent au nombre de données dans le réservoir. La performance globale est calculée à chaque pas de temps comme la moyenne des performances sur les exécutions.

Les algorithmes utilisés pour la comparaison sont :

- **c4.5** : Un algorithme hors-ligne de construction d'un arbre de classification, son critère de découpe est l'entropie d'information calculée sur les étiquettes reçues, l'arbre est reconstruit entièrement à chaque pas de temps, il n'inclut donc pas de notion de prudence,

- **Hoeffding** : Un algorithme en-ligne d'arbre incrémental adapté des résultats de [26], son critère de découpe est l'entropie d'information et prend en compte une incertitude calculée grâce à l'inégalité de Hoeffding.

Tous les algorithmes optimistes ici présents sont conditionnés par un paramètre δ permettant de contrôler la quantité d'exploration. Comme mentionné au début de cette partie, notre but est uniquement d'évaluer la viabilité du principe de l'optimisme face à l'incertitude pour notre problème. Ainsi, nous souhaitons montrer que de bonnes performances sont possible pour une valeur arbitraire du paramètre. La question de comment trouver sa valeur optimale étant pour l'instant laissée de côté. De ce fait, pour tous les algorithmes optimistes utilisés, ceci concernant aussi bien l'algorithme de découpe que la stratégie de sélection, la valeur du paramètre δ a été réglée en effectuant une recherche par quadrillage.

8.6 Résultats

Dans cette section, nous évaluons les performances des algorithmes introduits dans cette section et les comparons à celles de quelques algorithmes sélectionnés soit pour leur statut de référence, soit parce qu'ils utilisent une approche semblable. Pour cela, nous utilisons les banc de tests présentés dans la section précédentes. Afin d'évaluer les performances, les courbes que nous montrons

représentent le risque réel moyenné sur les exécutions en fonction du nombre de données étiquetées utilisées.

Problème jouet : Commençons par décrire les résultats obtenus sur le problème jouet. Ceux-ci sont affichés sur la Figure 8.5. Le but de cette expérience est de montrer que le critère de découpe dans l'algorithme de partition adaptative est bien calibré dans le sens où il arrive à ce que la découpe intervienne au meilleur moment. Pour cela, nous le comparons à deux algorithmes équivalent utilisant une partition fixe. La première possédant un faible nombre de zones, ce qui lui permet d'apprendre avec plus de précision les paramètres de chaque zone mais implique que la performance optimale soit restreinte à cause du fait qu'une seule étiquette ne peut être attribuée pour chaque zone. La deuxième possédant un nombre important de zone, ce qui lui permet de discrétiser plus finement l'espace d'entrée et donc d'obtenir une meilleure performance de classification optimale, mais possédant en contrepartie une lacune vis-à-vis de la vitesse d'apprentissage, ceci étant dû au fait que les paramètres doivent être appris indépendamment dans chaque zone, ce qui nécessite pour une même précision sur le paramètre, un nombre d'annotations proportionnel au nombre de zones. Ces remarques peuvent être observées sur les courbes. Nous voyons que OALC-FP$_8^{dd,m}$ possède une très bonne performance initiale mais qui converge vers une valeur plus forte du risque réel. OALC-FP$_{64}^{dd,m}$, pour sa part, arrive à une très bonne performance finale mais n'est pas très efficace au début.

En observant les performances de BOCT-AL, nous voyons qu'il est à la fois très bon au début et à la fin. En effet, il permet de commencer avec un faible nombre de zones et de l'accroitre lorsque cela est possible et nécessaire. Remarquez qu'en plus de cela, les performances initiales de BOCT-AL sont les mêmes que celles de OALC-FP$_8^{dd,m}$ et les performances finales sont les mêmes que celles de OALC-FP$_{64}^{dd,m}$. Cela veut dire que l'algorithme est capable de tirer parti de la meilleure solution. Ainsi, cela revient à considérer simultanément deux partitions fixes et à basculer de l'une à l'autre pour profiter des meilleures performances. De plus, non seulement il est capable d'effectuer ce changement mais il le fait au bon moment. En effet, un mauvais timing dans la

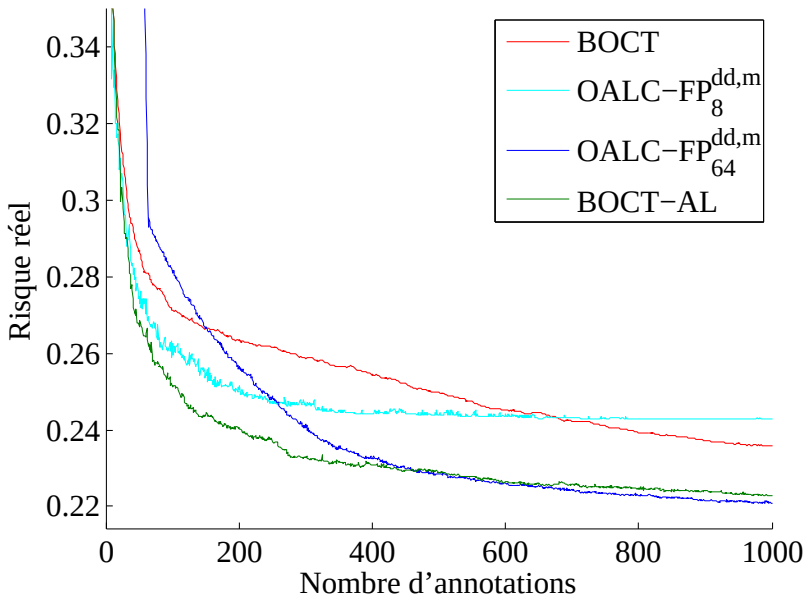
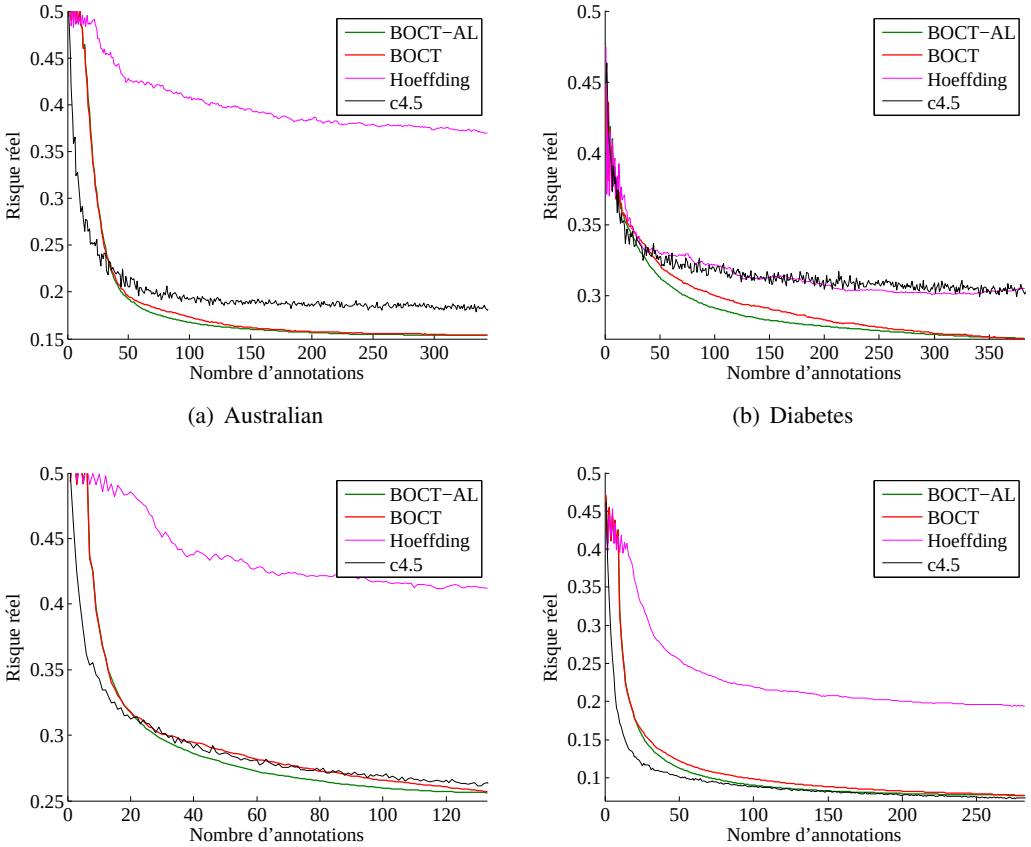


FIGURE 8.5 – Résultats sur la problème jouet



(a) Australian (b) Diabetes

(c) Heart (d) Wdbc

FIGURE 8.6 – Résultats sur les bases de données tirées du monde réel

décision affecterait grandement les performances. Ici, les résultats montrent que la l'algorithme est capable de gérer efficacement le compromis entre vitesse d'apprentissage et performance optimale.

Un autre constat est que l'algorithme possédant une stratégie de sélection des zones à échantillonner est bien meilleur que celui dont les données d'entrée étiquetées sont reçues aléatoirement. Ainsi, il est capable de privilégier les zones, qui soit peuvent voir leur performance améliorée directement en acquérant une meilleure précision sur le paramètre estimé, soit peuvent potentiellement être re-découpées pour améliorer l'homogénéité des zones résultantes.

Bases de données tirées du monde réel : Concentrons nous maintenant sur les résultats des expériences menées sur les bases de données tirées du monde réel. Les performances obtenues sont affichées sur la Figure 8.6. L'objectif premier de ces expériences porte plutôt sur l'approbation de la méthode d'entrainement des arbres incrémentaux introduit dans ce chapitre. Dans un second temps, nous évaluerons l'apport de la stratégie d'apprentissage actif.

Tout d'abord, comparons les performances de deux approches similaires, l'une utilisant des intervalles de confiances dérivés de l'inégalité de Hoeffding, l'autre utilisant les intervalles de crédibilité Bayésiens. Nous pouvons voir qu'étonnamment, les performances pour Hoeffding sont très faibles. Notre interprétation est que les bornes de Hoeffding ne sont pas très étroites en faible nombre d'échantillons, et donc l'algorithme a tendance à attendre trop longtemps avant de découper une zone, l'alternative étant réduire la taille des ces intervalles en réglant le paramètre δ, ce

qui engendre une mauvaise sélection de la découpe. Ici, le problème auquel est confronté cette méthode est que le budget d'annotation est très faible, ce qui n'était pas le cas dans les tests effectués dans la publication d'origine. Nous pouvons voir que l'utilisation des intervalles de crédibilité Bayésiennes donne ici de meilleures performances. En effet, celles-ci sont étroites quel que soit le nombre d'échantillons. De plus, au cours de ces expériences, nous avons relevé que le nombre d'échantillons dans une zone restait toujours très faible, étant découpée avant d'avoir le temps de grossir. Pour le bon fonctionnement de l'algorithme, il est donc impératif que les bornes soient utilisables en faible nombre d'échantillons.

Il peut être remarqué que l'algorithme c4.5 donne de meilleurs résultats en faible budget. Rappelons que cet algorithme est utilisé de manière hors-ligne, l'arbre peut donc être recalculé entièrement à chaque pas de temps. Ainsi, celui-ci n'a pas besoin d'être prudent quant à l'utilité d'une découpe dans le futur. Cette liberté permet de sélectionner à chaque pas de temps la meilleur découpe à l'instant présent. Notez que le paramètre de l'algorithme BOCT peut être réglé pour ne pas inclure d'exploration, ce qui revient à utiliser le critère de c4.5. En effet, la façon dont sont construits les intervalles de crédibilité implique que si le paramètre est fixé à $\delta = \frac{1}{2}$, alors la borne supérieure et inférieure sont confondues et le critère résultant est exactement égal à l'estimation du critère idéal. En prenant cette valeur du paramètre, notre algorithme aurait donc possédé les même performances initiales que c4.5. Cependant, le paramètre est ici choisi pour que les performances finales soient bonnes. Ceci nous amène à penser que l'algorithme pourrait profiter d'un paramètre dépendant du budget d'annotations disponible.

Nous remarquons également que les performances finales de BOCT sont, sur certaines bases de données, meilleures que celles de c4.5. Une explication possible est que ce dernier fait trop confiance à l'estimation et découpe trop finement certaines zones. Ainsi, nous relions ce phénomène à du sur-apprentissage. L'algorithme BOCT lancé de façon hors-ligne avec les mêmes paramètres, donne aussi de meilleures performances finales que c4.5, ceci n'est donc pas lié au chemin suivi par l'algorithme en-ligne.

Finalement, intéressons nous à l'algorithme incluant une stratégie de sélection des zones à échantillonner. Nous pouvons voir que les performances sont toujours supérieures à celui sans stratégie de sélection. Cependant, l'impact sur les performances n'est pas flagrant. Nous pouvons expliquer cela par le fait que le nombre d'observations dans chaque zone reste faible et qu'ainsi, la valeur du paramètre associé à chaque zone reste toujours très incertaine. Dans ce cas, sans différence notable dans la valeur du paramètre, il est difficile de privilégier une zone ou l'autre.

8.7 Conclusion intermédiaire

Les objectifs de ce chapitre étaient doubles. D'une part nous avons développé un nouvel algorithme d'arbre incrémental, qui est un classifieur en ligne partitionnant à chaque instant l'espace d'entrée en plusieurs zones indépendantes, c'est à dire que la prédiction dans chaque zone ne dépend que des observations issues de celle-ci. Les partitions successives ne pouvant pas changer entièrement mais devant suivre des règles, ceci peut être vu comme une partition adaptative de l'espace d'entrée, laquelle part d'une unique zone couvrant l'espace entier et se voit découpée au fur et à mesure de l'avancement de l'échantillonnage. La méthode que nous avons développée utilise des intervalles de crédibilité Bayésiens dans le but de gérer l'incertitude vis-à-vis de l'utilité d'une découpe. Ceci permettait alors d'intégrer une notion de prudence en travaillant avec la borne inférieure de ces intervalles. Nous avons montré à travers des expériences menées sur des bases de données tirées du monde réel que l'utilisation d'intervalles de crédibilité Bayésiens apportait un réel avantage par rapport aux intervalles de confiance utilisés habituellement. Nous avons aussi montré

à travers un problème jouet que l'algorithme ainsi développé revenait à sélectionner à chaque pas de temps la partition possédant les meilleures performances parmi toutes celles possibles et de basculer de l'une à l'autre au meilleur moment. Nous pensons alors que cet algorithme gère le critère de découpe de façon optimale.

D'autre part, nous avons développé un second algorithme utilisant le classifieur ci-dessus dans le contexte de l'apprentissage actif en intégrant une stratégie de sélection de la zone à échantillonner. Cette dernière fait intervenir le principe de l'optimisme face à l'incertitude dans le but de gérer le dilemme entre exploration et exploitation du risque réel estimé. Les résultats ont montré que cette stratégie améliorait les performances par rapport à une stratégie aléatoire.

Chapitre 9

Utilisation d'une partition fixe avec des zones combinées

9.1 Introduction

Dans le chapitre précédent, nous avons étudié une première évolution possible du problème à savoir la relaxation de l'immuabilité de la partition, ce qui nous a amené à considérer une partition adaptative. Nous avons vu qu'ainsi le nombre de zones pouvait s'adapter automatiquement aux données reçues afin de garantir une vitesse d'apprentissage maximale à chaque instant. Cependant, le problème que l'on tentait de résoudre, était alors en constante évolution durant l'avancement de l'échantillonnage et l'effet sur le long terme était difficilement appréhendable. De ce fait, l'algorithme d'apprentissage actif ne pouvait pas définir de stratégie de sélection efficace. En effet, à chaque instant, les possibilités de prédiction données au classifieur n'étaient pas les mêmes. Ne pouvant pas prévoir si une zone allait être découpée ou non, la stratégie de sélection devait se contenter d'échantillonner celles qui se révélaient utiles vis-à-vis de la partition courante. Dans ce chapitre nous proposons une seconde piste d'évolution qui conserve l'aspect immuable de la partition mais relâche la contrainte d'indépendance entre les zones. Nous verrons qu'ainsi, un grand nombre de zones pourra être utilisé sans que la vitesse d'apprentissage ne soit limitée.

Dans le cas d'une partition fixe, le facteur limitant la vitesse d'apprentissage est le fait que les zones soient indépendantes. En effet, dans ce cas les prédictions dans chaque zones ne peuvent se baser que sur les observations obtenues dans leur zone respective. En conséquence, le nombre d'observations nécessaire pour atteindre une même précision dans chaque zone est démultiplié avec leur nombre. Cependant, on ignore totalement les positions relatives de chaque zone. En effet, rappelons qu'une telle partition provient de la découpe d'un espace d'entrée. Ainsi, les zones obtenues peuvent être localisées les unes par rapport aux autres de même qu'une certaine mesure de la distance entre deux zones peut être exprimée. De part la nature physique des données étudiées, la répartition de celles-ci et de leurs étiquettes est généralement continue et ne varie pas trop rapidement. Ainsi, deux zones voisines auront tendance à posséder des valeurs du paramètre proches. De ce fait, une observation dans une zone donnée pourrait fournir non seulement une indication sur la valeur du paramètre dans celle-ci, mais aussi dans les zones voisines. Notez toutefois que les observations provenant de chaque zone n'ont pas toutes la même influence, en particulier moindre que celles provenant de la zone concernée.

Le problème que nous étudions dans ce chapitre se base donc sur l'utilisation d'une partition fixe avec des zones combinées, celui-ci tirant parti des relations existantes entre chaque zone dans l'apprentissage des paramètres. Celle-ci se définit de la même manière que les partitions fixes étu-

diées précédemment en y ajoutant un taux de combinaison entre chaque paire de zones. Notez que cette dernière est définie par avance et ne peut en aucun cas voir sa valeur modifiée au cours de l'exécution de l'algorithme. Ceci fournit donc un cadre immuable permettant ainsi à l'algorithme d'apprentissage actif de se focaliser uniquement sur l'apprentissage des paramètres, sans nécessiter de prendre en compte la manière dont évoluerait le problème. L'avantage apporté par une telle partition par rapport à la partition fixe est que la vitesse d'apprentissage n'est plus limitée par le nombre de zones, puisqu'une annotation donne de l'information à chaque autre zone. Ainsi, la partition utilisée peut être extrêmement détaillée, ce qui est le cas dès le début de l'exécution contrairement à une partition adaptative. Dans cette dernière, tant que la première zone n'a pas été découpée, une seule prédiction peut être effectuée sur l'ensemble de l'espace d'entrée, alors qu'ici dès la deuxième étiquette reçue différentes zones peuvent prendre différentes étiquettes sans compromettre la précision de l'estimation, et ainsi posséder dès lors de meilleures performances globales.

Initialement, nos travaux avaient porté sur l'utilisation d'un jeu de différentes partitions fixes avec des zones indépendantes à la manière des Forêts aléatoires [10] ou plus exactement du Codage par tuiles [72]. Finalement, la façon dont nous l'avons traité nous a conduit à considérer le cas plus général des partitions fixes avec des zones combinées. Nous avons donc préféré présenter ici le problème le plus général des deux. Cependant, dans ce dernier, aucune indication n'est donnée sur la manière d'obtenir les taux de combinaisons. Nous utiliserons donc ce problème initial dans le but de fournir un exemple de la façon dont les taux de combinaisons peuvent être obtenus. De plus, il fournit une interprétation différente de la partition avec des zones combinées que celle avec laquelle nous l'avons introduite. L'idée derrière le Codage par tuiles est d'utiliser simultanément un ensemble de différentes partitions fixes avec des zones indépendantes. Un classifieur est ensuite appris sur chaque partition indépendamment, chacun utilisant les mêmes observations. Les critères de décision de chacun sont ensuite combinés, à la manière d'un comité d'experts, pour obtenir la prédiction finale. Notez que puisque chaque partition peut être vue comme résultant d'un arbre de décision, le classifieur ainsi obtenu peut être rapproché à une Forêt aléatoire, bien que dans ces dernières chaque arbre soit recalculé à chaque pas de temps ce qui n'est pas le cas ici. De cette façon, la prédiction finale peut varier sur une partition plus détaillée que celles incluses initialement dans l'ensemble, celle-ci résultant de l'intersection de ces dernières. Il est donc possible d'utiliser des partitions comportant un faible nombre de zones tout en laissant la possibilité d'une bonne performance maximale. En faisant cela, l'estimation du paramètre dans chaque zone reste précise pour chaque partition individuelle, ce qui en résulte une bonne vitesse d'apprentissage. En raisonnant avec la partition résultant de l'intersection, cela peut être vu comme une partition dans laquelle la prédiction dans une zone dépend des observations dans les autres zones. L'importance de ces dernières étant définie en fonction de la taille et du taux de recouvrement des zones de chaque partition individuelle. Une partition fixe avec des zones combinées peut donc être vue comme la généralisation du cas des partition multiples.

Dans ce chapitre, nous proposons un algorithme d'apprentissage actif fonctionnant sur une telle partition et utilisant le principe de l'optimisme face à l'incertitude. Pour cela, il est d'abord nécessaire de définir une stratégie de sélection en connaissance parfaite, qui a donc accès aux paramètres de la distribution de l'oracle. Nous verrons que celle-ci se base sur l'effet global d'un échantillonnage sur le risque réel, elle doit donc prendre en compte l'effet d'une nouvelle observation sur la prédiction dans les autres zones. Comme dans les chapitres précédents, nous verrons que pour suivre au plus près cette stratégie idéale, l'usage du principe de l'optimisme face à l'incertitude passe par la construction d'intervalles de crédibilité. Ici, les zones n'étant plus indépendantes, l'estimation du paramètre ne suivant plus une loi Binomiale, l'utilisation d'une loi Beta lors de l'in-

férence Bayésienne n'est plus possible. Nous approximerons donc la loi d'où est tirée l'estimateur du paramètre par une loi Normale. Enfin, notez que les résultats présentés ici ont fait l'objet d'une publication [22].

Nous évaluerons finalement les algorithmes introduits dans ce chapitre à l'aide d'expériences utilisant des bases de données tirées du monde réel. Deux versions seront testées, l'une utilisant l'approche par des partitions multiples, et l'autre produisant directement la fonction de combinaison entre les zones calculée à partir d'une fonction de noyau Gaussien à partir des centres des zones. Ces algorithmes seront ensuite comparés à d'autres algorithmes issus de l'état de l'art afin de juger de leur efficacité.

9.2 Une Partition fixe avec des zones combinées

Cette section a pour but de définir les notations relatives à la partition fixe avec des zones combinées ainsi que le classifieur se basant sur cette dernière. Il s'agit en particulier d'exprimer les paramètres auxquels aura accès l'algorithme d'apprentissage actif pour définir la stratégie de sélection des zones à échantillonner. De même que les paramètres intrinsèques de la partition que l'algorithme tentera d'estimer et qui définit la stratégie de sélection en connaissance parfaite.

