Group Variable Selection via $\ell_{p,0}$ regularization and application to Optimal Scoring

Duy Nhat Phan[a,b], Hoai An Le Thi[c,*]

[a] *Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam*
[b] *Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam*
[c] *Université de Lorraine, LGIPM, F-57000 Metz, France*

Abstract

The need to select groups of variables arises in many statistical modeling problems and applications. In this paper, we consider the $\ell_{p,0}$-norm regularization for enforcing group sparsity and investigate a DC (Difference of Convex functions) approximation approach for solving the $\ell_{p,0}$-norm regularization problem. We show that, with suitable parameters, the original and approximate problems are equivalent. Considering two equivalent formulations of the approximate problem we develop DC programming and DCA (DC Algorithm) for solving them. As an application, we implement the proposed algorithms for group variable selection in the optimal scoring problem. The sparsity is obtained by using the $\ell_{p,0}$-regularization that selects the same features in all discriminant vectors. The resulting sparse discriminant vectors provide a more interpretable low-dimensional representation of data. The experimental results on both simulated datasets and real datasets indicate the efficiency of the proposed algorithms.

Keywords: Group variable selection, $\ell_{p,0}$ regularization, DC approximation, DC programming, DCA, Optimal scoring

*Corresponding author
Email addresses: phanduynhat@tdtu.edu.vn (Duy Nhat Phan), hoai-an.le-thi@univ-lorraine.fr (Hoai An Le Thi)

1. Introduction

Variable selection plays an important role in many applications and has drawn increased attention from many researchers. In the literature, the use of sparsity-inducing norms is a powerful technique for variable selection in the high-dimensional settings. In this direction, the ℓ_0-norm has been studied extensively on both theoretical and practical aspects for individual variable selection in many practical problems. However, when the data possesses certain group structures, we are naturally interested in selecting important groups of variables rather than individual ones. For instance, in multi-factor analysis of variance, a factor with several levels may be expressed through a group of dummy variables. In nonparametric additive regression, each component can be represented by a linear combination of a set of basis functions. In genomic data analysis, the correlations between genes sharing the biological pathway can be high. Hence these genes should be considered as a group. In such cases, the selection of important factors/nonparametric components/groups of genes amounts to the selection of groups of variables. In recent years, there are many works based on regularization methods for group variable selection in various application domains such as machine learning, statistics, computational biology, signal processing, and other related areas. In this paper, we introduce a natural approach for enforcing group sparsity by using the $\ell_{p,0}$-regularization with $p \geq 1$.

We define the step function $s : \mathbb{R} \to \mathbb{R}$ by $s(t) = 1$ if $t \neq 0$ and $s(t) = 0$ otherwise. Assume that $x = (x_1, ..., x_d) \in \mathbb{R}^d$ is partitioned into J non-overlapping groups $x^1, ..., x^J$, then the $\ell_{p,0}$-norm of x is defined by

$$\|x\|_{p,0} = \sum_{j=1}^{J} s(\|x^j\|_p).$$

The $\ell_{p,0}$-regularization problem takes the form:

$$\min \left\{ f(x, y) + \lambda \|x\|_{p,0} : (x, y) \in K \subset \mathbb{R}^d \times \mathbb{R}^m \right\}, \tag{1}$$

where f is the loss function related to the learning task, and λ is a nonnegative tuning parameter that makes the trade-off between the criterion f and the sparsity.

Let us mention some important applications of group variable selection corresponding to the model (1).

Group variable selection in linear regression: Given n observations (x_i, y_i), $i = 1, ..., n$, where y_i is the response variable, $x_i = (x_{i1}^T, ..., x_{iJ}^T)^T$ is the corresponding covariates with J groups of predictors and x_{ij} is the d_j-dimensional sub-covariate vector, $j = 1, ..., J$. Let $X_j = [x_{1j}; ...; x_{nj}]$ be the $n \times d_j$ design matrix corresponding to the j-th group and β^j be the vector of the regression coefficients in the j-th group. The problem of selecting the important covariates and estimating the corresponding coefficient vector takes the form of (1):

$$\min \left\{ \frac{1}{2n} \|y - \sum_{j=1}^{J} X_j \beta^j\|_2^2 + \lambda \sum_{j=1}^{J} s(\|\beta^j\|_p) \right\}. \tag{2}$$

Multi-task feature selection: Let T be the number of tasks. For the j-th task, the training set $\mathcal{D}_j$ consists of n_j labeled data points in the form of ordered pairs $(x_i^j, y_i^j), i = 1, ..., n_j$, with $x_i^j \in \mathbb{R}^d$ and its corresponding output $y_i^j \in \mathbb{R}$. Multi-task learning aims to estimate T functions $f_j(x) : \mathbb{R}^d \to \mathbb{R}, j = 1, ..., T$, which well fit the data and are statistically predictive. Here, we focus on linear functions, i.e., $f_j(x) = W_{:j}^T x + b_j$. The multi-task feature selection problem takes the form of (1):

$$\min \left\{ \sum_{j=1}^{T} \sum_{i=1}^{n_j} \mathcal{L}(y_i^j, W_{:j}^T x_i^j + b_j) + \lambda \| |W\|_{p,0} \right\}, \tag{3}$$

where $\mathcal{L}$ denotes the loss function, W is the $d \times T$ matrix whose columns are $W_{:1}, ..., W_{:T}$ and $\|W\|_{p,0}$ denotes the $\ell_{p,0}$-norm of the matrix W, i.e., $\|W\|_{p,0} = \sum_{j=1}^{d} s(\|W_{j:}\|_p)$ with $W_{j:}$ being the j-th row of W. In this problem, each row of W is regarded as a group. Note that the group variable selection problem in multiclass support vector machines (SVMs) is a special case of the problem (3) where $\mathcal{L}$ is the hinge loss function given by (Obozinski et al., 2006)

$$\mathcal{L}(y_i^j, W_{:j}^T x_i^j + b_j) = \max\left(1 - y_i^j (W_{:j}^T x_i^j + b_j), 0\right).$$

Group sparse principal component analysis (PCA): Let $X \in \mathbb{R}^{n \times d}$ be a data matrix which comprises n observations $x_i \in \mathbb{R}^d$, where d is the number

of features. We assume that the features have been centered to have mean 0. Denote by I_k the $k \times k$ identity matrix. Zou et al. (2006) has transformed the PCA problem into a regression type optimization problem.

$$\min_{A \in \mathbb{R}^{p \times k}} \left\{ \|X - XAA^T\|_F^2 : A^T A = I_k \right\}. \tag{4}$$

The columns of the solution to the problem (4) are referred as the first k loading vectors of PCA. One way to obtain sparse loading vectors is imposing the $\ell_{p,0}$ penalty on the regression coefficients.

$$\min_{A,B \in \mathbb{R}^{d \times k}} \left\{ \|X - XBA^T\|_F^2 + \lambda \|B\|_{p,0} : A^T A = I_k \right\}, \tag{5}$$

where the columns $\beta_1, ..., \beta_k$ of B correspond to the required sparse loading vectors. Note that if $\lambda = 0$, the problem (5) reduces to finding the ordinary loading vectors, i.e., B is proportional to the solution of (4) (see Zou et al. (2006)). When $\lambda > 0$, some rows of B are forced to zero, we get the sparse loading vectors.

Group sparse Fisher linear discriminant analysis (LDA): Let $\{(x_i, y_i) : i = 1, ..., n\}$ be a set of labeled training data with observation vector $x_i \in \mathbb{R}^d$ and label $y_i \in \{1, ..., C\}$. The original LDA formulation is known as the Fisher linear discriminant analysis (Fisher, 1936). Fisher criterion aims to find a linear transformation $W \in \mathbb{R}^{d \times L}$ that maps the data in the d-dimensional space to the L-dimensional space ($L \leq C - 1$), in which the between-class variance is maximized while the within-class variance is minimized, i.e.,

$$\max_{W \in \mathbb{R}^{d \times L}} \left\{ \text{tr}\left((W^T \Sigma_w W)^{-1} (W^T \Sigma_b W) \right) \right\}, \tag{6}$$

where Σ_b and Σ_w are the between-class covariance matrix and the within-class covariance matrix, respectively. We use the $\ell_{p,0}$-regularization to select the same features in all discriminant vectors $W_{:1}, ..., W_{:L}$ which are the columns of W. The resulting sparse discriminant vectors can provide a more interpretable low-dimensional representation of data. The regularization problem takes the form of (1):

$$\min_{W \in \mathbb{R}^{d \times L}} \left\{ -\text{tr}\left((W^T \Sigma_w W)^{-1} (W^T \Sigma_b W) \right) + \lambda \|W\|_{p,0} \right\}. \tag{7}$$