La partition fixe, c'est à dire uniquement la manière dont l'espace d'entrée est découpé, reste identique à celle utilisée dans le cas où les zones étaient indépendantes. Commençons par rappeler ces notations. Soit l'espace d'entrée X, à celui-ci est attribué une partition N qui le découpe en K zones :

$$N = \{X_1, \ldots, X_K\},$$

telle que les propriétés suivantes soient respectées :

- $\forall i \in [\![1, K]\!] : X_i \neq \emptyset$ (aucune zone n'est vide),
- $\cup_{i=1}^{K} X_i = X$ (les zones couvrent l'ensemble de l'espace d'entrée),
- $\forall (i, j) \in [\![1, K]\!]^2 : i \neq j \implies X_i \cap X_j = \emptyset$ (les zones ne se recouvrent pas entre elles).

Précisons que cette partition est fixe, le découpage de l'espace d'entrée ne peut en aucun cas être modifié ou évoluer au cours de l'exécution de l'algorithme. Ceci correspond à la fois au nombre de zones présentes dans la partition et à leur délimitation. Celui-ci doit être donné comme argument de l'algorithme ou bien être défini par ce dernier avant de débuter le processus d'apprentissage.

Cette partition est caractérisée par plusieurs paramètres. Tout d'abord, comme précédemment, chacune des zones $k \in \{1, ..., K\}$ est associée à des paramètres intrinsèques liés aux différentes distributions caractéristiques du problème :

- μ_k : le paramètre de la loi de Bernoulli représentant l'étiquette renvoyée par l'oracle d'une donnée tirée aléatoirement dans la zone k, rappelons que l'ensemble des étiquettes possible est $Y = \{0, 1\}$,
- w_k : l'importance relative de la zone k définie à partir de la distribution naturelle des données d'entrée comme la probabilité qu'une donnée d'entrée tirée aléatoirement provienne de cette zone.

Ensuite, ce qui fait la spécificité d'une partition fixe avec des zones combinées est qu'à ceux-ci s'ajoutent les paramètres suivants :

- w_{k_1,k_2} : le taux de combinaison définissant l'influence de la zone k_1 sur la zone k_2 qui sera utilisée par le classifieur lors de la prédiction de l'étiquette à attribuer à cette dernière.

Notez que ce taux renvoie à la proportion de chaque zone à prendre en compte dans l'estimation du paramètre, ainsi la somme de ces proportions doit donc être égale à 1. Nous avons donc, $\forall k \in \{1, \dots, K\}$

$$\sum_{k'=1}^{K} w_{k',k} = 1.$$

Ces taux peuvent soit être inhérents au problème ou bien être définis par l'utilisateur. Toutefois, ceux-ci ne doivent pas être modifiés au cours de l'exécution de l'algorithme.

Voyons maintenant comment la prédiction est effectuée par le classifieur ainsi que les données relatives à l'échantillonnage utilisable par l'algorithme d'apprentissage actif. Durant le processus d'apprentissage, une étiquette $y_t \in Y$ est reçue à chaque pas de temps $t \in \{1, \dots, n\}$ en provenance d'une zone k_t. Ces observations permettent d'estimer les paramètres de l'oracle. Soit au temps t, le nombre d'étiquettes reçues dans chaque zone : $\forall k \in \{1, \dots, K\}$,

$$T_{k,t} = \sum_{i=1}^{t} \mathbb{1}_{k=k_t}.$$

La moyenne de ces étiquettes est notée : $\forall k \in \{1, \dots, K\}$,

$$\hat{\mu}_{k,t} = \begin{cases} \frac{1}{T_{k,t}} \sum_{i=1}^{t} \mathbb{1}_{k=k_t} y_t & \text{si } T_{k,t} \neq 0 \\ \frac{1}{2} & \text{si } T_{k,t} = 0. \end{cases}.$$

Le classifieur base sa prédiction sur l'estimation du paramètre dans chaque zone. Ici, afin de prendre en compte les observations reçues dans les zones voisines, un premier estimateur possible du paramètre μ_k de la zone k serait

$$\hat{m}_{k,t}^{tmp} = \sum_{k'=1}^{K} w_{k',k} \hat{\mu}_{k',t}.$$

Notez que cet estimateur est biaisé, en effet son espérance sur les tirages vaut

$$m_k^{tmp} = \mathbb{E}_{\forall i \in \{1,\dots,t\}, y_i \sim \mathcal{B}er(\mu_{k_i})}[\hat{m}_{k,t}^{tmp}] = \sum_{k'=1}^{K} w_{k',k} \mu'_k \neq \mu_k.$$

Cependant, nous supposerons que le paramètre de l'oracle varie assez lentement dans l'espace d'entrée et que la forme de la fonction de combinaison considérée est assez locale (taux de combinaison décroissant avec la distance) pour que ce biais n'entrave pas les performances. Deuxièmement, la perte en performance dû à ce biais est nettement contre-balancée par le gain en vitesse d'apprentissage permis par cet estimateur. De plus, comme nous le verrons dans le paragraphe suivant, l'utilisation d'un tel classifieur est justifiée puisque liée à des algorithmes existants. Nous nous concentrerons ici sur la façon d'exploiter ce classifieur pour en tirer les meilleurs performances.

Une légère modification est apportée à cet estimateur afin que les zones n'ayant pas encore reçu d'étiquette ne soient pas pris en compte. En effet, dans le cas contraire, lorsqu'une seule étiquette est reçue, le maximum de vraisemblance que nous définirons sera très proche de $\frac{1}{2}$ alors qu'il est plus cohérent qu'il soit égal à la valeur de l'étiquette reçue. Ainsi, nous utiliserons plutôt l'estimateur suivant :

$$\hat{m}_{k,t} = \frac{\sum_{k'=1}^{K} w_{k',k} \hat{\mu}_{k',t} - \frac{1}{2}}{\sum_{k'=1}^{K} w_{k',k} \mathbb{1}_{T_{k',t}>0}} + \frac{1}{2}.$$

Notez que cette formule est due au fait que $\hat{\mu}_{k,t} = \frac{1}{2}$ si $T_{k,t} = 0$. De plus, remarquez que dans le contexte de la classification, cela ne change pas la prédiction. En effet, $\hat{m}_{k,t} \geq \frac{1}{2} \iff \hat{m}^{tmp}_{k,t} \geq \frac{1}{2}$.

Finalement, le classifieur qui sera utilisé dans ce chapitre est celui qui prédit pour la zone k l'étiquette

$$f_k = \lfloor \hat{m}_{k,t} \rceil,$$

avec $\lfloor . \rceil$ l'opérateur d'arrondi.

9.2.1 Un exemple : les partitions multiples

Dans cette sous-section, nous étudions un autre type de classifieur issu de l'utilisation de multiples partitions fixes avec des zones indépendantes combinées à la manière d'un comité d'expert. Nous verrons qu'en effectuant de légères modifications ce dernier peut être assimilé à une partition fixe avec des zones combinées. Ceci permet d'une part de justifier l'étude de ce type de partition car reprenant un classifieur existant. D'autre part, cela permet de donner un exemple sur la façon de définir le taux de combinaison à utiliser avec ce type de partition. Celui-ci aurait pu être défini arbitrairement par l'utilisateur, à partir, par exemple de la distance entre les différentes zones, comme dans le cas de l'utilisation d'une fonction de noyau Gaussien. Cependant, il est ici directement lié à la configuration du problème, notamment à la distribution naturelle des données d'entrée, ce qui le rend selon nous plus adéquat.

Nous nous éloignons tout d'abord brièvement du cadre mis en place dans la partie précédente pour aborder le problème sous un autre angle. Pour pouvoir augmenter la vitesse d'apprentissage, l'idée est ici d'utiliser simultanément plusieurs partitions fixes avec des zones indépendantes. Chaque partition est apprise indépendamment, en calculant la valeur de l'estimateur du paramètre pour chaque zone. En limitant le nombre de zones dans chaque partition, ceci permet de garder une bonne précision sur l'ensemble des estimations effectuées. Celles-ci seront ensuite combinées afin d'effectuer la décision sur l'étiquette à attribuer à chaque zone. Cela permet à la prédiction finale de varier plus rapidement sur l'espace d'entrée et donc de permettre une bonne performance finale.

La relation avec un classifieur existant provient du fait que cette approche partage des similarité avec les forêts aléatoires. Ce classifieur consiste à apprendre indépendamment un ensemble d'arbres de classification qui comportent un facteur aléatoire, puis à combiner les prédictions effectuées par chacun de ces arbres dans la décision finale. Notez que chaque arbre peut être vu comme définissant une partition de l'espace d'entrée. Toutefois, dans l'utilisation classique de cette dernière, la construction de l'arbre, et donc la délimitation des zones de chaque partition, dépend des étiquettes reçues à l'instant présent, et sont reconstruit à chaque pas de temps. Dans notre problème, il est important que les partitions utilisées ne changent pas au cours de l'exécution, celles-ci doivent donc être définies à l'avance. Cependant, une autre méthode en adéquation avec ce dernier point est utilisée dans le domaine de l'apprentissage par renforcement, il s'agit du codage par tuile (tile coding).

Tachons maintenant de définir plus formellement l'utilisation de partitions multiples. Soit un ensemble de p partitions

$$\mathbf{N} = \{N_1, \ldots, N_p\},$$

avec $\forall N_j \in \mathbf{N}$,

$$N_j = \{x_{1,j}, \ldots, X_{K_j,j}\},$$

avec K_j le nombre de zones de la partition N_j. Chacune de ces partitions respecte les propriétés listées dans la partie précédente.

Chaque partition est apprise indépendamment avec le même jeu de données étiquetées. Soit $T_{j,k,t}$ le nombre d'étiquettes reçues dans la zone k de la partition j au temps t :

$$T_{j,k,t} = \sum_{i=1}^{t} \mathbb{1}_{x_i \in X_{k,j}},$$

et $\hat{\mu}_{j,k,t}$ leur moyenne :

$$\hat{\mu}_{j,k,t} = \begin{cases} \frac{1}{T_{j,k,t}} \sum_{i=1}^{t} \mathbb{1}_{x_i \in X_{k,j}} y_i & \text{si } T_{j,k,t} \neq 0 \\ \frac{1}{2} & \text{si } T_{j,k,t} = 0. \end{cases}$$

Voyons maintenant comment définir la prédiction finale à partir des estimateurs de chaque partition individuelle. Notez que deux données d'entrée provenant de la même zone quelle que soit la partition individuelle considérée subissent nécessairement la même prédiction. Pour simplifier, nous définissons donc la partition la plus détaillée comme celle regroupant dans une même zone toutes les données partageant la même prédiction finale. Elle résulte donc de l'intersection de toutes les partitions individuelles définies précédemment. On la note

$$\mathcal{N} = \{\mathcal{X}_1, \ldots, \mathcal{X}_\mathcal{K}\},$$

et elle est telle que $\forall c \in \{1, \ldots, \mathcal{K}\}, \forall (x_a, x_b) \in \mathcal{X}_c^2, \forall j \in \{1, \ldots, p\}$,

$$\exists k \in \{1, K_j\}, x_a \in X_{k,j} \iff x_b \in X_{k,j}.$$

Comme dans le cas d'une partition fixe, la distribution de l'oracle est représentée dans chacune de ces zones c par le paramètre d'une loi de Bernoulli μ_c et la distribution naturelle par son importance relative w_c. La Figure 9.1 montre un exemple simple de jeu de partitions individuelles et de la partition la plus détaillée associée.

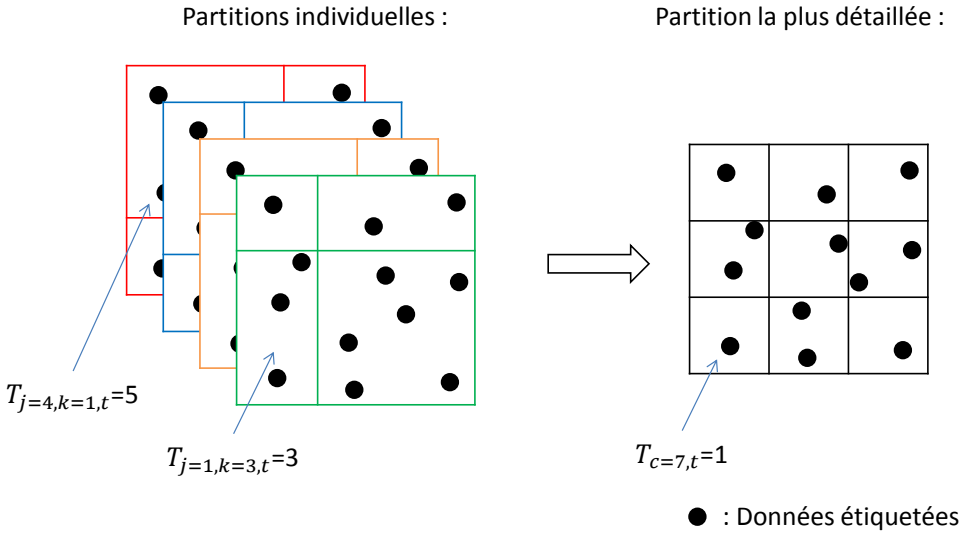


FIGURE 9.1 – Jeu de partitions et la partition la plus détaillée associée

Ainsi, la prédiction $f_{c,t}$ associée à la zone c de la partition la plus détaillée provient de la moyenne de l'estimateur associé à la zone de chaque partition incluant la zone c. Soit $\hat{m}_{c,t}$ cette moyenne,

$$\hat{m}_{c,t} = \frac{1}{p} \sum_{j=1}^{p} \hat{\mu}_{j,k_{j,c},t}, \tag{9.1}$$

avec $k_{j,c} \in \{1, \ldots, K_j\}$ tel que $\mathcal{X}_c \subset X_{k_j}$. La prédiction effectuée est alors

$$f_{c,t} = \lfloor \frac{1}{p} \sum_{j=1}^{p} \hat{\mu}_{j,k_{j,c},t} \rceil, \tag{9.2}$$

avec $\lfloor . \rceil$ l'opérateur d'arrondi.

Remarquez qu'il est aussi possible de compter le nombre d'étiquettes reçues dans chaque zone de la partition la plus détaillée

$$T_{c,t} = \sum_{i=1}^{t} \mathbb{1}_{x_i \in \mathcal{X}_c},$$

et leur moyenne

$$\hat{\mu}_{c,t} = \begin{cases} \frac{1}{T_{c,t}} \sum_{i=1}^{t} \mathbb{1}_{x_i \in \mathcal{X}_c} y_i & \text{si } T_{c,t} \neq 0 \\ \frac{1}{2} & \text{si } T_{c,t} = 0. \end{cases}$$

qui agit comme un estimateur de μ_c. Nous allons voir que la prédiction finale peut aussi être calculée directement à partir de ces valeurs.

Étant donné qu'aucune des zones de la partition la plus détaillée n'est à cheval sur deux partitions de n'importe quelle partition individuelle, et qu'elles ne s'intersectent pas, alors une zone d'une partition individuelle est strictement égale à l'union d'un certain nombre de zones de la partition la plus détaillée. Ainsi, l'estimateur associé à une zone d'une partition individuelle peut être vu comme la combinaison linéaire des estimateurs des paramètres de la partition la plus détaillée. Soit

$$\hat{\mu}_{j,k,t} = \frac{1}{\sum_{c \text{ t.q. } \mathcal{X}_c \subset X_{k,j}} T_{c,t}} \sum_{c \text{ t.q. } \mathcal{X}_c \subset X_{k,j}} T_{c,t} \hat{\mu}_{c,t}. \tag{9.3}$$

Finalement, en combinant les équations 9.1 et 9.3, nous obtenons

$$\hat{m}_{c,t} = \frac{1}{p} \sum_{j=1}^{p} \frac{1}{\sum_{c' \text{ t.q. } \mathcal{X}'_c \subset X_{k,j}} T_{c',t}} \sum_{c' \text{ t.q. } \mathcal{X}'_c \subset X_{k,j}} T_{c',t} \hat{\mu}_{c',t}.$$

Ceci peut être vu comme une combinaison linéaire des estimateurs des paramètres de l'oracle dans chaque zone de la partition la plus détaillée. Cependant, les poids de cette combinaison linéaire dépendent du nombre d'observations reçues et changent donc à chaque pas de temps.

Toutefois, une autre remarque permet d'apporter une solution à ce problème. En effet, cette définition du classifieur suppose que les données d'entrée étiquetées aient été tirées aléatoirement selon la distribution naturelle. En effet, dans le cas contraire, l'équation 9.3 n'est plus vérifiée. Or, dans le cas de l'apprentissage actif, la distribution des données étiquetées se trouve modifiée. La solution est alors d'introduire une correction similaire à celle observée en échantillonnage d'importance, ce qui revient à remplacer dans l'équation du classifieur le nombre de données étiquetées par l'importance relative de chaque zone. Ainsi,

$$\hat{m}_{c,t} = \frac{1}{p} \sum_{j=1}^{p} \frac{1}{\sum_{c' \text{ t.q. } \mathcal{X}'_c \subset X_{k,j}} w_{c'}} \sum_{c' \text{ t.q. } \mathcal{X}'_c \subset X_{k,j}} w_{c'} \hat{\mu}_{c',t}.$$

Notez que ceci est équivalent à

$$\hat{m}_{c,t} = \sum_{c'=1}^{\mathcal{K}} w_{c',c} \hat{\mu}_{c',t},$$

avec

$$w_{c',c} = \frac{1}{p} \sum_{j=1}^{p} \mathbb{1}_{\mathcal{X}_{c'} \subset X_{k_{j,c},j}} \frac{w_{c'}}{\sum_{c'' \text{ t.q. } \mathcal{X}''_c \subset X_{k,j}} w_{c''}}.$$

Nous obtenons donc une expression du classifieur issu l'utilisation de multiple partitions sous la forme d'une combinaison des zones de la partition la plus détaillée. Il est alors possible d'ignorer totalement les partitions individuelles pour ne travailler qu'avec cette dernière. Ceci revient alors à employer une partition avec des zones combinées comme décrit au début de la section. Celle-ci pouvant être vue comme une généralisation de ce problème avec des taux de combinaison quelconques.

9.3 Le critère de sélection en connaissance parfaite

Dans la partie précédente, nous avons pu exprimer le classifieur qui est étudié dans ce chapitre. Le but est maintenant de voir comment construire un algorithme d'apprentissage actif qui utilise ce classifieur en lui fournissant les données qui lui permettent d'effectuer rapidement une bonne prédiction. Cela consiste ici à sélectionner la zone de la partition dans laquelle piocher un exemple aléatoirement. Le but est alors de définir un critère associé à chaque zone qui permettra de choisir la meilleure zone en fonction des données acquises dans les étapes antérieures. Notre volonté est ici encore d'utiliser le principe de l'optimisme face à l'incertitude pour évaluer l'intérêt relatif au choix d'une zone. Cette approche passe dans un premier temps par la définition d'un critère en connaissance parfaite, basé sur les paramètres inconnus du problème, avant de considérer un incertitude sur ce critère liée à l'estimation de ces paramètres, puis de la gérer de façon optimiste. Dans cette section, nous nous concentrons donc sur l'expression de ce critère en connaissance parfaite.

La différence de ce classifieur par rapport à ceux étudiés dans les chapitres précédents est que l'échantillonnage d'une étiquette dans une zone modifie/améliore la prédiction dans les autres zones. Afin rendre compte de l'intérêt de sélectionner une zone, il est alors nécessaire d'évaluer le gain en performance engendré dans chacune de ces zones. La stratégie envisagée ici est d'effectuer la minimisation myope du risque réel. Ainsi, l'échantillonnage de chaque zone est simulé et le risque réel résultant est enregistré, puis la zone menant à la plus forte décroissance du risque réel est effectivement échantillonnée.

Rappelons l'expression du risque réel qui est

$$R_t(f) = \sum_{k=1}^{K} w_k R_{k,t}(f_k),$$

avec

$$R_{k,t}(f_k) = \mathbb{1}_{f_k=0} \mu_k + \mathbb{1}_{f_k=1}(1 - \mu_k)$$

la part du risque réel associée à la zone k. Notez que le risque optimal est atteint pour $f_k = [\mu_k]$ avec $[.]$ l'opérateur d'arrondi et que le risque réel de la zone k peut aussi s'écrire

$$R_{k,t}(f_k) = |\mu_k - \frac{1}{2}| \mathbb{1}_{f_k=[\mu_k]} + R_{k,t}([\mu_k]),$$

avec |.| la valeur absolue. Soit

$$\mathbf{T}_t = (T_{1,t}, \ldots, T_{K,t})^t$$

le vecteur contenant les nombres d'observations dans chaque zone au temps$t$ et

$$\hat{\boldsymbol{\mu}}_t = (\hat{\mu}_{1,t}, \ldots, \hat{\mu}_{K,t})^t$$

le vecteur contenant les moyennes des étiquettes reçues dans chaque zone au temps t. Soit

$$\mathbf{W}_k = (w_{1,k}, \ldots, w_{K,k})^t$$

le vecteur contenant les taux de combinaisons reflétant l'influence de chaque zone sur la zone k. Alors,

$$R_{k,t}(\mathbf{T}_t, \hat{\boldsymbol{\mu}}_t, \mu_k) = |\mu_k - \frac{1}{2}|\mathbb{1}_{\mathbf{W}_k^t \hat{\boldsymbol{\mu}}_t = [\mu_k]} + R_{k,t}([\mu_k]).$$

Tachons maintenant d'évaluer le risque réel engendré par l'échantillonnage dans une zone k'. Notons

$$\mathbf{T}_t^{k'} = (T_{1,t}, \ldots, T_{k'-1,t}, T_{k',t} + 1, T_{k'+1,t}, \ldots, T_{K,t})^t$$

le nombre d'étiquettes dans chaque zone résultant de cet échantillonnage, et

$$\hat{\boldsymbol{\mu}}_t^{k'+} = (\hat{\mu}_{1,t}, \ldots, \hat{\mu}_{k'-1,t}, \frac{T_{1,t}\hat{\mu}_{k',t} + 1}{T_{k',t} + 1}, \hat{\mu}_{k'+1,t}, \ldots, \hat{\mu}_{K,t})^t$$

et

$$\hat{\boldsymbol{\mu}}_t^{k'-} = (\hat{\mu}_{1,t}, \ldots, \hat{\mu}_{k'-1,t}, \frac{T_{1,t}\hat{\mu}_{k',t}}{T_{k',t} + 1}, \hat{\mu}_{k'+1,t}, \ldots, \hat{\mu}_{K,t})^t$$

les moyennes résultant de cet échantillonnage en fonction de la valeur de l'étiquette reçue. Alors la décroissance du risque réel pour la zone k obtenu à la suite de cet échantillonnage est

$$\Delta_{k'} R_{k,t}(\mathbf{T}_t, \hat{\boldsymbol{\mu}}_t, \mu_k, y_{k'}) = \mathbb{1}_{y_{k'}=0} R_{k,t}(\mathbf{T}_t^{k'}, \hat{\boldsymbol{\mu}}_t^{k'-}, \mu_k) + \mathbb{1}_{y_{k'}=1} R_{k,t}(\mathbf{T}_t^{k'}, \hat{\boldsymbol{\mu}}_t^{k'+}, \mu_k) - R_{k,t}(\mathbf{T}_t, \hat{\boldsymbol{\mu}}_t, \mu_k).$$

Ainsi, la décroissance du risque global engendré par l'échantillonnage dans la zone k' est

$$\Delta_{k'} R_t(\mathbf{T}_t, \hat{\boldsymbol{\mu}}_t, \boldsymbol{\mu}, y_{k'}) = \sum_{k=1}^{K} \Delta_{k'} R_{k,t}(\mathbf{T}_t, \hat{\boldsymbol{\mu}}_t, \mu_k, y_{k'}),$$

avec $\boldsymbol{\mu}$ le vecteur contenant les paramètres de l'oracle associé à chaque zone.

La stratégie de sélection en connaissance parfaite est alors celle qui maximise la décroissance du risque réel global. C'est à dire

$$k_{t+1} = \underset{k' \in \{1,\ldots,K\}}{\arg\min} \; \Delta_{k'} R_t(\mathbf{T}_t, \hat{\boldsymbol{\mu}}_t, \boldsymbol{\mu}, y_{k'}).$$

Ici les valeur de $\boldsymbol{\mu}$ et de $y_{k'}$ sont inconnues. L'objet du prochain chapitre est de prendre en compte la connaissance que l'on a de ces paramètres et de définir une stratégie de sélection qui approche cette dernière tout en permettant d'acquérir de la connaissance en parallèle.

9.4 Prise en compte de l'incertitude

Nous sommes désormais en possession d'un critère de sélection des zones à échantillonner en connaissance parfaite. En poursuivant la démarche détaillée au début de cette partie, nous allons maintenant utiliser le principe de l'optimisme face à l'incertitude pour définir un critère de sélection qui tente d'approcher ce critère idéal tout en ayant accès qu'à des observations sur les vrais paramètres. Pour cela, nous devrons d'abord établir des intervalle de confiance autours du critère idéal pour chaque zone. Celle qui possède la borne supérieure la plus forte sera ensuite sélectionnée pour échantillonnage. Comme dans les chapitres précédents, des intervalles de confiance Bayésiens seront utilisés étant donné la certitude sur la famille de distributions de l'oracle. Il s'agira alors de calculer une distribution *a posteriori* sur les paramètres pour pouvoir la transférer au critère idéal pour ensuite couper les queues de la distribution résultante, obtenant ainsi les intervalles recherchés.

Dans les chapitres précédents, la distribution *a posteriori* du critère se déduisait directement de celle des paramètres. Ici, le critère n'est plus fonction d'un seul paramètre mais de plusieurs. Ainsi, bien que la distribution *a posteriori* puisse être calculée exactement, cette dernière ne possède pas de forme fermée et son calcul numérique est extrêmement coûteux. Deux approches seront proposées pour permettre de calculer un intervalle de crédibilité sur le critère. La première se base sur l'estimation de la distribution *a posteriori* par Monte Carlo. La deuxième se ramènera à une hypothèse non-bruitée, dans laquelle le paramètre ne peut prendre que deux valeurs discrètes, pour pouvoir calculer la distribution *a posteriori* de façon combinatoire.

D'autre part, les paramètres de l'oracle dont dépend le critère seront estimés de la même manière que lors de la prédiction effectuée par le classifieur. En effet, en apprentissage actif deux entités distinctes sont considérées : le classifieur et l'algorithme d'apprentissage actif, il est possible que chacun utilise sa propre estimation des paramètres de l'oracle (l'algorithme d'apprentissage actif ne pouvant voir le classifieur que comme une boite noire renvoyant une prédiction). Cependant, dans la plupart des cas, le même estimateur est utilisé. Ici, la forme de la distribution *a posteriori* sur les paramètres n'est pas connue, mais étant associée à une somme de variables aléatoires, elle sera approximée par une Gaussienne.

L'algorithme d'apprentissage actif pour la classification proposé dans ce chapitre, utilisant le principe d'optimisme face à l'incertitude sur des partitions fixe à zones combinés, est décrit sur l'Algorithme 8. Il prend en argument un paramètre δ qui correspond à la probabilité désirée que le vrai critère soit supérieur à la borne considéré. Ce critère est calculé en suivant l'une des deux méthodes présentées dans la suite de cette section.

Données : N, $(w_k)_{k\in\{1,\dots,K\}}$, $(w_{k_1,k_2})_{(k_1,k_2)\in\{1,\dots,K\}^2}$, δ
Initialisation : $\forall k \in \{1,\dots,K\}$, $\hat{\mu}_{k,0} = \frac{1}{2}$, $T_{k,0} = 0$
pour *t = 1, ..., n* **faire**
  Calculer $C_{k,t}$ en utilisant la **Méthode 1** ou la **Méthode 2**
  Échantillonner la zone $k_t = \arg\max_k C_{k,t}(\delta)$
  Recevoir $y_t \sim \mathcal{B}er(\mu_{k_t})$
  Mettre à jour $\hat{\mu}_{k_t,t}$ et $T_{k_t,t}$
fin
Résultat : $[\hat{\mu}_{k,n}]$ pour chaque zone $k \in \{1,\dots,K\}$

Algorithme 8 : Algorithme optimiste d'apprentissage actif pour la classification sur une partition fixe à zones combinées

Commençons par voir comment attribuer une distribution *a posteriori* aux paramètres de l'oracle

dans chaque zone. Rappelons l'expression de l'estimateur du paramètre associé à la zone k :

$$\hat{m}_{k,t} = \frac{\sum_{k'=1}^{K} w_{k',k}\hat{\mu}_{k',t} - \frac{1}{2}}{\sum_{k'=1}^{K} w_{k',k}\mathbb{1}_{T_{k',t}>0}} + \frac{1}{2}.$$

Cependant, nous souhaitons obtenir une distribution *a posteriori* sur la valeur du paramètre. A la place, nous définirons une distribution *a posteriori* sur l'espérance de cet estimateur par

$$m_k = \frac{\sum_{k'=1}^{K} w_{k',k}\mu_{k'} - \frac{1}{2}}{\sum_{k'=1}^{K} w_{k',k}\mathbb{1}_{T_{k',t}>0}} + \frac{1}{2}$$

que nous utiliserons en lieu et place de celle sur le paramètre de la zone k.

Nous voyons ici que m_k est une combinaison linéaire des $(\mu_{k'})_{k'\in\{1,\dots,K\}}$. Chacun de ces paramètres possède une distribution *a posteriori* pouvant être calculée indépendamment et s'exprimant pour la zone k comme

$$\mu_k \sim \mathcal{B}eta(T_{k,t}\hat{\mu}_{k,t} + 1, T_{k,t}(1 - \hat{\mu}_{k,t}) + 1).$$

Ainsi, si le nombre de zones K est assez grand, une bonne approximation sur la forme de la distribution *a posteriori* du m_k est une Gaussienne. Pour calculer les paramètres de celle-ci, il suffit d'exprimer sa moyenne et sa variance. Sa moyenne résulte de la combinaison linéaire des moyennes de chaque distribution, ce qui revient à $\hat{m}_{k,t}$. Sa variance est calculée comme suit :

$$var_t(m_k) = \sum_{k'=1}^{K} \left(\frac{w_{k',k}}{\sum_{k''=1}^{K} w_{k'',k}\mathbb{1}_{T_{k'',t}>0}}\right)^2 var_t(\mu_k),$$

avec $var_t(\mu_k) = \frac{T_{k,t}^2\hat{\mu}_{k,t}(1-\hat{\mu}_{k,t})+T_{k,t}+1}{(T_{k,t}+2)^2(T_{k,t}+3)}$ la variance de la loi Beta de paramètres $T_{k,t}\hat{\mu}_{k,t} + 1$ et $T_{k,t}(1 - \hat{\mu}_{k,t}) + 1$. Alors, la distribution *a posteriori* de l'espérance de l'estimateur pour la zone k s'exprime comme :

$$m_k \sim \mathcal{N}(\hat{m}_{k,t}, var_t(m_k)).$$

Dans la suite, nous supposerons que le paramètre μ_k peut être tiré selon cette distribution. Bien sûr, cette supposition n'est pas exacte car m_k et μ_k ont des valeurs différentes, mais sous une hypothèse de régularité du paramètre et d'un taux de combinaison local, ces deux valeurs sont assez proches pour être assimilées. Deuxièmement, ceci améliore la vitesse d'apprentissage de cette distribution *a posteriori*, ce qui lui donne un avantage permettant de contrebalancer son inexactitude.

Désormais, nous connaissons, un critère de sélection dépendant des paramètres de l'oracle ainsi qu'une distribution *a posteriori* sur ces paramètres. Il s'agit maintenant de pouvoir calculer une distribution *a posterori* sur le critère afin d'en tirer la borne supérieure qui nous servira de critère de sélection. Nous proposons pour cela deux méthodes.

Méthode 1 : Estimation par Monte Carlo Cette méthode est à cheval entre deux méthodes déjà existantes. La première est l'échantillonnage de Thompson qui consiste, lors d'une approche Bayésienne à échantillonner les paramètres selon la distribution *a posteriori*, puis à calculer le critère en fonction des paramètres ainsi échantillonnés. La deuxième celle utilisant l'optimisme face à l'incertitude, qui passe par la construction d'un intervalle de confiance contenant la paramètre avec une forte probabilité puis à utiliser comme critère de sélection la borne supérieure de cet intervalle. Ici, l'idée est d'échantillonner plusieurs fois les distributions *a posteriori*, de calculer le

critère correspondant pour chaque échantillonnage, puis d'ordonner les critères ainsi obtenus, pour finalement utiliser le quantile correspondant à une probabilité définie préalablement.

Soit n_{its} le nombre d'échantillons des distributions *a posteriori* utilisé à chaque étape. Alors, $\forall k \in \{1, \dots, K\}, \forall it \in \{1, \dots, n_{its}\}$,

$$\mu_{it,k,t} \sim \mathcal{N}(\hat{m}_{k,t}, var_t(m_k)).$$

et $\forall k \in \{1, \dots, K\}$,

$$y_{it,k,t} \sim \mathcal{B}in(\mu_{it,k,t}).$$

Soit

$$C_{it,k,t} = -\Delta_k R_t(\mathbf{T}_t, \hat{\boldsymbol{\mu}}_t, (\mu_{it,k',t})_{k' \in \{1,\dots,K\}}, y_{it,k,t}).$$

Soit $\delta \in [0, 1]$ une probabilité fixée. Alors, soit $C_{k,t}$ tel que

$$\sum_{it=1}^{n_{its}} \mathbb{1}_{C_{it,k,t} \geq C_{k,t}} \leq n_{its}\delta.$$

Méthode 2 : Calcul combinatoire Cette méthode consiste à ne donner que deux valeurs possible pour les paramètres de l'oracle dans chaque zone. Dans ce cas, le nombre de valeurs que peut prendre le critère est discret de sorte que la probabilité de chaque cas puisse être calculée individuellement. Le nombre de cas possibles croit donc exponentiellement avec le nombre de zones et son énumération peut alors être difficile, toutefois, nous verrons que celui-ci peut être limité et son utilisation possible en temps raisonnable. L'hypothèse effectuée ici va à l'encontre de la nature bruitée du problème qui nous intéresse. Cependant, dans le cas où le nombre de zones est très grand, et ne contient qu'une seule donnée encore non étiquetée, cela revient à associer une probabilité à son étiquette et à tenter de minimiser le risque empirique du classifieur. Habituellement, la minimisation du risque empirique est évitée car elle engendre un potentiel sur-apprentissage. Cependant, rappelons que nous nous intéressons ici uniquement à la sélection des données constituant la base d'apprentissage. Le classifieur quant à lui est supposé parvenir à généraliser. Ainsi fournir les données telles que la prédiction sur au moins l'ensemble du réservoir soit juste devrait engendrer une bonne performance globale.

Dans la suite, nous associerons la probabilité que le paramètre μ_k prenne la valeur 0 à la probabilité tirée de la distribution *a posteriori* précédente que μ_k soit inférieur à $\frac{1}{2}$. Soit nous notons

$$P_{0,k} = \mathbb{P}(\mu_k = 0 | \hat{\boldsymbol{\mu}}_t) = \Phi(\frac{1}{2}, \hat{m}_{k,t}, var_t(m_k)).$$

Ainsi, soit la simulation d'un échantillonnage dans la zone k. La donnée reçue à la suite de cet échantillonnage est

$$y_k = \begin{cases} 0 & \text{avec une probabilité } P_{0,k} \\ 1 & \text{avec une probabilité } 1 - P_{0,k}. \end{cases}$$

Pour chacun des cas possible, la distribution *a posteriori* sur les paramètres est recalculée en considérant la nouvelle observation. Soit $\forall k' \in \{1, \dots, K\}$,

$$\hat{\mu}_{k',t}^{k,y_k} = \begin{cases} \hat{\mu}_{k',t} & \text{si } k' \neq k \\ \frac{T_{k',t}\hat{\mu}_{k',t} + y_k}{T_{k',t}+1} & \text{si } k' = k \end{cases}$$

et

$$T^k_{k',t} = \begin{cases} T_{k',t} & \text{si } k' \neq k \\ T_{k',t} + 1 & \text{si } k' = k. \end{cases}$$

Soit l'estimateur du paramètre qui en découle

$$\hat{m}^{k,y_k}_{k',t} = \frac{\sum_{k''=1}^{K} w_{k'',k'} \hat{\mu}^{k,y_k}_{k'',t} - \frac{1}{2}}{\sum_{k''=1}^{K} w_{k'',k'} \mathbb{1}_{T_{k'',t}>0}} + \frac{1}{2},$$

ainsi que la variance de la distribution *a posteriori* du paramètre

$$var^{k,y_k}_t(m_{k'}) = \sum_{k''=1}^{K} (\frac{w_{k'',k'}}{\sum_{k^{(3)}=1}^{K} w_{k^{(3)},k'} \mathbb{1}_{T_{k^{(3)},t}>0}})^2 var^{k,y_k}_t(\mu_{k'}).$$

Ce qui donne la probabilité que le paramètre de la zone k' soit égal à 0 suivante :

$$P^{k,y_k}_{0,k'} = \Phi(\frac{1}{2}, \hat{m}^{k,y_k}_{k',t}, var^{k,y_k}_t(m_{k'})).$$

La décroissance du risque réel dans la zone k' engendré par cet échantillonnage dans la zone k est égale à

$$\Delta_{k,y_k} R_{k',t} = \mathbb{1}_{[\hat{m}^{k,y_k}_{k',t}] \neq [\mu_{k'}]} - \mathbb{1}_{[\hat{m}_{k',t}] \neq [\mu_{k'}]}.$$

Notez que dans le cas où la prédiction du classifieur ne change pas à la suite de l'échantillonnage, alors la décroissance du risque est nulle quelle que soit la valeur du paramètre. Ainsi la décroissance du risque est égale à

$$\Delta_{k,y_k} R_{k',t} = \begin{cases} 0 & \text{avec une probabilité} \quad 1 & \text{si } [\hat{m}^{k,y_k}_{k',t}] = [\hat{m}_{k',t}] \\ -1 & \text{avec une probabilité} \quad P^{k,y_k}_{0,k'} & \text{si } [\hat{m}^{k,y_k}_{k',t}] \neq [\hat{m}_{k',t}] \\ 1 & \text{avec une probabilité} \quad 1 - P^{k,y_k}_{0,k'} & \text{si } [\hat{m}^{k,y_k}_{k',t}] \neq [\hat{m}_{k',t}]. \end{cases}$$

Notez qu'après que seulement quelques annotations aient été effectuées, la prédiction du classifieur ne change que localement, ce qui limite le nombre de zones affectées par le changement, et donc le nombre de cas à considérer.

L'ensemble des cas associés à chaque zone avec leur probabilité respective sont ensuite combinés pour en déduire la probabilité de la décroissance du risque global :

$$\mathbb{P}(\Delta_{k,y_k} R_t = Q) = \sum_{\sum_{k'=1}^{K} q_{k'} = Q} \prod_{k'=1}^{K} \mathbb{P}(\Delta_{k,y_k} R_{k',t} = q_{k'}).$$

Puis

$$\mathbb{P}(\Delta_k R_t = Q) = \sum_{y_k=0}^{1} P_{y_k,k} \mathbb{P}(\Delta_{k,y_k} R_t = Q).$$

Une borne supérieure de crédibilité $C_{k,t}$ est ensuite calculée en fonction d'une probabilité fixée $\delta \in [0, 1]$ de la manière suivante :

$$C_{k,t} \text{ telle que } \mathbb{P}(-\Delta_k R_t \geq C_{k,t}) \leq \delta.$$

L'algorithme consiste alors à chaque pas de temps à sélectionner la zone pour laquelle le critère $C_{k,t}$ est maximal. Ce dernier étant calculé en suivant la **Méthode 1** ou la **Méthode 2**.

9.5 Expériences

Dans cette section, nous établissons les expériences qui nous permettront d'évaluer l'algorithme introduit dans ce chapitre. Afin de correspondre au problème que celui-ci tente de résoudre, les expériences se présentent sous la forme d'une partition fixe à laquelle est associé un taux de combinaison entre chaque zone. Cette partition étant telle que définie dans la section 9.2, chacune des zones de celle-ci est dotée de paramètres intrinsèques représentant l'oracle ainsi que la distribution naturelle des données d'entrée. Ici, ces paramètres seront déterminés de sorte qu'ils correspondent à un problème tiré du monde réel. En effet, nous avons déjà vu qu'une partition fixe avec des zones indépendantes était difficilement exploitable sur un tel problème. Ceci étant principalement dû au fait que, pour obtenir une bonne vitesse d'apprentissage, le nombre de zones devait être restreint, ce qui a pour conséquence de devoir travailler avec de grandes zones et rend la performance optimale fortement dépendante de la forme de la partition. Or, définir une bonne partition n'est pas trivial, surtout lorsque aucune observation n'est disponible initialement. Ces expériences serviront donc en partie à montrer que le cadre de travail étudié dans ce chapitre convient à un problème tiré du monde réel, notamment lorsque l'espace d'entrée est continu et en grande dimensions.

Les problèmes considérés dans ces expériences sont tous tirés de bases de données provenant du UCI Machine Learning Repository [**?**]. Celles-ci se composent de données d'entrée appartenant à un espace de type $\mathbb{R}^d$ avec d le nombre de dimensions et des étiquettes correspondantes appartenant à $\{0, 1\}$. Avant le traitement de chaque base de données par un algorithme, et avant chaque exécution sur la même base, cette dernière est mélangée et est divisée en deux parties égales. L'une, appelée réservoir, contient les données disponibles à l'algorithme d'apprentissage actif dont les étiquettes sont initialement cachées mais peuvent être acquises l'une après l'autre. L'autre, appelée base de test, est réservée à l'évaluation des performances des algorithmes en comparant les prédiction effectuées sur ces données avec les vraies étiquettes toujours cachées à l'algorithme. L'algorithme ne peut se servir pour l'apprentissage ni des étiquettes ni de la répartition des données appartenant à cette base de test.

Les données incluses dans le réservoir servent ensuite à la définition des partitions. Dans ces expériences, deux types de partitionnement sont considérés. Le premier suit la démarche décrite en Section 9.2.1, qui passe par l'emploi de multiples partitions fixes avec des zones indépendantes. Ici, l'ensemble des partitions contient 10.000 partitions individuelles. Chaque partition individuelle provient de sept découpes successives de l'espace d'entrée le long d'une des dimensions à la manière d'un arbre de décision. Pour chaque découpe, la dimension selon laquelle elle est effectuée ainsi que le seuil sont choisis aléatoirement de façon uniforme. Notez qu'aucune contrainte n'est appliquée quant à la partition la plus détaillée. De ce fait, le nombre de zones de la partition la plus détaillée peut être extrêmement grand et la taille de chaque zone extrêmement petite. Intuitivement, cela devrait rendre le temps de traitement extrêmement long puisqu'il faut calculer le critère pour chaque zone. Toutefois, de nombreuses zones ne contiennent pas de donnée issue du réservoir. Ainsi, d'une part il est impossible de les sélectionner pour l'échantillonnage, et il est donc inutile de calculer le critère pour celles-ci. D'autre part, l'importance relative de chaque zone est calculée en fonction du nombre de données du réservoir dans celles-ci. De ce fait, ces zones possèderont une importance relative nulle et n'interviendront pas dans la prédiction. En conséquence, toute zone vide peut être simplement ignorée, le nombre de zones restantes étant limité par le nombre de données dans le réservoir. Ainsi, cela revient à ne plus considérer de zones de l'espace d'entrée mais à travailler directement sur les données en y associant des relations entre chacune d'elles. Toutefois, le concept de partition était un passage nécessaire pour définir le taux de combinaison entre chaque zone. Notez que ce cadre de travail se trouve être général puisqu'il reste valable dans les cas où il

serait possible d'échantillonner à plusieurs reprises au même endroit. Finalement, remarquez qu'il est important que la partition soit fixe, et donc que les taux de combinaisons entre les données restent inchangés.

Le deuxième type de partitionnement que l'on utilise dans nos évaluations part du constat que lorsque, comme dans le cas précédent, les zones sont définies pour englober individuellement chaque donnée disponible dans le réservoir, les taux de combinaisons définissant des relations entre chaque peuvent être vus comme la covariance intervenant dans les méthodes à noyau. De plus, l'aspect local de l'influence entre les zones qui ressort de l'approche précédente, avec un taux de combinaison fonction de la distance entres les zones dans l'espace d'entrée, nous incite d'autant plus à faire le rapprochement avec ce type de méthodes. Ainsi, nous avons envisagé de remplacer les taux de combinaisons par une fonction de noyau classique, celle-ci possédant l'avantage de pouvoir être calculée directement. De ce fait, si nous supposons que les zones k_1 et k_2 sont centrées respectivement sur les données d'entrée x_1 et x_2, nous définissons le taux de combinaison entre elles comme

$$w_{k_1,k_2} = \alpha_{k_2} e^{\frac{||x_1 - x_2||^2}{2\sigma^2}},$$

avec α_{k_2} une coefficient tel que $\sum_{k_1=1}^{K} w_{k_1,k_2} = 1$ et σ un paramètre supplémentaire permettant de contrôler l'influence des données voisines.