Joint sparse compressed sensing: Compressed sensing aims to recover the sparse signal w from a measurement vector $b = Aw$ for a given matrix A. Compressed sensing can be extended to the multiple measurement vector in which the signals are represented as a set of jointly sparse vectors sharing a common set of the nonzero elements (Cotter et al., 2005; Chen & Huo, 2006; Sun et al., 2009). Joint compressed sensing considers the reconstruction of the signal represented by a matrix W, which is given by a dictionary A and a multiple measurement vector B such that $B = AW$. Since there usually exists noise in the data, the joint sparse compressed sensing can be formulated as the sparse optimization problem of the form (1):

$$\min_{W} \left\{ \|AW - B\|_F^2 + \lambda \|W\|_{p,0} \right\}. \tag{8}$$

Other applications of group variable selection include multiple graphical models (Danaher et al., 2014), multi-task reinforcement learning (Calandriello et al., 2014), etc.

Existing works considered group variable selection as a natural extension of variable selection. The first approach, named the group Lasso (Yuan & Lin, 2006), is closely connected to the Lasso (ℓ_1-norm) approximation of the ℓ_0-norm. Works in this direction include the $\ell_{\infty,1}$-norm and the $\ell_{2,1}$-norm which were widely used for group variable selection in linear regression (Turlach et al., 2005), multi-task learning (Quattoni et al., 2009; Argyriou et al., 2008; Bi et al., 2008; Liu et al., 2009; Obozinski et al., 2006, 2010; Zhang et al., 2010; Nie et al., 2010; Lan et al., 2015), multiclass SVMs (Zhang et al., 2008; Blodel et al., 2013), PCA (Kha et al., 2015), Fisher LDA (Gu et al., 2011), optimal scoring (Leng, 2008; Merchante et al., 2012), and compressed sensing (Sun et al., 2009). In general, these convex regularization methods are not efficient for the group variable selection purpose (Wei & Huang, 2010). The second approach deals with nonconvex approximation and has been developed for the $\ell_{2,0}$-norm. More precisely, nonconvex approximation approaches based on the smoothly clipped absolute deviation penalty (SCAD) (Fan & Li, 2001) and the minimax concave penalty (MCP) (Zhang, 2010) have been studied for the $\ell_{2,0}$-norm in linear

regression problems (see e.g. (Lee et al., 2016; Huang & Wei, 2012; Wang et al., 2007; Wei & Zhu, 2012)). These works have proved that these nonconvex approximation approaches for the $\ell_{2,0}$-norm are more efficient than the methods using the $\ell_{2,1}$-norm. Recently, the $\ell_{p,0}$ (resp. $\ell_{p,q}$ with $p \geq 1, 0 \leq q \leq 1$) regularization has been considered in Eksioglu (2014) (resp. Hu et al. (2017)) for a very special convex least squares loss function. Eksioglu (2014) addressed the following problem

$$\min_{x\in\mathbb{R}^d} \left\{ \frac{1}{2}\sum_{i=0}^{n} \gamma^{n-i}|y_i - x^T a_i|^2 + \lambda\|x\|_{p,0} \right\}. \tag{9}$$

By using the exponential approximation of the step function, the author proposed an inexact method via approximate subgradients for solving the resulting approximate problem

$$\min_{x\in\mathbb{R}^d} \left\{ \frac{1}{2}\sum_{i=0}^{n} \gamma^{n-i}|y_i - x^T a_i|^2 + \lambda\sum_{j=1}^{J}(1 - e^{-\alpha\|x^{(j)}\|_p}) \right\}. \tag{10}$$