Au cours de ces expériences, en suivant un procédé classique d'apprentissage actif, les données seront une-à-une sélectionnées par l'algorithme, soumises à l'oracle puis incluses avec l'étiquette renvoyée dans la base d'apprentissage. Cette dernière est initialement vide et l'algorithme effectue sa sélection dès la première données à inclure. A chaque pas de temps, le classifieur est entrainé à partir de la base d'apprentissage courante, puis ses performances sont évaluées en comparant ses prédictions sur les données de la base de test avec les étiquettes correspondantes. Le taux d'erreur de classification est enregistré à chaque pas de temps. Les algorithmes sont exécutés plusieurs fois sur une même base de donnée. Ici le nombre d'exécutions s'élève à 1000. La performance globale à chaque pas de temps est évaluée en effectuant la moyenne des taux d'erreurs sur l'ensemble des exécutions.

Ces expériences ont pour objectif premier de valider l'algorithme d'apprentissage actif introduit dans ce chapitre. C'est à dire de voir si celui-ci arrive à sélectionner des données d'entrée adaptées au classifieur considéré, ce qui lui permet d'améliorer sa vitesse d'apprentissage. Dans un second temps l'intérêt sera de comparer les performance de cet algorithme avec d'autres algorithmes existants, afin de voir que l'utilisation du principe de l'optimisme face à l'incertitude présente effectivement un avantage. Le classifieur que nous avons utilisé dans ce chapitre est une version modifié des forêts aléatoires standards, peu de méthodes d'apprentissage actif ont donc été conçues spécifiquement pour celui-ci. Dans cette section, nous prendrons pour point de comparaison l'algorithme d'échantillonnage d'incertitude, qui est le plus simple et le plus utilisé des algorithmes d'apprentissage actif. Le deuxième point de comparaison sera la version en connaissance parfaite de l'algorithme que nous avons développé. En effet, le but d'une approche optimiste est de gérer l'incertitude relatif aux paramètres dans le critère de sélection en connaissance parfaite. En ayant directement accès aux paramètres nous donne donc un aperçu des performances maximales pouvant être atteintes. De plus en comparant notre algorithme à ce dernier, cela nous permet de voir si le compromis entre exploration et exploitation a bien été géré. En résumé, les différents algorithmes que nous étudierons dans la section suivantes sont :

- **Échantillonnage aléatoire** : c'est la plus simple stratégie de sélection présente uniquement à titre de comparaison, une stratégie d'apprentissage actif doit nécessairement faire mieux

que celle-ci, une donnée d'entrée est tirée aléatoirement selon la distribution naturelle ce qui équivaut à tirer une zone avec une probabilité proportionnelle à son importance relative,

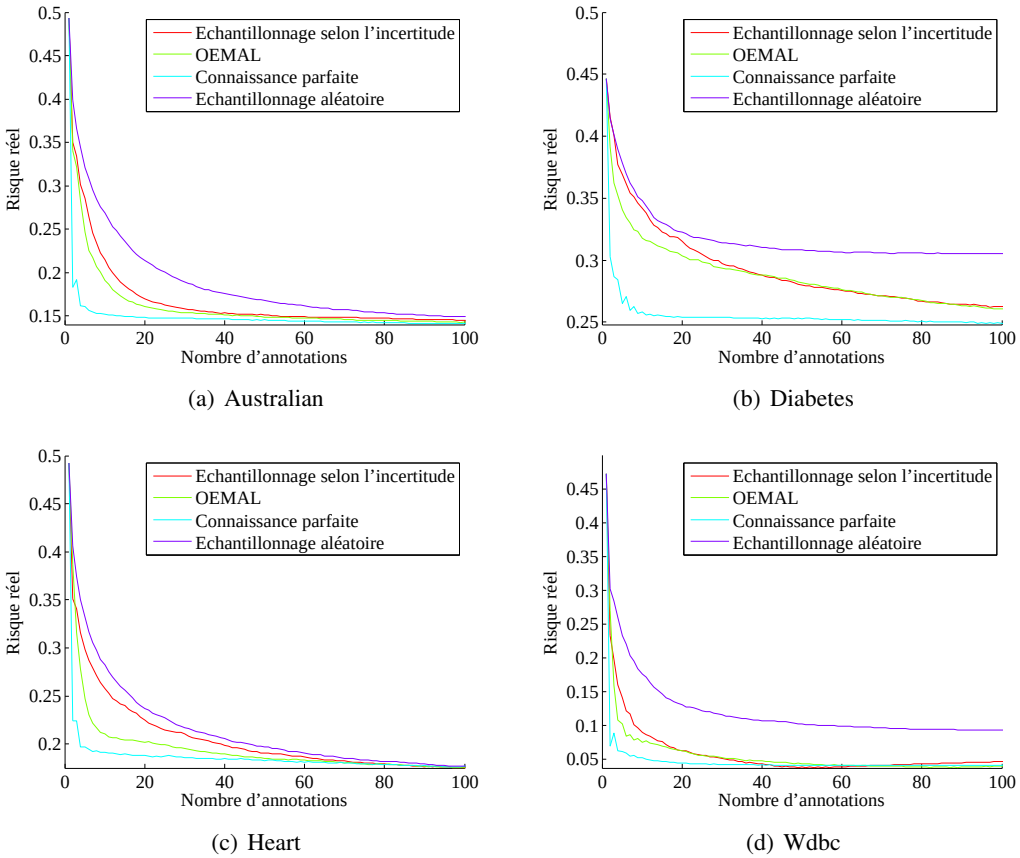
- **Échantillonnage selon l'incertitude** : une stratégie d'apprentissage actif standard, la zone sélectionnée est celle pour laquelle le paramètre estimé est le plus proche de $\frac{1}{2}$,
- **Version en connaissance parfaite** : le critère de sélection en connaissance parfaite définit dans ce chapitre et employé par notre algorithme est ici utilisé directement en lui fournissant les vraies valeurs des paramètres, il sert à voir quelles sont les performances maximales qui peuvent être atteintes par notre algorithme ainsi qu'à évaluer la gestion de l'incertitude dans le critère,
- **OEMAL** : l'algorithme introduit dans ce chapitre, que nous nommons Minimisation de l'Erreur Optimiste pour l'Apprentissage Actif (Optimistic Error Minimization for Active Learning).

9.6 Résultats

Nous commençons par décrire les résultats obtenus avec le premier type de partitionnement. La Figure 9.2 montre ceux obtenus sur les quatre bases de données suivantes : 9.2(a) Australian, 9.2(b) Diabetes, 9.2(c) Heart, 9.2(d) Wdbc. Les résultats ne sont affichés que pour les 100 premiers pas de temps afin de mettre en évidence le gain en performances apporté par l'algorithme optimiste introduit dans ce chapitre. En effet, c'est sur les premiers pas de temps que l'on observe la plus grosse différence de performances entre les algorithmes d'apprentissage actif étudiés ici. Rappelons qu'en apprentissage actif, le but est d'atteindre les meilleures performances avec le moins d'annotations possible. Une bonne performance initiale est donc un atout majeur même si celles-ci sont rattrapées par la suite.

Nous pouvons voir que sur ces bases de données les performances d'OEMAL sont toujours meilleures que celles obtenues par échantillonnage selon l'incertitude. Dans deux bases sur quatre, les performances de ce premier sont rapidement rattrapées par ce dernier, mais elles restent toutefois convenables n'étant pas nettement dépassées. De plus, sur les deux autres bases présentées ici, OEMAL reste distinctement supérieur à l'échantillonnage selon l'incertitude sur toute la durée du processus. Précisons que ceci n'est pas lié à la taille de la base de données puisque le réservoir utilisé sur Heart 9.2(c) est de 104 données d'entrée. Remarquez que lorsque toutes les données du réservoir ont été échantillonnées, les stratégies de sélection ont nécessairement toutes les mêmes performances puisqu'il n'y a plus de liberté dans le choix des données. Notez que sur Heart 9.2(c), le gain en performance entre OEMAL et l'échantillonnage selon l'incertitude est nettement supérieur à celui entre ce dernier et l'échantillonnage aléatoire. De même que pour les performances initiales sur Diabetes 9.2(b). Ceci nous permet de montrer que l'idée d'utiliser l'optimisme face à l'incertitude dans un algorithme d'apprentissage actif peut être envisagé. En effet, il surpasse au moins le plus simple et le plus standard des algorithmes d'apprentissage actif.

En observant les performances de la version en connaissance parfaite de l'algorithme relatif à ce chapitre, nous voyons qu'OEMAL reste encore assez éloigné de la sélection optimale des données d'entrée. Notez cependant que celles-ci ne peuvent pas être confondues dès le début du processus. En effet, la version en connaissance parfaite peut effectuer la meilleure sélection dès le début alors qu'OEMAL doit inévitablement passer par l'apprentissage des paramètres. Toutefois, le but est que la gestion du compromis entre exploration et exploitation permette à OEMAL de se rapprocher au mieux et le plus rapidement de la stratégie idéale. Nous pouvons voir que sur les deux bases de



(a) Australian (b) Diabetes

(c) Heart (d) Wdbc

FIGURE 9.2 – Résultats sur les bases de données tirées du monde réel

données 9.2(a) et 9.2(d) les performances des deux versions sont équivalentes au bout de 100 pas de temps, en plus de Heart 9.2(c) où le ceci est dû à la taille du réservoir.

Question du critère d'arrêt : Un phénomène intriguant peut être remarqué sur la base de donnée Wdbc Figure 9.2(d). Nous pouvons apercevoir la stratégie d'échantillonnage selon l'incertitude atteindre une performance maximale aux alentours de 55 données étiquetées, celle-ci voyant ses performances diminuer par la suite alors que le nombre d'observations augmente. Un tel phénomène peut parfois être rencontré lorsque le classifieur fait face à du sur-apprentissage. N'ayant pas poussé l'étude plus loin, nous préférons ne pas nous avancer sur des causes de ce phénomène. Quoi qu'il en soit, il apparait qu'une performance supérieure puisse être atteinte avec une distribution des données sélectionnées différente de la distribution naturelle. Dans le cas de l'échantillonnage dans un réservoir, la distribution finale des données d'entrée sélectionnées revient nécessairement à la distribution naturelle. La Figure 9.3 affiche les résultats des algorithmes sur une échelle de temps plus grande. Nous pouvons voir que ce phénomène est également présent sur l'algorithme OEMAL. Il serait alors intéressant de pouvoir arrêter d'échantillonner lorsque les performances maximales ont été atteintes. La question d'un critère d'arrêt a déjà été étudiée en apprentissage actif dans le but de limiter le nombre de requêtes à l'oracle ou de ne pas dépenser de budget inutilement lorsque les performances commencent à stagner. Ici, il nous permet en plus de cela d'obtenir des performances non-atteignables autrement. Cela nous amène à penser que même lorsque l'on ne souhaite pas nécessairement réduire le nombre de données annotées il peut être intéressant d'employer tout

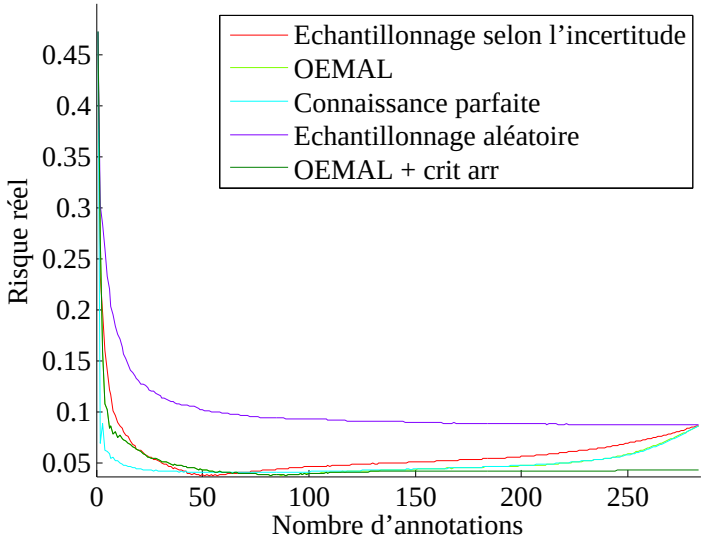


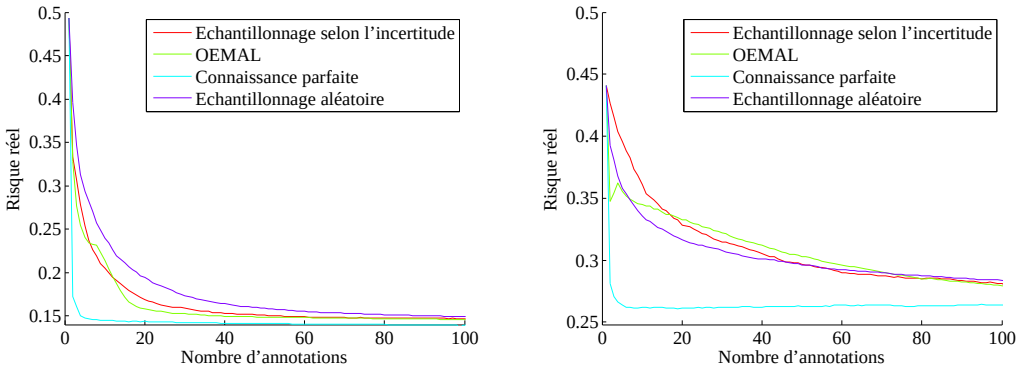
FIGURE 9.3 – Wdbc - grande échelle. Résultats comportant une version d'OEMAL avec critère d'arrêt

Base de donnée	OEMAL avec critère d'arrêt	OEMAL sans critère d'arrêt
Australian	0.1367	0.1382
Diabetes	0.2422	0.3095
Heart	0.1724	0.1722
Wdbc	0.0397	0.0860

TABLE 9.1 – Risque réel moyen final d'OEMAL avec ou sans critère d'arrêt

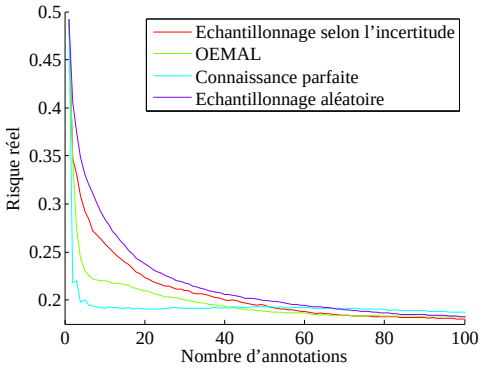
de même l'apprentissage actif combiné à un critère d'arrêt uniquement dans le but d'obtenir de meilleures performances de classification. Dans le cas de l'échantillonnage selon l'incertitude la définition d'un critère d'arrêt n'est pas évidente car il ne possède pas de notion d'évolution de la performance. L'avantage d'une méthode de minimisation de l'erreur est de pouvoir comparer la performance actuelle à celle faisant suite à l'échantillonnage et ainsi de savoir lorsque celle-ci va décroitre. De plus, le critère optimiste utilisé dans notre algorithme fournit directement un critère d'arrêt. En effet, si le critère associé à l'échantillon sélectionné est inférieur à 0, alors il existe une forte probabilité $1 - \delta$ que la performance décroisse à la suite de l'échantillonnage. Celui-ci étant celui qui avait potentiellement la plus grande croissance des performances. Ainsi, si le critère de sélection maximal est inférieur à 0 alors le processus est interrompu. La Figure 9.3 affiche les performances moyennes obtenues en incluant le critère d'arrêt. Nous pouvons voir que les performances finales restent élevées. Elles sont toutefois pas égales à la performance maximale à cause du fait qu'il existe une faible probabilité δ que l'algorithme décide de stopper alors que la performance peut encore monter. Le Tableau 9.1 affiche les résultats finaux de l'algorithme OEMAL avec ou sans critère d'arrêt sur les bases de données étudiées. Nous pouvons voir que dans deux bases sur quatre les performances finales peuvent être améliorées en incluant un critère d'arrêt. Dans les autres cas, il n'existe simplement pas de performance intermédiaire maximale.

Voyons maintenant les résultats des expériences menées avec le deuxième type de partitionnement. Ceux-ci sont affichés sur la Figure 9.4. Nous pouvons voir que OEMAL se comporte mieux

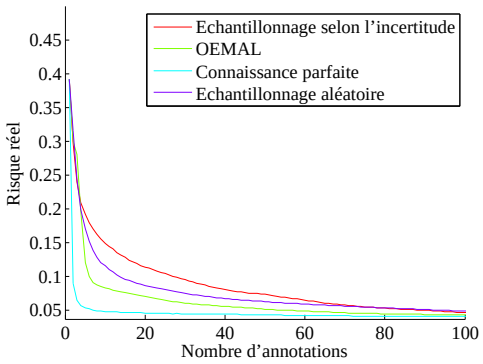


(a) Australian

(b) Diabetes



(c) Heart



(d) Wdbc

FIGURE 9.4 – Résultats sur les bases de données tirées du monde réel

que l'échantillonnage selon l'incertitude à l'exception de la base de donnée Diabetes 9.4(b) dans laquelle les performances initiales sont tout de même meilleures et les performances finales équivalentes. De plus, sur Wdbc 9.4(d), l'échantillonnage selon l'incertitude possède des performances pires que l'échantillonnage aléatoire alors que OEMAL arrive à obtenir une amélioration par rapport à ce dernier. Ceci est probablement dû au fait que le séparateur soit disjoint et qu'il soit nécessaire d'explorer, ce qui est le point faible de l'échantillonnage selon l'incertitude. Les performances d'un algorithme d'apprentissage actif dépendent fortement des capacités de base du classifieur. Par exemple, si nous comparons les performances des d'OEMAL entre les Figures Wdbc 9.2(d) et 9.4(d) nous pouvons voir que l'échantillonnage aléatoire donne déjà de meilleurs résultats pour la version utilisant un noyau Gaussien et donc l'algorithme OEMAL sera aussi meilleur dans ce cas. Toutefois, nous voyons que le gain en performance est plus grand lorsque le classifieur est moins bon. De ce fait, les performances des deux versions sont tout de même assez proches. Ceci n'est pas le cas sur la base de donnée Diabetes 9.2(b) et 9.4(b), dans laquelle les meilleures performances d'OEMAL sont atteintes avec le classifieur le moins bon. Pour tenter d'expliquer cela, nous pouvons remarquer que la version en connaissance parfaite souffre déjà d'une moins bonne performance sur le deuxième type de partitionnement.

9.7 Conclusion intermédiaire

Dans ce chapitre, nous avons introduit un nouvel algorithme d'apprentissage actif qui utilise un classifieur basant sa prédiction sur une combinaison linéaire des estimations indépendantes du paramètre dans différentes zones. Cet algorithme se servant de l'idée de l'optimisme face à l'incertitude, nous avons ainsi montré que ce principe était viable pour l'apprentissage actif et pouvait donner lieu à une stratégie performante. Bien que ce principe avait déjà été utilisé dans les premiers chapitres, les méthodes d'apprentissage actif qui y étaient développées ne pouvaient être employées que sur des problèmes simples sans quoi les performances chutaient drastiquement. Dans un deuxième temps, l'utilisation d'une partition adaptative permettait de travailler avec des problèmes complexes tirés du monde réel mais aucune autre stratégie d'apprentissage actif avait été utilisée à titre de comparaison. Ici, c'est donc la première fois qu'une stratégie d'apprentissage actif utilisant le principe de l'optimisme face à l'incertitude peut à la fois être employée sur un problème tiré du monde réel et être comparée à au moins un algorithme standard d'apprentissage actif. Les résultats obtenus nous confortent donc dans l'idée que ce principe fonctionne et doit faire l'objet d'une comparaison plus approfondie avec d'autres algorithmes usuels. Seulement, plutôt que ce classifieur, il existe une majorité d'algorithmes d'apprentissage actif utilisant une régression linéaire du paramètre. Or, dans le cas où la partition est définie de telle sorte que chaque zone englobe une unique donnée d'entrée, celui-ci utilise une combinaison linéaire des étiquettes pondérées par ce qui peut s'apparenter à une covariance entre les données d'entrée. Ce classifieur se rapproche donc d'une régression linéaire du paramètre à la différence que les poids de chaque donnée d'entrée ne sont pas recalculés à chaque pas de temps. Ceci étant dû à la contrainte stipulant que la partition ne doit pas changer au cours du processus. Dans le prochain chapitre, nous lèverons donc cette contrainte et dériverons l'algorithme d'apprentissage actif introduit ici pour le cas d'une régression linéaire du paramètre.

L'algorithme introduit dans ce chapitre peut aussi faire preuve d'une amélioration que nous n'aborderons pas dans cette thèse mais qu'il est nécessaire de mentionner. Notez que nous avons choisi de définir le critère de sélection en connaissance parfaite comme la minimisation myope du risque réel. Or, ceci n'inclut pas de vision sur le long terme. En effet, il se peut qu'une paire de données d'entrée sélectionnée ensemble permette d'obtenir les meilleures performances mais qu'aucune d'entre elle ne donne la meilleur décroissance du risque réel seule. Ceci a pour effet que la stratégie en connaissance parfaite n'est pas optimale. Pour la rendre optimale, il faudrait que le critère soit convexe quel que soit le chemin suivi dans la suite. Pour cela, une idée serait d'utiliser la somme de critères analogues à ceux du Chapitre 7. Ici, nous avons choisi de privilégier le temps de calcul au détriment d'une performance optimale.

Chapitre 10

Apprentissage Actif dans les Processus Gaussiens

10.1 Introduction

Dans ce chapitre nous nous intéressons à l'utilisation du principe de l'optimisme face à l'incertitude en apprentissage actif dans le cadre des Processus Gaussiens. La principale nouveauté par rapport aux algorithmes développés précédemment est que l'espace d'entrée sera traité de façon continue. Alors que nous n'avons étudié jusqu'ici que des classifieurs utilisant un partitionnement de l'espace d'entrée, regroupant ainsi de nombreuses données sous une même zone, les données seront ici considérées individuellement.

Nous avons vu dans le chapitre précédent que le fait de partager l'information provenant des étiquettes reçues avec l'ensemble des zones permettait de ne plus voir la vitesse d'apprentissage limitée par le nombre de zones de la partition. Ainsi, afin d'obtenir les meilleures performances finales, il était possible de discrétiser l'espace d'entrée à tel point que chaque zone de la partition englobe individuellement chaque donnée d'entrée issue du réservoir. De cette façon, l'ensemble des taux de combinaisons entre les donnée d'entrée spécifiait entièrement la structure des données, cela rappelant la matrice de covariance utilisée dans les méthodes à noyau. Ce chapitre poursuit cette approche faisant fi des zones et en traitant directement les données de façon continues à travers l'utilisation d'une régression du paramètre par moindres carrés régularisés à noyau ou plutôt sa version Bayésienne : les Processus Gaussiens.

Le passage de l'un à l'autre peut sembler direct, cependant rappelons qu'une des contraintes que l'on s'était fixé dans le chapitre précédent était que la partition devait rester fixe et donc que les taux de combinaison n'étaient pas autorisés à évoluer au cours du processus. Or, dans le cas d'une régression par moindres carrés, les poids correspondant à chaque donnée d'entrée sont recalculés à chaque pas de temps. L'inclusion de cette contrainte avait été justifiée par le fait que la prédiction optimale dépendait des poids attribués et que pour évaluer la qualité de la prédiction actuelle vis-à-vis de l'optimale il était préférable que cette dernière ne soit pas amenée à changer. L'autre raison était que le problème était posé comme un bandit à bras multiple dans lequel les distributions de récompenses devaient être fixes. Ceci étant dû au fait que les algorithmes usuels de résolution des bandits à bras multiples suivent une approche fréquentiste pour définir les intervalles de confiances utilisés. Cependant, dans notre problème précédent, bien que les différentes zones ne changeaient pas, la distribution des récompenses associée à une zone, à savoir ici le gain sur le risque réel, évoluait tout de même d'un pas de temps sur l'autre en fonction de l'intérêt de continuer à échantillonner. Or, nous avons vu qu'en suivant une approche Bayésienne, il était possible de constituer

un intervalle de crédibilité cette récompense, et que le principe de l'optimisme face à l'incertitude pouvait également s'appliquer à l'aide de ce type d'intervalles. Cela permettait alors de gérer la non-stationnarité du critère idéal. En suivant ce schéma, l'immuabilité des poids pourrait également être relâchée. L'objectif est donc ici d'évaluer ce que donne le principe de l'optimisme face à l'incertitude appliqué au problème d'apprentissage actif alors qu'à la fois la fonction à maximiser et les poids liés à la prédiction changent à chaque pas de temps.

Les Processus Gaussiens fournissent un cadre Bayésien à la régression du paramètre de l'oracle. En mettant un *a priori* Gaussien sur la valeur du paramètre, et en se basant sur les observations reçues ainsi que sur la covariance entre les données, ils permettent d'obtenir une distribution *a posteriori* sur la valeur du paramètre. Ceux-ci nous seront donc particulièrement utile pour établir les intervalles de crédibilité utilisés. Le principe de l'optimisme face à l'incertitude à déjà été employée conjointement à des Processus Gaussien dans GP-UCB [71]. Cependant, celui-ci traite un problème d'optimisation et la fonction sous-jacente dont on souhaite trouver sélectionner l'optimum le plus souvent possible reste la même tout le long du processus.

Les Processus Gaussiens sont déjà largement appréciés pour le problème d'apprentissage actif car ils fournissent une notion d'incertitude sur la prédiction effectuée utile pour guider la sélection. Ainsi, de nombreuses méthodes d'apprentissage actif utilisant les Processus Gaussiens ont été développées à la fois pour la minimisation du risque réel local et global. Parmi les méthodes existantes, la différence se trouve dans la façon de gérer l'incertitude liée au paramètre de l'oracle. Certaines le remplace directement par son estimation, d'autres utilisent l'espérance sur la distribution *a posteriori* et d'autres encore travaillent avec le pire cas. Ceci nous permettra donc de comparer nos algorithmes face à un ensemble de méthodes répandues, qui plus est dans leur contexte d'origine, toutes utilisant le même classifieur afin de se concentrer sur la solution en elle même. L'objectif est ici de montrer que le principe de l'optimisme face à l'incertitude fournit une alternative au moins comparable voir meilleure que les autres en termes de performances.

Dans ce chapitre, nous introduisons deux algorithmes utilisant le principe de l'optimisme face à l'incertitude dans l'apprentissage actif avec des Processus Gaussiens. Ces deux algorithmes diffèrent dans ce qu'ils cherchent à minimiser. Le premier tente de minimiser le risque local, il cherchera donc à échantillonner l'endroit de l'espace d'entrée où le risque est maximal. Cela consistera alors uniquement à construire un intervalle de confiance sur le risque. Le deuxième tentera de minimiser le risque global, pour cela il devra simuler l'échantillonnage de chaque donnée d'entrée et construire un intervalle de confiance sur la décroissance du risque réel engendrée par chacun. Enfin, notez que les résultats présentés ici ont fait l'objet d'une publication [23].

Ces algorithmes ainsi que ceux que nous voulons utiliser pour la comparaison feront l'objet d'expériences sur des problèmes tirés du monde réel. Nous montrerons alors que nos algorithmes sont capables de concurrencer les algorithmes existants sur ce type de problèmes.

10.2 Les Processus Gaussiens

Les Processus Gaussiens seront utilisés par les algorithmes définis dans ce chapitre à la fois en tant que classifieur et pour fournir la notion d'incertitude sur les paramètres nécessaire au critère de sélection. Avant de décrire plus en avant les différentes stratégies dont font l'objet ce chapitre, il s'agit, dans cette section, de donner la définition ainsi que de décrire la mise en œuvre des Processus Gaussiens.

Un processus stochastique est une généralisation d'une loi de probabilité dans laquelle la variable aléatoire, au lieu d'être un scalaire, ou un vecteur dans le cas multidimensionnel, est une fonction. Ainsi, chaque fonction possiblement définie sur l'espace d'entrée est associée à une pro-

babilité dépendant des valeurs qu'elle prend, chaque donnée d'entrée suivant indépendamment une loi de probabilité propre, et de sa régularité, c'est à dire de la proximité des valeurs pour des données d'entrée voisines. La particularité des Processus Gaussiens est que la loi de probabilité associée à chaque donnée d'entrée indépendamment est une loi Gaussienne.

Il est possible d'effectuer une régression à l'aide d'un Processus Gaussien. Supposons qu'initialement aucune observation de la fonction à régresser n'a encore été reçue. Toutes les fonctions sont alors possibles, on suppose cependant que les valeurs ne sont pas trop éloignées de zéro. On décide donc de placer un *a priori* Gaussien sur cette fonction où la valeur de chaque donnée d'entrée possède une moyenne nulle et une variance fixée. Supposons maintenant que des observations non-bruitées de la fonction à régresser soit reçues. Chacune d'entre elle va alors permettre d'éliminer, parmi toutes les fonctions possibles, celles qui ne correspondent pas à cette observation. Cela va donc modifier la distribution des fonctions possibles et, à travers la contrainte de régularité sur la fonction, modifier la distribution de probabilité associée aux valeurs des autres données d'entrées. Dans le cas où les observation sont bruitées, la distribution de fonctions va uniquement être modifiée pour favoriser les fonctions passant à proximité de l'observation reçue.

Plus formellement, soit une fonction à régresser $\mu \in Y^X$. Soit $\mathcal{T}_n = \{(x_i, y_i)\}_{i\in\{1,...,n\}}$ la base d'apprentissage contenant une liste de n observations de cette fonction. Dans le cas bruité, les observations sont obtenues avec une erreur supposée Gaussienne, ainsi $y_i = \mu(x_i) + \epsilon$, avec $\epsilon \sim \mathbb{N}(0, \sigma^2)$. On note $\mathbf{x} = (x_1, ..., x_n)^t$ le vecteur contenant l'ensemble des données d'entrée incluses dans la base d'apprentissage et $\mathbf{y} = (y_1, ..., y_n)^t$ le vecteur contenant les résultats correspondants. Soit x^* une donnée d'entrée générique dont on souhaite prédire la valeur.

Soit une fonction de covariance entre les valeurs de sortie $k \in [0, 1]^{X^2}$, celle-ci spécifie la régularité d'une fonction tirée selon le Processus Gaussien. Par conséquence, la distribution jointe *a priori* des valeurs observées et de la valeur à prédire se présente sous la forme

$$\begin{bmatrix} \mathbf{y} \\ \mu(x^*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\mathbf{x},\mathbf{x}) + \sigma^2 I_n & k(\mathbf{x}, x^*) \\ k(\mathbf{x}, x^*)^t & k(x^*, x^*) \end{bmatrix}\right),$$

avec $k(\mathbf{x}, \mathbf{x})$ la matrice contenant l'ensemble des covariances entre chacune des paires de données d'entrée issue de la base d'apprentissage, $k(\mathbf{x}, x^*) = (k(x_1, x^*), ..., k(x_n, x^*))^t$ et I_n la matrice identité de taille n.

Ainsi, en conditionnant la probabilité jointe ci-dessus aux observations, nous obtenons la distribution *a posteriori* suivante :

$$\mu(x^*) \sim \mathcal{N}(m_{\mathcal{T}_n}(x^*), \sigma^2_{\mathcal{T}_n}(x^*)),$$

avec

$$m_{\mathcal{T}_n}(x^*) = k(\mathbf{x}, x^*)^t (k(\mathbf{x},\mathbf{x}) + \sigma^2 I_n)^{-1}\mathbf{y} \tag{10.1}$$

la moyenne de la distribution associée à la valeur à prédire et

$$\sigma^2_{\mathcal{T}_n}(x^*) = k(x^*, x^*) - k(\mathbf{x}, x^*)^t (k(\mathbf{x},\mathbf{x}) + \sigma^2 I_n)^{-1} k(\mathbf{x}, x^*) \tag{10.2}$$

sa variance. Remarquez que la fonction de moyenne de la distribution *a posteriori* est équivalente à une régression par moindre carrés régularisés utilisant le même noyau. En passant par une approche Bayésienne, le Processus Gaussien permet d'ajouter à cela la variance de la distribution *a posteriori* qui agit comme une incertitude sur la prédiction effectuée.

Dans le contexte de classification binaire qui nous intéresse, l'idée va être d'effectuer une régression du paramètre de l'oracle sur l'espace d'entrée pour ensuite baser la prédiction sur le signe

le plus probable du paramètre. Afin de se servir des Processus Gaussiens comme définis précédemment sans nécessiter d'adapter les équations, il convient que la fonction que l'on souhaite régresser puisse être doté d'un *a priori* de moyenne nulle. Pour cela, nous changeons les valeurs associées aux étiquettes par 1 et -1 (Y = {-1,1}). L'oracle est donc toujours représenté par une loi de Bernoulli telle que

$$\mathbb{P}(y = 1|x) = \frac{\mu(x) + 1}{2}.$$

Celle-ci est donc toujours caractérisée par un unique paramètre $\mu(x)$ qui peut être associé à l'espérance des étiquettes reçues pour une donnée d'entrée x fixée.

Notez que les étiquettes agissent ici comme une observation bruitée du paramètre de l'oracle. Le bruit étant d'espérance nulle. Pour des raisons de simplicité, ce bruit sera approximé lors de la régression du paramètre par un bruit Gaussien de variance 1 ($\sigma^2 = 1$).

La classifieur qui dérive de ces Processus Gaussiens prédit pour une donnée d'entrée $x \in X$ quelconque l'étiquette correspondant au signe le plus probable du paramètre $\mu(x)$. Étant donné que la distribution Gaussienne est symétrique, cela revient à considérer uniquement le signe de fonction de moyenne de la distribution *a posteriori*. Ainsi,

$$f_n(x) = signe(m_{\mathcal{T}_n}(x)).$$

10.3 Le critère de sélection en connaissance parfaite

Cette section a pour but de concevoir les stratégies de sélection idéales sous l'hypothèse que la fonction de paramètre relative à l'oracle est entièrement connue. Celle-ci devront donc nous informer sur l'intérêt que porte l'échantillonnage d'une donnée d'entrée dans l'amélioration de la performance du classifieur. L'effet d'une nouvelle donnée d'entrée étant parfaitement maitrisée de par la connaissance de l'étiquette ainsi reçue et du risque réel engendré par les prédictions courantes du classifieur. Les critères ainsi définis serviront ensuite de stratégie cible pour les algorithmes qui seront définis dans la section suivante. Ici, deux stratégies seront considérées. La première se base sur le risque local en sélectionnant la donnée d'entrée pour laquelle il est maximal. Ceci a pour effet de minimiser l'erreur maximale commise par la prédiction associée à une donnée d'entrée quelconque. La deuxième se base sur le risque global en sélectionnant la donnée d'entrée qui permet d'obtenir la décroissance maximale. Pour cela elle devra considérer l'impact de l'échantillonnage sur la prédiction des données d'entrée sur l'ensemble de l'espace. L'intérêt d'une telle stratégie est qu'elle permet de considérer la représentativité des données, c'est à dire de favoriser les données issues de zones densément peuplées. Ceci permet d'obtenir une qualité de prédiction supérieure pour les données d'entrée qui seront effectivement soumises au classifieur durant son utilisation.

Tachons d'abord de définir le risque local. Celui-ci correspond pour une donnée d'entrée spécifique $x \in X$ à la probabilité que l'oracle renvoie une étiquette différente de la prédiction. On note

$$r_t(x) = \begin{cases} \frac{1+\mu(x)}{2} & \text{si } f_t(x) = -1 \\ \frac{1-\mu(x)}{2} & \text{si } f_t(x) = 1 \end{cases}$$

le risque local au temps t. Soit la fonction objectif suivante,

$$L_t = \arg\max_{x \in X} r_t(x)$$

Afin de minimiser cette dernière, la première stratégie est alors simplement celle qui consiste à échantillonner la donnée d'entrée

$$x_{t+1}^{loc} = \arg\max_{\mathcal{U}} r_t(x).$$

En effet, l'apport d'une nouvelle observation va nécessairement améliorer la qualité de la prédiction localement. De plus, même si l'échantillonnage d'une donnée d'entrée voisine permet aussi d'améliorer sa prédiction, l'impact maximal est observé pour la donnée d'entrée correspondante.

Le risque global, quant à lui, consiste en l'espérance du risque local sur la distribution naturelle. On note

$$R_t(\mathcal{T}_t) = \int_X r_t(x) dP(x)$$

le risque global. Cependant, la distribution naturelle nous est inconnue. Toutefois, il est possible de l'estimer en considérant l'ensemble des données initialement non-étiquetées contenues dans le réservoir qui suivent cette distribution. Nous définissons donc le risque global empirique :

$$\tilde{R}_t(\mathcal{T}_t) = \sum_{x \in \mathcal{R}} r_t(x).$$

Celui-ci nous sert également de fonction objectif pour la deuxième stratégie.

Définir une stratégie qui la minimise est NP-dur. Cependant, il est courant d'effectuer une approximation myope. Ainsi, nous définissons la stratégie qui consiste à échantillonner la donnée d'entrée pour laquelle la décroissance du risque global qu'elle engendre est maximale. Pour cela, il est nécessaire de simuler la simulation de chaque donnée d'entrée encore non-étiquetée x_s. La stratégie de minimisation du risque global est alors

$$x_{t+1}^{glo} = \arg\max_{\mathcal{U}} \Delta_{x_s} \tilde{R}_t(\mathcal{T}_t),$$

avec

$$\Delta_{x_s} \tilde{R}_t(\mathcal{T}_t) = \tilde{R}_t(\mathcal{T}_t) - \tilde{R}_t(\mathcal{T}_t \cup \{(x_s, y_s)\}),$$

où y_s est l'étiquette qui sera reçue pour la donnée d'entrée x_s.

Nous venons donc de définir deux stratégies de sélection en connaissance parfaite. Celles-ci ne peuvent pas être utilisées comme tel puisque la fonction de paramètre de l'oracle n'est pas connue. Il reste donc à définir les deux algorithmes qui vont permettre de se rapprocher au maximum de chacune de ces stratégies tandis que la fonction de paramètre doit être apprise simultanément.

10.4 Prise en compte de l'incertitude

La démarche suivie pour construire les algorithmes d'apprentissage actif ici est la même dans les chapitres précédents. La section précédente a permis de définir plusieurs stratégies de sélection idéales dépendant de la fonction de paramètres de l'oracle. Cette dernière doit être apprise à travers les observations qui proviennent des données d'entrée dont elle souhaite déjà guider la sélection. Les intérêts divergent donc entre les données d'entrée permettant d'améliorer le classifieur et celles permettant d'obtenir le bon critère de décision. Il est donc nécessaire de gérer un compromis entre l'exploration de l'espace d'entrée et l'exploitation du critère estimé. Cela se fait en considérant l'incertitude sur le critère de sélection en connaissance parfaite et en utilisant le principe de l'optimisme face à l'incertitude.

La méthode que nous employons ici apparait identique à celle du chapitre précédent. En effet, hormis le fait que l'espace d'entrée est traité ici de façon continue et non-plus à travers un partitionnement aussi fin que possible, ainsi que les poids associés aux données d'entrée changent à chaque pas de temps, ce qui ne modifie pas fondamentalement le calcul du critère de sélection, la forme des distributions mises en jeu ainsi que la façon de les utiliser reste les mêmes. La différence est tout de même que la distribution *a posteriori* sur les paramètres de l'oracle provient directement du Processus Gaussien et que les données non-étiquetées n'interviennent plus ni dans la prédiction du classifieur ni dans la distribution *a posteriori*.

Le procédé consiste donc à construire un intervalle de crédibilité sur le critère de sélection en connaissance parfaite désiré, puis à utiliser la borne supérieure de cet intervalle en tant que critère de sélection en connaissance partielle. Nous montrons dans la suite comment la borne supérieure de cet intervalle est calculé pour les différents critères.

Données : $k \in [0,1]^{X^2}$, δ
Initialisation : $\mathcal{T}_0 = \emptyset$
pour *t = 1, ..., n* **faire**
  $\forall x \in \mathcal{U}$, calculer $m_{\mathcal{T}_t(x)}$ et $\sigma^2_{\mathcal{T}_t(x)}$ selon Eq. (10.1) et (10.2)
  $\forall x \in \mathcal{U}$, calculer $\epsilon_t^{loc}(x)$ selon Eq. (10.3)
  $x_{t+1}^{loc} = \arg\max_{x \in \mathcal{U}} \epsilon_t^{loc}(x)$
  $y_{t+1} \sim P(y|x_{t+1}^{loc})$
  $\mathcal{T}_{t+1} = \mathcal{T}_{t+1} \cup (x_{t+1}^{loc}, y_{t+1})$
fin
Résultat : $\forall x \in X, [m_{\mathcal{T}_n}(x)]$

Algorithme 9 : OLRM : algorithme d'apprentissage actif pour un classifieur utilisant les Processus Gaussiens employant le principe de l'optimisme face à l'incertitude pour la minimisation du risque local

Nous commençons par considérer le premier critère de sélection basé sur le risque local. Dans un premier temps, étudions le cas où $f_t(x) = -1$. Ainsi,

$$r_t(x) = \frac{1 + \mu(x)}{2}.$$

Le Processus Gaussien entrainé avec les données étiquetées au temps t nous renvoie la distribution *a posteriori* suivante :

$$\mu(x) \sim \mathcal{N}(m_{\mathcal{T}_t}(x), \sigma^2_{\mathcal{T}_t}(x)),$$

avec $m_{\mathcal{T}_t}(x)$ et $\sigma^2_{\mathcal{T}_t}(x)$ provenant des équations 10.1 et 10.2. Ainsi, $\forall \epsilon_t^{loc}(x) \in [0,1]$,

$$\begin{aligned}
& \mathbb{P}(r_t(x) \geq \epsilon_t^{loc}(x)) = \delta \\
\iff\ & \mathbb{P}(\mu(x) \geq 2\epsilon_t^{loc}(x) - 1) = \delta \\
\iff\ & 2\epsilon_t^{loc}(x) - 1 = \Phi^{-1}(1-\delta, m_{\mathcal{T}_t}(x), \sigma^2_{\mathcal{T}_t}(x)) \\
\iff\ & 2\epsilon_t^{loc}(x) - 1 = -\Phi^{-1}(\delta, -m_{\mathcal{T}_t}(x), \sigma^2_{\mathcal{T}_t}(x)) \\
\iff\ & \epsilon_t^{loc}(x) = \frac{1 - \Phi^{-1}(\delta, -m_{\mathcal{T}_t}(x), \sigma^2_{\mathcal{T}_t}(x))}{2},
\end{aligned}$$

avec Φ la fonction de répartition de la loi Normale.

De la même manière, si $f_t(x) = 1$,

$$r_t(x) = \frac{1 - \mu(x)}{2}.$$

Et donc, $\forall \epsilon_t^{loc}(x) \in [0, 1]$,

$$\begin{aligned} & \mathbb{P}(r_t(x) \geq \epsilon_t^{loc}(x)) = \delta \\ \iff & \mathbb{P}(\mu(x) \leq 1 - 2\epsilon_t^{loc}(x)) = \delta \\ \iff & 1 - 2\epsilon_t^{loc}(x) = \Phi^{-1}(\delta, m_{\mathcal{T}_t}(x), \sigma^2_{\mathcal{T}_t}(x)) \\ \iff & \epsilon_t^{loc}(x) = \frac{1 - \Phi^{-1}(\delta, m_{\mathcal{T}_t}(x), \sigma^2_{\mathcal{T}_t}(x))}{2}. \end{aligned}$$

Et puisque $f_t(x) = signe(m_{\mathcal{T}_t}(x))$, alors on peut résumer les deux cas dans l'équation suivante :

$$\begin{aligned} & \mathbb{P}(r_t(x) \geq \epsilon_t^{loc}(x)) = \delta \\ \iff & \epsilon_t^{loc}(x) = \frac{1 - \Phi^{-1}(\delta, |m_{\mathcal{T}_t}(x)|, \sigma^2_{\mathcal{T}_t}(x))}{2}. \end{aligned} \tag{10.3}$$

Ainsi, $\epsilon_t^{loc}(x)$ joue le rôle de la borne supérieur d'un intervalle de crédibilité qui contient le risque réel local avec une probabilité δ. En suivant le principe de l'optimisme face à l'incertitude, notre algorithme l'utilise comme critère de sélection ce qui donne la stratégie suivante :

$$x_{t+1}^{loc} = \arg\max_{\mathcal{U}} \epsilon_t^{loc}(x).$$

Pour le critère de sélection basé sur le risque réel global, la démarche suivie est identique à celle du chapitre précédent, nous rappelons tout de même les détails de sa mise en œuvre. La décroissance du risque global dépend des valeurs de l'ensemble des paramètres associés à chacune des données d'entrée incluse dans le réservoir. Le calcul d'une distribution de probabilité sur la décroissance devra donc combiner les distributions *a posteriori* sur chacun de ces paramètres. Afin de rendre ce calcul possible, nous simplifierons le problème en considérant que les paramètres ne peuvent prendre que deux valeurs. Cette approximation pourrait sembler être un retour en arrière par rapport à l'étude d'un problème bruité, cependant, cet aspect n'est pas perdu étant donné que la probabilité de chaque cas provient de la régression de ce paramètre. Remarquez que cela revient aussi à étudier les cas possibles pour les étiquettes des données d'entrée d'une base de test qui coïnciderait avec l'emplacement des données dans le réservoir. Ce qui peut être vu comme un niveau d'estimation supplémentaire du risque global.

Soit la simulation de l'échantillonnage d'une donnée d'entrée $x_s \in \mathcal{U}$. L'étiquette reçue en retour est alors

$$y_s = \begin{cases} 1 & \text{avec un probabilité} \quad \Phi_1 = 1 - \Phi(0, m_{\mathcal{T}_t}(x_s), \sigma^2_{\mathcal{T}_t}(x_s)) \\ -1 & \text{avec un probabilité} \quad \Phi_{-1} = \Phi(0, m_{\mathcal{T}_t}(x_s), \sigma^2_{\mathcal{T}_t}(x_s)). \end{cases} \tag{10.4}$$

Nous traitons ensuite chacun des cas séparément.

Le but est de calculer la décroissance du risque global engendré par cette échantillonnage. La distribution *a posteriori* sur les paramètres ainsi que la prédiction sont recalculées en prenant en compte la nouvelle observation. Nous calculons donc indépendamment les décroissance du risque local pour chaque donnée d'entrée du réservoir $x \in \mathcal{R}$, celle-ci est égale à

$$\Delta_{x_s, y_s} r_t(x) = \mathbb{1}_{f_t(x) = signe(\mu(x))} - \mathbb{1}_{f_{t+1}(x) = signe(\mu(x))},$$

Données : $k \in [0,1]^{X^2}$, δ
Initialisation : $\mathcal{T}_0 = \emptyset$
pour *t = 1, ..., n* **faire**
  $\forall x_s \in \mathcal{U}$, calculer Φ_1 et Φ_{-1} selon Eq. (10.4)
  $\forall x_s \in \mathcal{U}, \forall y_s \in \{-1,1\}, \forall x \in \mathcal{R}, \forall q \in \{-1,0,1\}$ calculer $\mathbb{P}(\Delta_{x_s,y_s} r_t(x) = q)$ selon Eq. (10.5)
  $\forall x_s \in \mathcal{U}, \forall Q \in \{-car(\mathcal{R}), ..., card(\mathcal{R})\}$, calculer $\mathbb{P}(\Delta_{x_s} R_t(x) = Q)$ selon Eq. (10.6) et (10.7)
  $\forall x_s \in \mathcal{U}$ calculer $\epsilon_t^{glo}(x_s)$ Eq. (10.8)
  $x_{t+1}^{glo} = \arg\max_{x_s \in \mathcal{U}} \epsilon_t^{glo}(x_s)$
  $y_{t+1} \sim P(y|x_{t+1}^{glo})$
  $\mathcal{T}_{t+1} = \mathcal{T}_{t+1} \cup (x_{t+1}^{glo}, y_{t+1})$
fin
Résultat : $\forall x \in X, [m_{\mathcal{T}_n}(x)]$

Algorithme 10 : OGRM : algorithme d'apprentissage actif pour un classifieur utilisant les Processus Gaussiens employant le principe de l'optimisme face à l'incertitude pour la réduction du risque global

avec $f_{t+1}(x) = signe(m_{\mathcal{T}_t \cup \{(x_s,y_s)\}})$. En utilisant la nouvelle probabilité *a posteriori*, la probabilité que $signe(\mu(x)) = -1$ est égale à $\Phi(m_{\mathcal{T}_t \cup \{(x_s,y_s)\}}, \sigma^2_{\mathcal{T}_t \cup \{(x_s,y_s)\}})$. Ainsi les cas possibles pour la décroissance du risque local sont :

$$\Delta_{x_s,y_s} r_t(x) = \begin{cases} 0 & \text{avec un probabilité} \quad 1 & \text{si } f_{t+1}(x) = f_t(x) \\ -1 & \text{avec un probabilité} \quad \Phi(0, |m_{\mathcal{T}_t \cup \{(x_s,y_s)\}}|, \sigma^2_{\mathcal{T}_t \cup \{(x_s,y_s)\}}) & \text{si } f_{t+1}(x) \neq f_t(x) \\ 1 & \text{avec un probabilité} \quad 1 - \Phi(0, |m_{\mathcal{T}_t \cup \{(x_s,y_s)\}}|, \sigma^2_{\mathcal{T}_t \cup \{(x_s,y_s)\}}) & \text{si } f_{t+1}(x) \neq f_t(x). \end{cases} \tag{10.5}$$

En effet, si les prédictions avant et après échantillonnage sont identiques, alors la décroissance est nulle quelle que soit la valeur du paramètre. Dans le cas contraire, la décroissance est négative si la paramètre a le même signe que la prédiction après échantillonnage.

Finalement, l'ensemble des cas possibles pour les décroissances du risque local sont combinés pour obtenir une distribution de probabilité dur la décroissance du risque global sachant l'étiquette reçue. Ainsi,

$$\mathbb{P}(\Delta_{x_s,y_s} \tilde{R}_t(\mathcal{T}_t) = Q) = \sum_{\sum_{x \in \mathcal{R}} q(x) = Q} \prod_{x \in \mathcal{R}} \mathbb{P}(\Delta_{x_s,y_s} r_t(x) = q(x)). \tag{10.6}$$

Il reste ensuite à combiner les cas relatifs à l'étiquette y_s. Ce qui donne la distribution de probabilité pour la décroissance du risque global suivante :

$$\mathbb{P}(\Delta_{x_s} \tilde{R}_t(\mathcal{T}_t) = Q) = \Phi_1 \mathbb{P}(\Delta_{x_s,1} \tilde{R}_t(\mathcal{T}_t) = Q) + \Phi_{-1} \mathbb{P}(\Delta_{x_s-1} \tilde{R}_t(\mathcal{T}_t) = Q). \tag{10.7}$$

Notez qu'ici, la décroissance du risque local est considérée dans sa globalité après avoir reçu l'échantillon simulé. En effet, le paramètre ne change pas au cours de la simulation, seule la prédiction change. Ainsi, seule la distribution *a posteriori* recalculée (plus fine) est utilisée. Ceci est différent du cas où le risque estimé avant la simulation est comparé au risque estimé après la simulation que l'on retrouve dans la plupart des algorithmes de minimisation de l'erreur existants.

Une autre remarque est que l'utilisation d'une méthode combinatoire peu paraitre extrêmement couteuse. Cependant, seuls les données d'entrée qui voient leur prédiction changer au cours de la simulation possèdent des cas distincts. Aussi, hormis pour les phases initiales de l'apprentissage, seules quelques données d'entrée voisines voient leur prédiction évoluer. Ceci permet l'utilisation de cette méthode dans un temps raisonnable.

La borne supérieure $\epsilon_t^{glo}(x_s)$ d'un intervalle de crédibilité sur la décroissance du risque global en est déduite relativement à un paramètre δ quelconque afin qu'elle soit telle que

$$\mathbb{P}(\Delta_{x_s}\tilde{R}_t(\mathcal{T}_t) \geq \epsilon_t^{glo}(x_s)) = \delta. \tag{10.8}$$

La stratégie de sélection consiste alors à choisir la donnée d'entrée correspondant à la borne maximale. Ainsi,

$$x_{t+1}^{glo} = \arg\max_{x_s \in \mathcal{U}} \epsilon_t^{glo}(x_s).$$

10.5 Expériences

Les algorithmes qui viennent d'être introduits dans les sections précédentes vont maintenant faire l'objet d'une évaluation afin de déterminer si ils fonctionnent de la manière attendue et permettent d'obtenir de bons résultats. Cette évaluation consistera en une série d'expériences menées sur des problèmes sélectionnés dans le but de mettre en évidence différents comportements. Cette section a pour but de présenter et de définir les paramètres et autres détails de ces expériences.

L'intérêt de ce chapitre repose essentiellement sur le fait que de nombreux algorithmes d'apprentissage actif déjà existants utilisent le même classifieur. Cela nous permettra de montrer que nos algorithmes sont capable de s'inscrire dans l'état de l'art en obtenant des performances comparables ou supérieures à celles déjà obtenues. De plus, les algorithmes sélectionnés pour la comparaison sont tous une variation d'une même méthode et ne différent que par la façon dont l'incertitude est gérée. Par conséquence, cela nous permettra de valider l'utilisation du principe de l'optimisme face à l'incertitude comme une solution viable pour l'apprentissage actif.

La première série d'expériences ressemble à celles qui ont déjà été effectuées dans les chapitres précédents. Elle utilise plusieurs bases de données issues du UCI Machine Learning Repository [56] qui proviennent du monde réel. Comme précédemment, les algorithmes seront lancés un certain nombre de fois sur chacune de ces bases de données, ici 1000 fois, et leurs performances enregistrées à chaque pas de temps. Avant chaque exécution, les bases de données seront divisées en deux parties égales, l'une étant cachée des algorithmes et réservée pour tester la qualité de la prédiction des classifieurs, et l'autre servant de réservoir dans lequel il est possible de piocher des données d'entrée pour en recevoir l'étiquette associée. Les algorithmes sont ensuite initialisés avec un seul échantillon pris aléatoirement dans le réservoir, puis à chaque pas de temps la stratégie de sélection choisi une donnée d'entrée dont elle souhaite connaitre l'étiquette. Le classifieur est réentrainé à chaque pas de temps avec l'ensemble des étiquettes reçues et ses performances sont calculées en comparant sa prédiction sur la base de test avec les vraies étiquettes qui lui sont cachées. La performance globale d'un algorithme d'apprentissage actif donné est représenté par l'ensemble sur chaque pas de temps des moyennes des performances à travers les exécutions, donnant ainsi l'évolution des performances et donc la vitesse d'apprentissage du classifieur.

La deuxième expérience que nous avons souhaité mettre en œuvre est menée sur un problème créé de toutes pièces. Cela nous permet de mieux contrôler les caractéristiques du problème afin de mettre en évidence le comportement souhaité et de pouvoir interpréter les résultats obtenus. Ainsi,

l'une des qualité que doit posséder notre algorithme est la gestion du compromis entre exploration et exploitation, en effet, c'est pour cette raison que nous avons choisi d'utiliser le principe de l'optimisme face à l'incertitude. Pour mettre cela en évidence, il faut que le problème considéré possède plusieurs séparateurs distincts. En effet, un algorithme qui n'explorerait pas concentrerait les données étiquetées à proximité de la première frontière découverte, et par continuité l'ensemble de cette même frontière, mais ignorerait complètement une autre frontière existante. Nous avons donc choisi d'utiliser un problème connu sous le nom de XOR, qui sépare l'espace d'entrée 2D en quatre, les cases diagonales étant de classes opposées. Pour générer les données d'entrée quatre distributions Gaussiennes centrées en (0.25,0.25), (0.75,0.75), (0.25,0.75) et (0.75,0.25) sont utilisées, dont les deux premières fournissent des données étiquetées -1 et les deux autres étiquetées 1. Dans le but d'accentuer le phénomène lié à l'exploration, les données sont très peu bruitées afin que le bruit ne force pas l'exploration. La variance des distribution Gaussiennes est donc fixée à 0.1.

Les algorithmes d'apprentissage actif qui ont été utilisés pour la comparaison sont les suivants :

- **Échantillonnage aléatoire** : ce n'est pas à proprement parler une stratégie de sélection des données d'entrée puisque celle-ci n'est pas guidée par un critère. Ces dernières sont simplement tirées selon la distribution naturelle, c'est à dire sélectionnées aléatoirement dans le réservoir. Cela constitue le point de comparaison le plus simple que toute méthode d'apprentissage actif se doit de dépasser,

- **Échantillonnage selon l'incertitude** : la méthode d'apprentissage actif, la plus simple et la plus standard, celle-ci consiste à sélectionner à chaque pas la donnée d'entrée dont le critère de décision servant à la prédiction est le plus proche de $\frac{1}{2}$. Ceci peut être vu comme un algorithme de minimisation du risque local où le paramètre de l'oracle est directement remplacé par son estimation, et n'inclus donc aucune exploration,

- **GPAL (Uncertainty)** : un algorithme basé sur la minimisation du risque local créée dans le but d'améliorer la stratégie précédente en incluant de l'exploration. Celui-ci gère l'incertitude sur les paramètres en effectuant l'espérance sur leur distribution *a posteriori*. Le critère résultant peut aussi être vu comme pondérant le critère de l'algorithme précédent par l'inverse de l'incertitude provenant des Processus Gaussiens dans le but de favoriser les données d'entrée dont la prédiction est effectuée avec le moins d'information, c'est à dire les plus éloignées des données étiquetées.

- **QUIRE** : un algorithme de réduction de l'erreur remplaçant les paramètres inconnus par leur valeur la plus probable et prenant comme valeur pour l'étiquette simulée celle menant au pire risque.

- **VOI** : un algorithme de réduction de l'erreur effectuant l'espérance du risque réel sur la distribution *a posteriori* des paramètres avant et après simulation de l'échantillonnage de chaque donnée d'entrée considérée,

- **OLRM** : l'algorithme introduit dans ce chapitre basé sur la minimisation du risque local et utilisant le principe de l'optimisme face à l'incertitude pour faire face aux paramètre inconnus

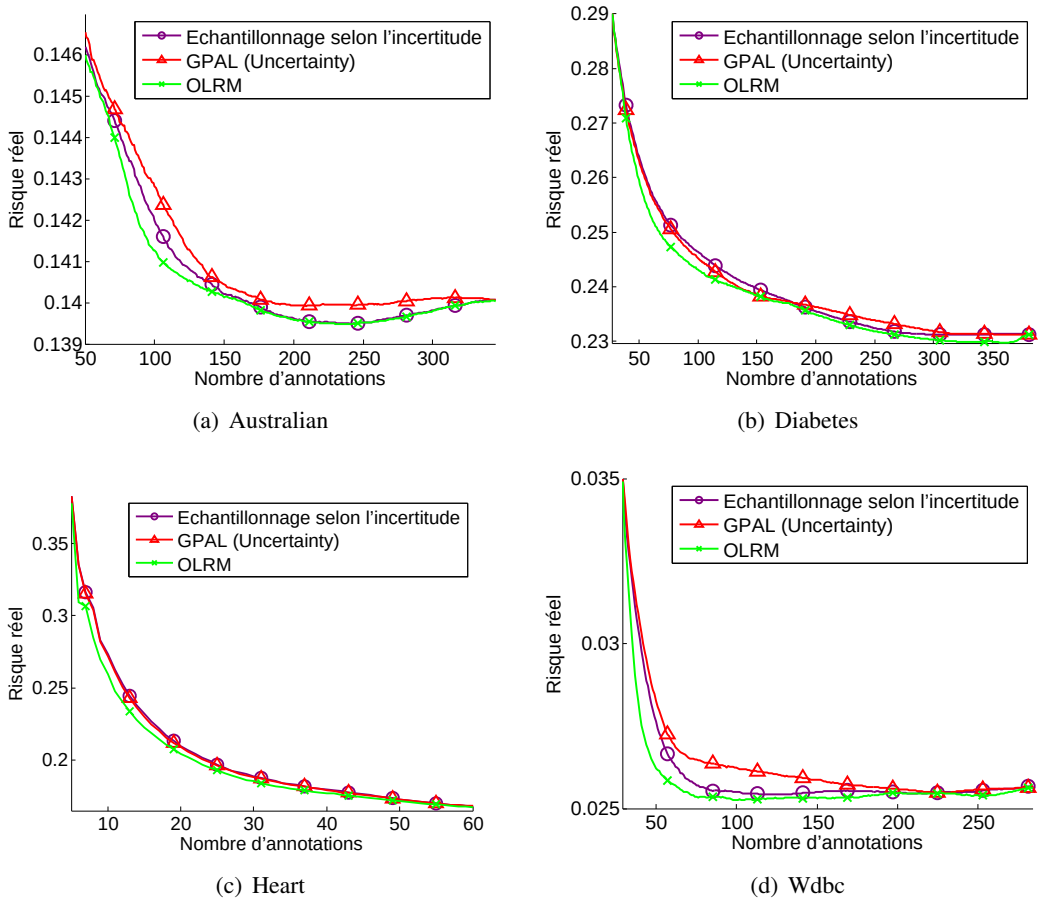
- **OGRM** : l'algorithme introduit dans ce chapitre basé sur la réduction du risque global utilisant aussi le principe de l'optimisme face à l'incertitude.

10.6 Résultats

Dans cette section, il s'agit de présenter les résultats des expériences décrites précédemment. Nous commençons par comparer entre eux uniquement les algorithmes basé sur la minimisation du risque local, à savoir l'échantillonnage selon l'incertitude, GPAL et OLRM. Leurs résultats sont disponibles sur la Figure 10.1 qui montre l'évolution du risque réel subit par le classifieur en fonction du budget d'annotations utilisé pour son apprentissage. Les courbes obtenues sont extrêmement proches, pour pouvoir les distinguer il a donc fallu zoomer sur les parties intéressantes. Étant donné la nature décroissante et convexe de ces courbes, elles finissent par s'aplatir ce qui implique de devoir se passer de l'affichage de la phase initiale afin de récupérer la plus grande plage exploitable. Ceci n'est cependant pas pénalisant car les performances sur la phase initiale est identiques. Il est ensuite important d'apporter une précision quand à la courbe de performances d'OLRM. Cet algorithme possède un paramètre δ qui doit être fixé initialement. Dans nos travaux, aucune indication n'a été donnée quant à la valeur idéale à utiliser. L'objectif étant uniquement de montrer que le principe de l'optimisme face à l'incertitude est viable, trouver une valeur de δ qui convient nous suffit. Pour cette raison, celui-ci fait habituellement dans nos travaux l'objet d'une recherche par quadrillage. En outre, dans de nombreuses publication dans le domaine des bandits à bras multiples, ce paramètre peut être fixé en fonction du budget connu à l'avance voire même d'évoluer au cours du processus. Dans cette optique, nous avons ici retenu à chaque pas de temps, la performance maximale qui a pu être obtenu par l'algorithme en utilisant un paramètre quelconque, celui-ci étant toujours fixe au cours d'une exécution. Ainsi, il ne faut pas voir la courbe comme donnant les performance d'une exécution possible de l'algorithme, mais plutôt les performances qu'il est possible d'atteindre pour chaque budget distinct, en supposant que le paramètre puisse être choisi idéalement en fonction du budget. Notez par ailleurs que l'échantillonnage selon l'incertitude correspond exactement à l'algorithme OLRM avec un paramètre fixé à $\frac{1}{2}$. Ceci étant dû au fait que la fonction de répartition de la Gaussienne est symétrique. C'est la raison pour laquelle l'algorithme OLRM donne obligatoirement des performances supérieures à ce dernier.

Nous pouvons voir que sur deux bases de données sur quatre10.1(a) et 10.1(d), l'algorithme de l'échantillonnage selon l'incertitude donne de meilleurs résultats que GPAL. Ceci est étonnant puisque ce dernier à été conçu comme une amélioration du premier permettant de prendre en compte l'incertitude du Processus Gaussien pour permettre l'exploration de l'espace d'entrée qui lui faisait défaut. Ici, cela apparait comme un effet négatif sur les performances. Cela pourrait s'expliquer par le fait que si les données sont très peu bruitées et ne possède pas de séparateurs disjoints, alors l'exploration de l'espace d'entrée est inutile et gaspille une partie du budget. Cette interprétation est renforcé par le fait que les deux bases de données dans lesquelles l'échantillonnage selon l'incertitude dépasse significativement GPAL sont les deux les moins bruitées, ce qui est appuyé par le fait que risque réel final soit faible. D'autre part, nous pouvons voir que l'algorithme OLRM donne toujours des performances supérieures aux deux autres algorithmes lorsque le nombre d'annotation est inférieur à 150. Cependant, sur le long terme, dans deux bases de données sur quatre 10.1(a) et 10.1(b) le meilleur paramètre pour OLRM est $\frac{1}{2}$, ce qui revient à utiliser l'échantillonnage selon l'incertitude. Ce que nous pouvons retenir de cette première expérience est que sur ces bases de données, l'algorithme OLRM se comporte globalement comme l'échantillonnage selon l'incertitude, avec une légère amélioration si le budget est connu et que le paramètre idéal peut être choisi en fonction de celui-ci. Or, ces bases de données ne reflètent pas forcement la nécessité d'exploration des algorithmes d'apprentissage actif.

Nous nous penchons maintenant sur le problème créé de toutes pièce dans le but d'exagérer les problèmes liés au manque d'exploration. Ici, la base de donnée est très peu bruitée pour ne pas



(a) Australian (b) Diabetes

(c) Heart (d) Wdbc

FIGURE 10.1 – Résultats sur les bases de données tirées du monde réel

engendrer d'exploration si celle-ci n'est pas spécifiquement inclue dans la stratégie de sélection. De plus le séparateur est disjoint de sorte qu'il soit nécessaire d'explorer. Les résultats sont affichés sur la Figure 10.2. Nous pouvons voir que échantillonnage selon l'incertitude donne les pires performances, suivies de GPAL. Tout d'abord, notons que celles-ci sont en particulier pires que celles de l'échantillonnage aléatoire. Ce qui implique que la sélection active des données dégrade les performances. En effet, l'échantillonnage aléatoire est la stratégie qui explore le plus puisqu'elle tire les données à étiqueter directement selon la distribution naturelle. Ainsi, aucune région n'est délaissée ce qui assure une qualité de prédiction minimale sur l'ensemble de l'espace d'entrée. Au contraire, l'échantillonnage selon l'incertitude est capable d'ignorer complètement des zones alors que la prédiction n'est pas juste et correspond à une large part du risque. En observant les performances de l'échantillonnage selon l'incertitude, nous pouvons constater une marche. Celle-ci est justement due au fait qu'elle découvre un premier séparateur et tente de l'améliorer jusqu'à ce que le hasard fasse qu'un autre séparateur soit découvert. Dans le problème étudié ici, le XOR, la première frontière correspond à séparer un quart de l'espace avec le reste, en ignorant le quart opposé. Cela peut être confirmé par le fait que de la première marche corresponde à une valeur du risque réel aux alentours de 0.25. Alors que le but de GPAL est de subvenir à cette nécessité d'exploration, nous pouvons voir qu'il possède un comportement équivalent. La première marche est cependant moins nette que dans le cas de l'échantillonnage de l'incertitude, mais elle existe tout de même. De plus, les performances sont toujours moins bonnes que celles de l'échantillonnage aléatoire. OLRM,

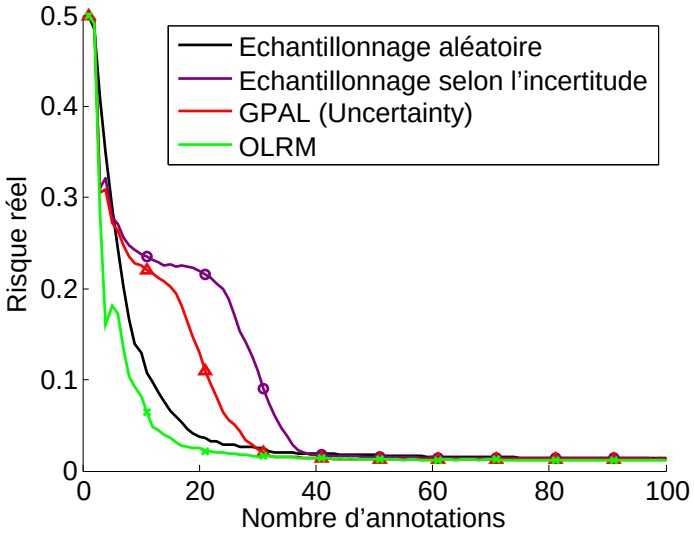
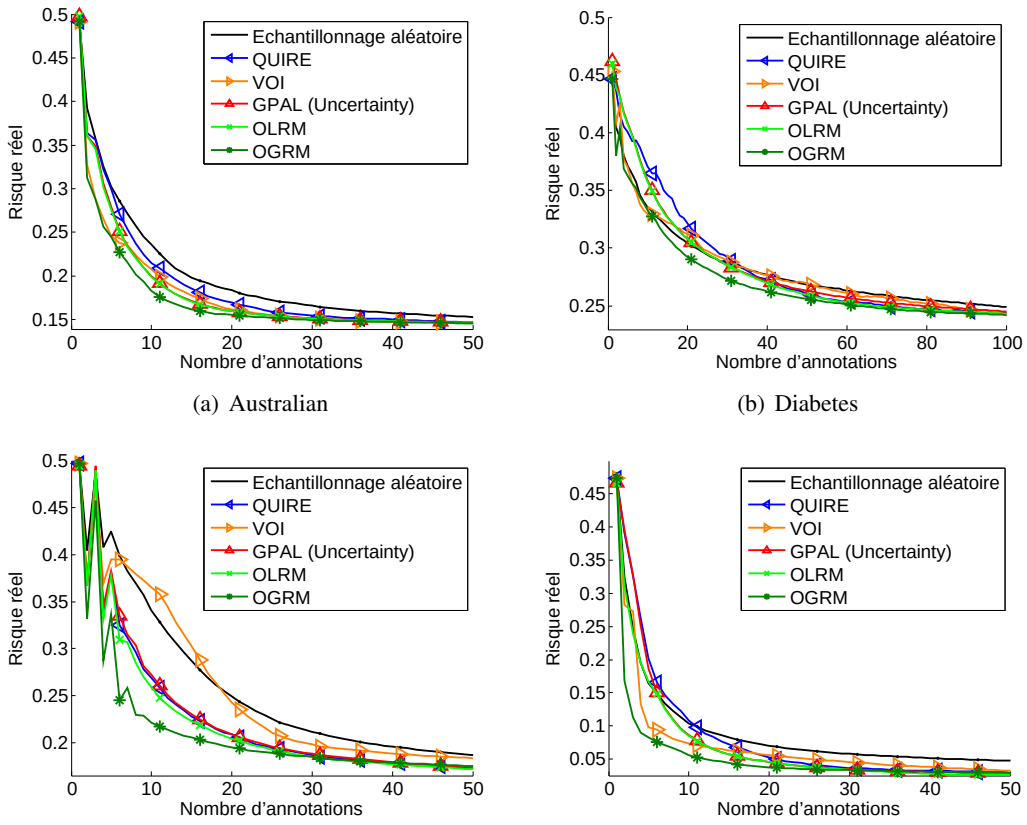


FIGURE 10.2 – Résultats sur les bases de données tirées du monde réel

quant à lui, arrive à explorer convenablement l'espace d'entrée tout en tirant parti des informations des étiquettes reçues pour concentrer les requêtes à l'oracle à proximité des frontières découvertes. Nous pouvons donc voir que ses performances sont meilleures celles de l'échantillonnage aléatoire et surpassent donc celles des autres algorithmes. Précisons qu'ici, une seule valeur du paramètre a été utilisée ($\delta = 0.1$) au sein d'OLRM, celle-ci restant fixe au cours du processus. Finalement, nous pouvons conclure des deux dernières expériences que l'algorithme OLRM donne des performances comparables aux algorithmes existants lorsque ceux-ci fonctionnent correctement, notamment sur des données tirées du monde réel, et est capable de donner de meilleurs résultats que ceux-ci lorsque le problème est difficile.

Intéressons nous maintenant aux algorithmes basés sur la minimisation du risque global à travers la simulation des échantillonnages possibles. L'algorithme qui nous intéresse en particulier ici est OGRM que nous avons introduit dans ce chapitre. Les autres algorithmes auxquels nous nous comparons sont QUIRE, qui remplace les étiquettes inconnues par la pire valeur et VOI qui effectue l'espérance du risque global sur la distribution *a posteriori* des paramètres de l'oracle. Les performances de ces algorithmes peuvent être observées sur la Figure 10.3 Tout d'abord, rappelons la différence entre la minimisation du risque global et celle du risque local qui est que le premier cas prend en considération la distribution naturelle des données d'entrée afin de favoriser les zones densément peuplée. En effet, celles-ci seront plus sollicitées lors de l'utilisation du classifieur et donc une meilleure précision pour la prédiction est recommandée, ce qui implique plus de données étiquetées. D'un autre point de vue, en raisonnant uniquement sur le réservoir, une étiquette acquise dans une zone dense modifie également la prédiction de toutes les données d'entrée voisines. Cette qualité de la stratégie de sélection est couramment appelée "représentativité" en opposition à l'"informativité" correspondant à l'amélioration de la précision locale. La conséquence de ceci est qu'il est possible de commencer à guider la sélection des données sans avoir encore reçu d'étiquettes, uniquement en privilégiant les zones denses. Ainsi, nous pouvons voir que sur la Figure 10.3 que l'algorithme OGRM accroit ses performances initiales beaucoup plus rapidement que les algorithmes basés sur la minimisation du risque local évoqués dans le paragraphe précédent. C'est aussi le cas de VOI sur deux bases de données sur quatre 10.3(a) et 10.3(b), mais celles-ci se dégradent vite jusqu'à devenir moins bonnes que ces dernières. De plus, c'est pour cette raison



(a) Australian (b) Diabetes

(c) Heart (d) Wdbc

FIGURE 10.3 – Résultats sur les bases de données tirées du monde réel

que sur la base de donnée Diabetes 10.3(b), la stratégie d'échantillonnage aléatoire est meilleure que OLRM et GPAL. En effet, celle-ci tire les données à étiqueter selon la distribution naturelle, et par conséquent, fournit d'avantage de données provenant des zones denses. Quant à QUIRE, ses performances sont toujours pires voire égale sur une seule base de donnée aux algorithmes de minimisation du risque local. Finalement, nous concluons en disant que d'une part, OGRM est meilleur que les algorithmes de minimisation du risque local, ce qui montre que la prise en compte de la représentativité des données est nécessaire. Et d'autre part, qu'il meilleur que les autres algorithmes de minimisation du risque global. Ainsi, puisque la seule différence réside dans la façon de gérer l'incertitude, cela permet de dire que le principe de l'optimisme face à l'incertitude se présente comme une solution efficace pour ce problème. Ceci ouvre donc de nouvelles perspectives dans le domaine de l'apprentissage actif.

10.7 Conclusion intermédiaire

Dans ce chapitre, nous avons introduit deux nouvelles méthodes d'apprentissage actif travaillant avec un classifieur dérivant des Processus Gaussiens et qui utilise le principe de l'optimisme face à l'incertitude. L'une basée sur la minimisation du risque local dans le but de garantir la plus faible erreur de classification moyenne commise pour une donnée d'entrée quelconque. L'autre basée sur la réduction du risque global afin de minimiser l'erreur de classification commise pour une donnée

d'entrée tirée aléatoirement selon la distribution naturelle. La première nouveauté par rapport au chapitre précédent est que les données d'entrée sont considérées de façon continues et non plus à travers une partition. Cependant, celle-ci est anecdotique puisque cette partition pouvait être aussi fine que possible jusqu'à isoler toutes les données d'entrée du réservoir, ce qui devenait équivalent à les traiter de façon continue. La réelle nouveauté est qu'ici la combinaison linéaire des étiquettes reçues définissant le critère de décision pour la prédiction de l'étiquette à attribuer voit ses poids recalculés à chaque pas de temps. Ainsi, la structure du problème considéré n'était plus fixe, cela ajouté au fait que le critère de sélection idéal évoluait lui aussi à chaque pas de temps. Toutefois, l'approche Bayésienne que constituait les Processus Gaussien a permis l'utilisation du principe de l'optimisme face à l'incertitude sur ce critère changeant. Le problème a donc été de savoir, à travers des évaluations empiriques, si ce principe se révélait encore efficace dans ce cas.

Deux algorithmes ont donc été introduits en suivant la même démarche. Tout d'abord, le critère de sélection en connaissance parfaites ont été exprimés. Ceux-ci conditionnent la prochaine donnée à annoter idéalement en connaissant exactement sa répercution sur le risque réel. Puis, le principe de l'optimisme face à l'incertitude à été employé pour gérer l'incertitude sur ces deux critères de sélection provenant du fait que les paramètres de l'oracle ne sont pas connus initialement et doivent être appris. Celui-ci consistant à utiliser la borne supérieur d'un intervalle de confiance défini sur le critère idéal souhaité. Nous avons montré comment utiliser la distribution *a posteriori* renvoyée par les Processus Gaussiens pour construire des intervalles de crédibilité relatifs aux deux critères de sélection en connaissance parfaite venant d'être exprimés.

Des expériences ont ensuite été menées afin de valider les algorithmes introduits. Pour cela, nous avons utilisé des bases de données tirées du monde réel. Pour l'évaluation d'OLRM, un problème jouet à aussi été utilisé dans le but de mettre en évidence le comportement spécifique des algorithmes face à la nécessité d'exploration de l'espace d'entrée. Ainsi, nous avons pu voir qu'OLRM se comporte aussi bien que les autres algorithmes tentant de minimiser le risque local sur les bases de données tirées du monde réel. De plus, lorsque confronté à un problème difficile nécessitant la gestion efficace du compromis entre exploration et exploitation, dans lequel les algorithmes comparés échouent, OLRM arrive a obtenir de bons résultats. Ainsi, l'algorithme OLRM possède des performances comparables aux autres algorithmes sur la plupart des problèmes mais est plus général, conservant ces performances sur une plus large étendue de problèmes. Dans un deuxième temps, nous avons vu que l'algorithme OGRM, en plus d'avoir pour objectif la réduction de l'erreur qui est fondamentalement plus efficace que la minimisation du risque local car tirant parti de la densité des données d'entrée, se comportait mieux que les autres algorithmes de réduction de l'erreur ayant servi à la comparaison. Finalement, nous pouvons conclure de ces l'étude de ces deux algorithmes que le principe de l'optimisme face à l'incertitude fournit une solution alternative pour la gestion de l'incertitude dans l'un ou l'autre des critères de sélection idéaux qui est potentiellement plus efficace que ceux existants.

Cependant, plusieurs sujets restent à aborder. En effet, le paramètre lié à la probabilité relative aux intervalles de crédibilité et permettant de contrôler la quantité d'exploration de l'algorithme est ici passé en argument des différents algorithmes. La valeur à attribuer à ce paramètre pose question. Lors de nos expériences, celui-ci a été obtenu par recherche par quadrillage, avec sa valeur pouvant soit dépendre du budget soit être unique, celle-ci restant toujours fixe sur la durée du processus. Une étude théorique approfondie pourrait permettre de donner une valeur idéale en fonction du budget initialement défini. D'autre part, ce paramètre pourrait être autorisé à évoluer durant le processus. C'est le cas par exemple dans GP-UCB [71] ou Bayes-UCB [51] dont les algorithmes introduits en sont tous deux inspirés. Des évaluations supplémentaires ainsi qu'une étude théorique pourrait aussi être effectués dans ce contexte. Finalement, ce chapitre montre que le principe de l'optimisme face

à l'incertitude possède un potentiel intéressant dans le domaine de l'apprentissage actif, et ouvre la voie à de nouvelles perspectives.

Chapitre 11

Conclusion

11.1 Conclusion générale

Cette thèse avait pour objectif d'étudier la possibilité d'utiliser le principe de l'optimisme face à l'incertitude en tant que solution alternative pour la gestion des paramètres inconnus dans le critère idéal permettant de sélectionner la prochaine donnée d'entrée à soumettre à l'oracle dans le cadre de l'apprentissage actif pour la classification. Dans cette optique, nous avons introduit et évalué expérimentalement plusieurs algorithmes ayant tous en commun l'utilisation de ce principe, mais une mise en œuvre différente. En effet, ceux-ci avaient pour but de guider l'apprentissage de classifieurs différents et chacun d'entre eux a été conçu spécifiquement pour ces derniers afin de s'y adapter et de tirer pleinement profit de leurs caractéristiques. De plus, l'ensemble de ces algorithmes se sont inscrits dans une démarche de progression, partant d'un classifieur simple, avec des comportements visualisables permettant de comprendre et de développer l'algorithme, pour finalement arriver à un classifieur plus compétitif mais faisant intervenir des comportements plus complexes. En outre, cette progression prenait sa source sur des travaux existants, que nous avons adaptés et fait évolué. Ceux-ci utilisant déjà le principe de l'optimisme face à l'incertitude, mais pour l'allocation d'un budget d'échantillonnage dans une partition fixe et dans un contexte lié à la régression. Vis-à-vis de ces travaux, l'objectif était double, d'une part il fallait les adapter au contexte de classification, et d'autre part il fallait les étendre au traitement continu des données d'entrée. Nous allons maintenant détailler la démarche effectuée ainsi que ce que l'on peut tirer des résultats obtenus.

Avant toute chose, nous avons commencé par étudier au Chapitre 3 les algorithmes existants dans le domaine de l'apprentissage actif. Ce travail préliminaire nous a d'abord servi à appréhender la façon dont ce problème était habituellement abordé afin d'en reprendre les bases. Il nous a aussi permis d'en comprendre les enjeux à travers les comportements attendus ainsi que ceux faisant défaut afin d'orienter nos recherches dans le but de combler un manque. En outre, recenser les méthodes existantes permet de s'assurer de ne pas reproduire de travaux déjà effectués. De notre côté, nous avons effectué une synthèse des algorithmes présentés et fait part de notre point de vue pour ainsi pousser légèrement plus loin la vision générale déjà exprimée dans les articles de référence. Ainsi, nous avons vu que les algorithmes d'apprentissage actifs pouvaient d'abord être catégorisés en trois grands sous-domaines en fonction de l'angle sous lequel était abordé le problème. Les deux premiers basant leur stratégie directement sur les performances du classifieur étudié, en considérant soit le risque local soit le risque global du classifieur. Le dernier renverse le problème pour travailler dans l'espace des classifieurs. Ceci nous a permis de décider de situer nos travaux au sein de ces deux premiers sous-domaines, car ils se prêtaient mieux à l'utilisation

du principe de l'optimisme face à l'incertitude. Nous avons ensuite mis en évidence le fait que l'ensemble des algorithmes appartenant à chacun de ces deux sous-domaines pouvaient être vus comme des déclinaisons d'un même algorithme ayant en commun le critère idéal qu'ils tentent de suivre. Les différents algorithmes correspondant à différentes façons de gérer l'incertitude liée aux paramètres de l'oracle dans le critère idéal. A travers cela, nous avons positionné la solution que nous envisagions au sein du contexte général du domaine en montrant qu'elle s'y inscrivait tout à fait. Nous avons finalement vu que la gestion du critère idéal inconnu soulevait le problème du dilemme entre exploration et exploitation. Le principe de l'optimisme face à l'incertitude ayant fait ses preuves pour gérer ce dilemme, celui-ci se présentait comme une solution idéale pour la gestion de l'incertitude sur le critère inconnu en apprentissage actif.

Dans le Chapitre 4, nous avons étudié le problème de l'optimisation en budget fini duquel provient le principe de l'optimisme face à l'incertitude. Ceci permettant de voir comment ce principe était classiquement mis-en-œuvre afin d'être capable de l'adapter à l'apprentissage actif. Nous avons également comparé les autres solutions afin de pouvoir remettre en cause notre choix. Nous avons donc d'abord présenté le problème des bandits à bras multiples servant généralement de cadre à l'étude de ce dilemme ainsi que le problème de l'optimisation en budget fini dans les Processus Gaussiens. Nous en avons ensuite entrevu les solutions les plus courantes. Nous nous sommes attardé sur les méthodes se référant au principe de l'optimisme face à l'incertitude pour lesquelles nous avons discuté leur performances relatives aussi bien théoriques qu'empiriques. Ce qui nous a permis de mettre en avant les qualités d'une approche Bayésienne. Pour finir, nous avons étudié d'autres travaux faisant mention de la présence du dilemme entre exploration et exploitation dans le problème d'apprentissage actif. Nous avons vu que, comme la notre, la plupart des solutions envisagées provenaient de l'optimisation en budget fini. Cependant, dans le contexte de classification, l'usage du principe de l'optimisme face à l'incertitude n'avait pas encore été largement exploité. L'un deux le faisant intervenir au sein d'une étape mais pas à la base de l'algorithme. Toutefois, un algorithme utilise ce principe dans le cadre de l'apprentissage actif pour la régression. Ainsi, nous en avons conclu que le principe de l'optimisme face à l'incertitude n'avait pas encore été envisagé pour l'apprentissage actif en classification mais qu'il existait des pistes montrant que cette solution possédait de bonnes qualités pour ce problème et méritait que l'on s'y intéresse. Enfin, l'étude de ce dernier algorithme nous a incité à nous en servir comme point de départ de notre démarche évolutive.

Après avoir bien étudié l'état de l'art, nous avons entamé la présentation des travaux effectués en tant que contribution au domaine. Tout d'abord il s'est agit de planifier la démarche évolutive que l'on souhaitait suivre dans le but de développer efficacement un algorithme d'apprentissage actif optimiste dont on contrôle tous les aspects. Pour cela, nous avons détaillé les différents problèmes traités et les classifieurs associés ainsi que la façon dont ils s'articulent entre eux. Nous avons ensuite mis en place un schéma de construction pour servir à l'ensemble de nos algorithmes en s'inspirant de la méthodologie habituellement suivie pour développer un algorithme optimiste. Nous avons ainsi donné les instructions pour développer les différents critères de sélection en connaissance parfaite. Puis, nous avons donné le procédé général servant à établir des intervalles de crédibilité sur ce critère idéal. La borne supérieur de ces intervalles devant ensuite être utilisée en tant que critère définitif. Nous avons ainsi d'abord indiqué dans les grandes lignes comment obtenir une distribution *a posteriori* à l'aide d'une approche Bayésienne. Le type de distributions ainsi que le moyen de les obtenir à parti des observations disponibles devant de toutes évidences être adapté spécifiquement aux problèmes étudiés. Puis nous avons montré la façon d'utiliser cette distribution *a posteriori* dans la définition des intervalles de crédibilité. Il ne restait donc plus qu'à appliquer ce schéma de construction à chaque tape de la démarche évolutive en tenant compte de leur spécificités.

La première étape de cette démarche, étudiée au Chapitre 6, se positionnait comme une étude préliminaire vis-à-vis du problème d'apprentissage actif. Elle avait pour objet l'emploi d'une partition fixe de l'espace d'entrée avec des zones indépendantes tout en ne considérant que la recherche de l'allocation optimale du budget d'annotations. Cherchant ainsi uniquement à adapter les travaux étudiés dans [13] et [14] à la classification. Ce problème extrêmement simplifié avait pour but d'étudier les mécanismes propres à l'apprentissage actif hors de toutes perturbations et d'étudier l'effet du principe de l'optimisme face à l'incertitude. La simplicité du problème nous a permis d'une part de définir un critère ayant un vision sur le long-terme de l'influence d'un échantillonnage dans une zone donnée. Le critère de sélection en connaissance parfaite, en étant ainsi privé de toutes approximations, était ainsi optimal. Et d'autre part, la distribution *a posteriori* sur les paramètres pouvait être le plus juste possible, permettant ainsi à l'intervalle de crédibilité ainsi établi d'être exact. De cette façon, la seule source d'erreur restante revenait à l'utilisation de la solution en elle même, ce qui permettait donc de voir si ce choix était raisonnable. Finalement, des évaluations ont été menées sur des problèmes jouets pouvant représenter tout problème faisant usage d'une partition fixe, même ceux étant issu du monde réel. Nous avons pu en tirer que le principe de l'optimisme face à l'incertitude pouvait s'appliquer à la classification aussi bien qu'à la régression. De plus, cela a permis de valider le principe de l'optimisme face à l'incertitude comme une solution efficace. Le contexte était bien évidemment extrêmement simplifié mais cette étude préliminaire nous a permis d'appréhender les possibilités de cette solution.

Plutôt que d'obtenir l'allocation optimale, dans le Chapitre 7, l'objectif recherché était de directement maximiser les performances du classifieur courant. Le classifieur utilisé a été conservé, et, avec lui, l'ensemble des éléments mentionnés au paragraphe précédent, à savoir la simplicité du problème, la vision à long terme garantissant l'optimalité du critère de sélection en connaissance parfaite, le calcul d'une distribution *a posteriori* exacte. Pour cela, nous avons dû modifier l'interprétation que l'on avait du problème. En effet, bien que les principes mis en jeux étaient les même, les objectifs différaient. Dans le cas précédent, seule importait la proximité l'allocation finale obtenue après exécution l'algorithme avec l'allocation optimale. Le but étant ainsi de minimiser le regret simple. Ici, le but est de sélectionner à chaque pas de temps le meilleur critère idéal, celui-ci étant de toutes façon recalculé après chaque décision. Cela revient donc à minimiser le regret cumulé. Les évaluations, menées sur les mêmes problèmes jouets que précédemment, ont montré que les algorithmes nouvellement introduits permettaient d'obtenir des performances au moins comparables aux algorithmes de l'état de l'art. Ceci nous a permis d'une part d'en déduire que le principe de l'optimisme face à l'incertitude restait viable avec cette forme de critère idéal. Dans la suite nous n'avons donc utilisé plus que celle-ci. Et d'autre part, que cette solution est capable de concurrencer celles déjà existantes.

Ensuite, nous avons introduit petit à petit de la complexité au problème dans le but de tendre vers un classifieur standard. Pour commencer, nous avons envisagé, au Chapitre 8, d'autoriser à re-partitionner l'espace d'entrée au cours du processus tout en conservant l'indépendance des zones. Cependant, nous avons vu qu'il était nécessaire que le re-partitionnement respecte un certain nombre de règles, comme ne pas redéfinir ou supprimer les frontières existantes, les zones pouvant seulement être subdivisées. Ceci nous ramenant ainsi à une partition adaptative. Pour cela, nous avons donc décidé d'utiliser un algorithme d'arbre incrémental. En outre, nous avons développé notre propre algorithme d'arbre incrémental en nous servant des outils que nous étions habitués à utiliser vis-à-vis du principe de l'optimisme face à l'incertitude. Ensuite, nous avons développé l'algorithme d'apprentissage actif basé sur ce classifieur utilisant à peu de chose près le même critère de sélection que dans le cas de la partition fixe. Finalement, les évaluations menés à la fois sur des problèmes jouets, servant à la comparaison avec la partition fixe, et sur des problèmes tirés

du monde réel, servant à la comparaison avec les algorithmes de l'état de l'art, nous ont permis de valider notre algorithme d'arbre incrémental. Cependant, nous avons vu que, sur les problèmes du monde réel, ce classifieur ne laissaient pas une grande marge de manœuvre à notre stratégie d'apprentissage actif.

Nous avons ensuite décidé, au Chapitre 9, d'opter pour une autre voie de complexification, en conservant l'immuabilité de la partition mais relâchant la contrainte d'indépendance des zones, ceci résultant en l'étude d'une partition fixe à zones combinées. Cela avait pour effet de devoir prendre en compte, dans le critère idéal, l'influence d'un échantillonnage dans une zone sur l'ensemble des prédictions du classifieur. D'une part nous avons vu que ce problème était plus coûteux en opérations que le précédent étant donné que les critères idéaux dépendent chacun de l'ensemble des paramètres. Mais également que l'inférence des paramètres de l'oracle dans chaque zone était moins aisée que précédemment. Pour cela, nous avons décidé d'effectuer une approximation Gaussienne sur la distribution de l'estimateur des paramètres afin d'obtenir simplement un distribution *a posteriori* sur ces derniers. Était aussi couteux le transfert des distributions *a posteriori* sur les paramètres à celle sur la prédiction optimale puisqu'elle dépend de l'ensemble des ceux-ci. Nous avons donc également introduit une méthode permettant de la calculer rapidement. Nous avons ensuite évalué cet algorithme sur des données d'entrée tirées du monde réel en les comparant aux courbes de référence en apprentissage actif. A partir des résultats obtenus, nous en avons conclu que l'adaptation de notre solution à ce problème était un succès et que celle-ci présentait encore un certain potentiel pour être confrontée aux autres solutions. Pour finir, malgré l'utilisation d'une partition, cette méthode possédait l'avantage de ne pas voir le nombre de zones limiter la vitesse d'apprentissage comme c'était le cas précédemment. Ainsi, il était possible de discrétiser à l'extrême l'espace d'entrée jusqu'à ce que chaque zone n'englobe plus qu'une seule donnée d'entrée provenant du réservoir, ceci étant ainsi équivalent à traiter continu les données de façon continues.

Lors du développement de cette dernière méthode, nous avons vu que les données d'entrée pouvaient être traitées de façon individuelles. De plus, les données d'entrée étaient reliées entre elles par l'influence qu'elles avaient l'une sur l'autre. Ceci nous a permis de faire le rapprochement avec l'utilisation d'une méthode à noyau. Le Chapitre 10 a donc a donc été l'objet de l'adaptation de notre solution sur un classifieur dérivé d'une méthode à noyau, à savoir les Processus Gaussiens. Nous avons vu que cela faisait intervenir une difficulté supplémentaire. En effet, les poids de la combinaison linéaire devaient dès lors être autorisés à changer au cours du processus pour s'adapter à la distribution des données échantillonnées. Nous avons donc adapté notre solution aux Processus Gaussiens dans le but de voir si elle restait valable sous ces conditions. La dérivation du critère étant en soit très similaire à la précédente, avec l'avantage de pouvoir utiliser directement la distribution *a posteriori* sur les paramètres renvoyée par les Processus Gaussiens. Nous avons effectué des évaluations sur des problèmes tirés du monde réel qui ont permis de montrer que notre solution arrivait encore à obtenir de bons résultats. Nous en avons conclu que, bien que les poids du problème changeaient, le principe de l'optimisme face à l'incertitude restait une solution adéquate, ceci étant en particulier dû à l'utilisation d'une approche Bayésienne. De plus, lorsque confrontée aux autres méthodes standard d'apprentissage actif, nos algorithmes donnaient de meilleurs performances, ceux-ci ne différant que par la façon de gérer les paramètres inconnus. Nous en avons déduit que le principe de l'optimisme face à l'incertitude se présentait comme une solution efficace pour ce problème.

Finalement, au travers de cette thèse, nous avons étudié en profondeur l'adaptation du principe de l'optimisme face à l'incertitude au problème d'apprentissage actif. La mise en œuvre d'un développement progressif nous a permis de bien maitriser chaque aspect de l'algorithme final en justifiant toutes les approximations effectuées et essayant de maximiser les performances de chaque

sous-fonction utilisée. Ceci nous a permis de voir quels étaient les obstacles à une adaptation directe du principe de l'optimisme face à l'incertitude telle qu'il était utilisé son contexte habituel. Nous avons ainsi mis en évidence la non-stationnarité du critère idéal découlant du principe même de l'apprentissage actif. De plus, nous avons vu que les arguments de valeur du paramètre et de nombre d'échantillons n'étaient pas découplés dans le critère idéal. Un des points clés de notre solution était l'utilisation d'une approche Bayésienne, qui pouvait tirer parti de la connaissance de la famille de distributions en classification et permettait de gérer la non-stationnarité du critère idéal. Nous avons également positionné notre méthode au sein des travaux existants ce qui a permis d'évaluer jusqu'où s'étendaient nos contributions. Un grand intérêt à été accordé aux évaluations qui ont permis d'interpréter les comportements résultants, de valider nos algorithmes ainsi que de les comparer avec les algorithmes équivalents de l'état de l'art, montrant ainsi que notre solution était la plus adaptée.

11.2 Perspectives et travaux futurs

Les résultats que nous venons de présenter sont probants dans le sens où nous avons montré des exemples d'algorithmes d'apprentissage actif utilisant le principe de l'optimisme face à l'incertitude comme alternative aux solutions permettant de gérer les paramètres de l'oracle dans le critère idéal, et que les évaluations empiriques ont permis de conclure que ces algorithmes avaient les capacités de détrôner ceux déjà existants. Ainsi, l'objectif de cette thèse est atteint, celui-ci étant d'estimer la viabilité de cette solution. Cependant, ce travail ne constitue qu'une première approche. En effet, seuls quelques classifieurs ont été utilisés, ceux-ci ayant été sélectionnés par nos soins du fait de la possibilité d'obtenir facilement une distribution *a posteriori* sur les paramètres. De plus, les problèmes qui ont servi aux expériences étaient soit tirés du monde réel, ce qui permettait d'avoir une idée de l'utilisation pratique de nos algorithmes, soit créés de toutes pièces dans le but de forcer la présence de certaines caractéristiques pour mettre en évidence les comportements souhaités. Toutefois, ces problèmes restaient en nombre limité et il nous était impossible d'entrevoir tous les aspects possibles. Cependant, cela ouvre la voie à de nombreuses perspectives.

Dans l'immédiat, nous souhaiterions concentrer notre étude sur le paramètre de ces algorithmes. En effet, rappelons que le principe de l'optimisme face à l'incertitude repose sur l'utilisation d'intervalles de confiance, ou dans notre cas d'intervalles de crédibilité, ceux-ci contenant le critère idéal avec une certaine probabilité δ. Afin de comparer les différentes zones ou données d'entrée faisant l'objet de la décision, il est nécessaire que ce paramètre soit identiques pour chacune d'entre elles. Ce paramètre, dont la valeur peut varier entre 0 et 1, doit donc être fixé. Dans nos algorithmes, nous avons pris le parti de laisser le choix de cette valeur libre à l'utilisateur. Celui-ci est donc passé en argument de tous nos algorithmes. En effet, étant donné que l'objectif était uniquement de montrer le potentiel de la solution proposée, il nous suffisait de montrer qu'une valeur du paramètre permettait d'obtenir de bon résultats. Cependant, les performances des algorithmes peuvent varier en fonction du paramètre choisi. Dans nos expériences, l'obtention d'un paramètre adéquat passait par un recherche par quadrillage. Seulement, ceci consistait donc à tester un certain nombre de valeur et à garder celles avec lesquelles les performances obtenues était bonnes. Ceci ne convient donc pas à une utilisation pratique. Dans un premier temps, nous aimerions effectuer une étude de sensibilité aux paramètres afin de savoir si la valeur de ce dernier doit absolument être choisi avec précision ou non. Par exemple, dans le dernier chapitre, nous avons remarqué que celui-ci ne modifiait pas drastiquement les performances du moment qu'il était choisi en dessous de $\frac{1}{2}$. De plus, nous avons vu que la valeur optimale de ce paramètre n'était pas toujours la même en fonction de la base de données employée. Ainsi, nous aimerions pouvoir fixer automatiquement la valeur du paramètre en

fonction des caractéristiques du problèmes, une solution serait alors d'effectuer une analyse théorique des performances de nos algorithmes puis de choisir la valeur du paramètre théoriquement la meilleure. D'autre part, le budget d'annotations disponible est connu à l'avance. Ainsi, le paramètre peut dépendre de ce dernier. Encore une fois une analyse théorique permettrais d'exprimer le critère idéal en fonction du budget. Dans nos algorithmes, la valeur du paramètre était unique et restait constante durant tout le processus. Cependant, dans de nombreuses publications liées au principe de l'optimisme face à l'incertitude, ce paramètre est autorisé à évoluer. Il serait possible d'effectuer une modification mineure de nos algorithmes afin d'autoriser cela. Nous n'avons pas encore testé cela, mais il serait intéressant de voir quelle est la répercutions sur les résultats et si il est possible d'obtenir de meilleures performances comme cela. Ainsi, cette hypothèse pourrait aussi être sujette à d'autres expériences empiriques peuvent être menées. Toutefois, une étude théorique serait elle aussi plus efficace pour fixer la façon dont le paramètre évolue en fonction du pas de temps. Enfin, bien qu'une analyse théorique ne nous permette de trouver la valeur du paramètre que pour un budget d'annotations fini, il existe des scénarios où ce dernier est illimité mais où la vitesse d'apprentissage est importante (échantillonnage sélectif dans un flux). Nous pourrions alors tenter d'y adapter les résultats obtenus en utilisant l'astuce dite du doublage [69]. Celle-ci consiste à traiter le problème par périodes, chacune associée à un budget d'annotations doublant de taille par rapport à la précédente.

Si il nous reste du temps, nous disposons de quelques pistes d'améliorations qui pourraient être apportées aux algorithmes venant d'être introduits. Dans les premiers chapitres portant sur nos contributions, lorsqu'une partition fixe avec des zones indépendantes était utilisée, nous avons insisté sur le fait qu'un critère de sélection strictement décroissant, et donc un pseudo-risque convexe, permettait d'assurer une sélection optimale des données en connaissance parfaite. Or, dans la suite, nous avons laissé tomber cette contrainte au profit d'une solution moins couteuse en temps de calcul, à savoir l'utilisation d'un critère myope. Cependant, les résultats obtenus sur l'algorithme dérivé directement du critère de sélection en connaissance parfaite en ayant accès aux vrais paramètres de l'oracle ont permis de mettre en évidence le fait que ce dernier n'était pas optimal. En effet, nous avons vu que ses performances atteignaient un maximum puis retombaient. D'autre part, d'autres expériences ont été effectuées sans être mentionnées dans cette thèse et qui ont montré que pour un nombre d'annotations donné, un autre sous ensemble de données du réservoir que celui sélectionné par la stratégie idéale permettait d'obtenir de meilleures performances. En effet, prenons pour exemple le cas de deux données, il se peut qu'aucune des deux données appartenant à la meilleure combinaison ne donne seule le meilleur gain en performance, ainsi aucune d'entre elles ne sera sélectionnée au premier pas de temps et il sera alors impossible d'atteindre la meilleure combinaison. Ainsi, nous pensons que l'utilisation d'un critère prenant en compte l'effet à long terme de la sélection des données pourrait permettre d'améliorer les performances de nos derniers algorithmes. Deux solutions peuvent être envisagées. La première est, dans le cadre d'une partition fixe avec des zones combinées, de remplacer la décroissance du risque local de chaque zone engendrée par un échantillonnage donné par le décroissance d'un pseudo-risque local convexe comme dans les premiers chapitres. Puis de sommer les décroissances obtenues. Le problème est que ceci ne pourrait marcher que dans le cas d'une partition fixe et non pour une régression linéaire ou des Processus Gaussiens. La deuxième est de rendre convexe la décroissance du risque réel quelles que soit les données d'entrée sélectionnées dans le futur. C'est à dire que le critère de sélection courant représenterais la plus grande décroissance qu'une donnée d'entrée peut engendré que ce soit maintenant ou dans un scénario futur, ce qui s'apparenterait à de la programmation dynamique. Notez que ceci reste dans le cadre de la connaissance parfaite. Ensuite, le critère résultant devrait être associé à un intervalle de confiance en étudiant la distribution *a posteriori* de l'ensemble des

paramètres du problème. Cette solution parait extrêmement couteuse en temps de calcul et devrait, pour fonctionner en temps raisonnable, être sujette à des approximations.

A plus long terme, une problématique envisagée serait d'utiliser les travaux contenus dans ce manuscrit pour l'exploration en apprentissage par renforcement. En effet, le principe de l'optimisme face à l'incertitude est aujourd'hui couramment employé en apprentissage par renforcement pour guider la politique d'exploration. Dans ce problème, la décision sur les actions à effectuer est basée sur la fonction de valeur, qui représente la récompense à long terme qui peut être obtenue en suivant une certaine politique. La politique d'exploration consiste généralement à faire diminuer l'incertitude sur la fonction de valeur tout en tentant d'obtenir la récompense maximale. Or, seule l'action choisie nous intéresse. Dans ce cas il n'est pas utile de connaitre la fonction de valeur précisément si l'on sait que celle-ci est supérieur à celle des autres actions. Nous avons montré dans le Chapitre 6 qu'il y avait un avantage non négligeable à considérer le problème de classification plutôt que d'employer un algorithme conçu pour un problème de régression. Ici, déterminer une action est associé d'un problème de classification. Nous pensons donc que les méthodes utilisées dans cette thèse pourraient être utilisées pour guider plus précisément l'exploration en apprentissage par renforcement. La difficulté est de savoir comment inclure de même l'amélioration de la prédiction de l'action optimale dans les états accédés par la suite.

Bibliographie

[1] Dana Angluin. Queries and concept learning. *Machine learning*, 2(4) :319–342, 1988.

[2] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19) :1876–1902, 2009.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3) :235–256, 2002.

[4] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1) :48–77, 2002.

[5] Baruch Awerbuch and Robert D. Kleinberg. Adaptive routing with end-to-end feedback : Distributed learning and geometric approaches. In *Proc. of STOC*, 2004.

[6] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proc. of ICML*, 2006.

[7] Maria-Florina Balcan and Phil Long. Active and passive learning of linear separators under log-concave distributions. In *Proc. of COLT*, 2013.

[8] Eric B. Baum and Kenneth Lang. Query learning can work poorly when a human oracle is used. In *Proc. of the International Joint Conference on Neural Networks*, 1992.

[9] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proc. of ICML*, 2009.

[10] Leo Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

[11] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Proc. of ALT*, 2009.

[12] Alexandra Carpentier. *De l'échantillonage optimal en grande et petite dimension*. PhD thesis, Lille 1, 2012.

[13] Alexandra Carpentier, Alessandro Lazaric, Mohammad Ghavamzadeh, Rémi Munos, and Peter Auer. Upper-confidence-bound algorithms for active learning in multi-armed bandits. In *Proc. of ALT*, 2011.

[14] Alexandra Carpentier and Rémi Munos. Finite time analysis of stratified sampling for monte carlo. In *Proc. of NIPS*, 2011.

[15] Nicolò Cesa-Bianchi and Paul Fischer. Finite-time regret bounds for the multiarmed bandit problem. In *Proc. of ICML*, 1998.

[16] Robert W. Clayton, Thomas Heaton, Mani Chandy, Andreas Krause, Monica Kohler, Julian Bunn, Richard Guy, Michael Olson, Mathew Faulkner, MingHei Cheng, et al. Community seismic network. *Annals of Geophysics*, 54(6), 2012.

[17] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2) :201–221, 1994.

[18] David A. Cohn. Neural network exploration using optimal experiment design. In *Proc. of NIPS*, 1994.

[19] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

[20] Timothé Collet and Olivier Pietquin. Active learning for classification : An optimistic approach. In *Proc. of ADPRL*. IEEE, 2014.

[21] Timothé Collet and Olivier Pietquin. Bayesian credible intervals for online and active learning of classification trees. In *Proc. of ADPRL*. IEEE, 2015.

[22] Timothé Collet and Olivier Pietquin. Optimism in active learning. *Computational intelligence and neuroscience*, 2015, 2015.

[23] Timothé Collet and Olivier Pietquin. Optimism in active learning with gaussian processes. In *Proc. of ICONIP*. Springer, 2015.

[24] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751, 2005.

[25] Ido Dagan and Sean P. Engelson. Committee-based sampling for training probabilistic classifiers. In *Proc. of ICML*, 1995.

[26] Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Proc. of SIGKDD*, 2000.

[27] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3) :133–168, 1997.

[28] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Springer, 2001.

[29] Atsushi Fujii, Takenobu Tokunaga, Kentaro Inui, and Hozumi Tanaka. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4) :573–597, 1998.

[30] Joao Gama, Pedro Medas, and Ricardo Rocha. Forest trees for on-line data. In *Proc. of SAC*, 2004.

[31] Ravi Ganti and Alexander G. Gray. Upal : Unbiased pool based active learning. In *Proc. of AISTATS*, 2012.

[32] Ravi Ganti and Alexander G. Gray. Building bridges : Viewing active learning from the multi-armed bandit lens. In *Proc. of UAI*, 2013.

[33] Aurélien Garivier and Olivier Cappé. The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond. In *Proc. of COLT*, 2011.

[34] Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal bayesian active learning with noisy observations. In *Proc. of NIPS*, 2010.

[35] Quanquan Gu, Tong Zhang, and Jiawei Han. Batch-mode active learning via error bound minimization. In *Proc. of UAI*, 2014.

[36] Quanquan Gu, Tong Zhang, Jiawei Han, and Chris H. Ding. Selective labeling via error bound minimization. In *Proc. of NIPS*, 2012.

[37] Andrew Guillory and Jeff A. Bilmes. Label selection on graphs. In *Proc. of NIPS*, 2009.

[38] Yuhong Guo. Active instance sampling via matrix partition. In *Proc. of NIPS*, 2010.

[39] Yuhong Guo and Russell Greiner. Optimistic active-learning using mutual information. In *Proc. of IJCAI*, 2007.

[40] Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning. In *Proc. of NIPS*, 2008.

[41] Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7(2-3) :131–309, 2014.

[42] Steve Hanneke et al. Rates of convergence in active learning. *The Annals of Statistics*, 39(1) :333–361, 2011.

[43] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301) :13–30, 1963.

[44] Steven CH. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Batch mode active learning and its application to medical image classification. In *Proc. of ICML*, 2006.

[45] Steven CH. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Semi-supervised svm batch mode active learning for image retrieval. In *Proc. of CVPR*. IEEE, 2008.

[46] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. In *Proc. of NIPS*, 2010.

[47] Tzu-Kuo Huang, Alekh Agarwal, Daniel J. Hsu, John Langford, and Robert E. Schapire. Efficient and parsimonious agnostic active learning. In *Proc. of NIPS*, 2015.

[48] Rebecca Hwa. Sample selection for statistical parsing. *Computational linguistics*, 30(3) :253–276, 2004.

[49] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *Proc. of ICCV*. IEEE, 2007.

[50] Ashish Kapoor, Eric Horvitz, and Sumit Basu. Selective supervision : Guiding supervised learning with decision-theoretic active learning. In *Proc. of IJCAI*, 2007.

[51] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *Proc. of AISTATS*, 2012.

[52] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes : Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9 :235–284, 2008.

[53] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1) :4–22, 1985.

[54] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proc. of ICML*, 1994.

[55] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proc. of SIGIR*, 1994.

[56] M. Lichman. UCI machine learning repository, 2013.

[57] Michael Lindenbaum, Shaul Markovitch, and Dmitry Rusakov. Selective sampling for nearest neighbor classifiers. *Machine learning*, 54(2) :125–152, 2004.

[58] Yifei Ma, Tzu-Kuo Huang, and Jeff Schneider. Active search and bandits on graphs using sigma-optimality. In *Proc. of UAI*, 2015.

[59] Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. A finite-time analysis of multi-armed bandits problems with kullback-leibler divergences. In *Proc. of COLT*, 2011.

[60] Andrew Kachites McCallumzy and Kamal Nigamy. Employing em and pool-based active learning for text classification. In *Proc. of ICML*, 1998.

[61] Robert Nowak. Generalized binary search. In *Proc. of the Allerton Conference on Communications, Control, and Computing*, pages 568–574, 2008.

[62] Robert Nowak. Noisy generalized binary search. In *Proc. of NIPS*, 2009.

[63] Thomas Osugi, Deng Kim, and Stephen Scott. Balancing exploration and exploitation : A new algorithm for active machine learning. In *Proc. of ICDM*, 2005.

[64] Herbert Robbins et al. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5) :527–535, 1952.

[65] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. In *Proc. of ICML*, 2001.

[66] Burr Settles. Active learning literature survey. Computer Science Technical Report 1648, University of Wisconsin–Madison, 2009.

[67] Burr Settles and Mark Craven. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *Proc. of EMNLP*, pages 1069–1078. ACL, 2008.

[68] H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proc. of COLT*, 1992.

[69] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2) :107–194, 2011.

[70] CE Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 :379–423, 623–656, 1948.

[71] Niranjan Srinivas, Andreas Krause, Matthias Seeger, and Sham M. Kakade. Gaussian process optimization in the bandit setting : No regret and experimental design. In *Proc. of ICML*, 2010.

[72] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning : An introduction*. The MIT Press, Cambridge, MA, 1998.

[73] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933.

[74] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2 :45–66, 2002.

[75] Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *Proc. of ICML*, 2006.

[76] Chicheng Zhang and Kamalika Chaudhuri. Beyond disagreement-based agnostic active learning. In *Proc. of NIPS*, 2014.

[77] Tong Zhang and F. Oles. The value of unlabeled data for classification problems. In *Proc. of ICML*, 2000.

[78] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003.