This method requires the differentiability of the approximate term, and its convergence analysis was not provided. Later, Hu et al. (2017) considered the following problem

$$\min_{x\in\mathbb{R}^d} \left\{ \|Ax - b\|^2 + \lambda\|x\|_{p,q}^q \right\}, \tag{11}$$

with $p \geq 1, 0 \leq q \leq 1$, by the proximal gradient method (PGM) which iteratively computed the proximal operator of the $\ell_{p,q}$ norm:

$$x^{l+1} \in \arg\min_{x\in\mathbb{R}^d} \left\{ \frac{1}{2\rho}\|x - (x^l - 2\rho A^T(Ax^l - b)\|^2 + \lambda\|x\|_{p,q}^q \right\}. \tag{12}$$

The limitation of this method is to require the computation of proximal operators of discontinuous functions which may be very expensive or impossible when the loss function is nonconvex and nondifferentiable.

In this paper, we investigate DC (Difference of Convex functions) approximation approaches for the general case, i.e., the $\ell_{p,0}$-norm with $p \geq 1$. We consider the problem (1), where K is a compact polyhedral convex set in $\mathbb{R}^d \times \mathbb{R}^m$ and f is a finite DC function on $\mathbb{R}^d \times \mathbb{R}^m$. The paper makes the following contributions.

Firstly, we approximate the $\ell_{p,0}$-norm by a DC function via the Capped-ℓ_1 approximation of the step function. The Capped-ℓ_1 has been proved theoretically in Le Thi et al. (2015) to be the tightest DC approximation for the ℓ_0-norm case. We extend the interesting results concerning the ℓ_0-norm obtained in Le Thi et al. (2015) to the $\ell_{p,0}$-norm. We prove that, with suitable parameters, the nonconvex approximate problem is equivalent to the original problem (1). This result gives an important mathematical foundation for our approximation method. It is worth noting that our $\ell_{p,0}$ approximation is not a simple DC function, but rather a composite function of a DC function and a convex function. This is the main difference that causes more difficulties when moving from individual variable selection to group variable selection. From a theoretical point of view, the link between the original problem and its approximate problem is not evident. Moreover, from an algorithmic point of view, the $\ell_{p,0}$ approximate regularization is not separable in their variables, so the minimization problems involving this regularization are more challenging than the ones related to the ℓ_0-norm.

Secondly, we develop solution methods based on DC programming and DCA (DC Algorithms), a powerful technique in nonconvex optimization (Le Thi & Pham Dinh, 2005; Pham Dinh & Le Thi, 1997), for solving the approximate problem. Although DCA is a standard approach, how to design an efficient DCA scheme for a concrete problem is still a challenge in nonconvex programming framework, since the general DCA scheme is rather a philosophy than an algorithm. The elaboration of suitable DC decompositions such that the convex subproblems can be efficiently solved, and the effective techniques for convex subproblems are crucial questions to be studied while using DCA. Considering two equivalent formulations of the approximate problem we propose two DCA schemes which can be viewed as an $\ell_{p,1}$-perturbed algorithm and a reweighted-$\ell_{p,1}$ algorithm. When $p = 1$ and $p = 2$, our algorithms include $\ell_{2,1}$-perturbed algorithm, reweighted-$\ell_{2,1}$ algorithm, ℓ_1-perturbed algorithm, and reweighted-ℓ_1 algorithm with different weights on groups and the same weight on each group.

Finally, as an application, we consider the problem of group variable selection

in optimal scoring and develop four DCA schemes for solving it. Thanks to suitable DC decompositions, the convex subproblems in these algorithms can be solved by (block) coordinate descent methods. Especially, in the two out of four DCAs, the convex subproblem can be decomposed into separable smaller sub-problems and solved in parallel. We provide special convergence properties of the proposed algorithms and perform a careful empirical experiment to study their performance.

Among $\ell_{p,0}$ approximate regularizations, we show that the approximate $\ell_{1,0}$-regularization is the most interesting with several useful properties from both theoretical and computational aspects. More precisely, the $\ell_{1,0}$ approximate regularization is the closest approximation of the $\ell_{p,0}$-regularizations (Section 2). In the optimal scoring application, the DCA schemes for solving the resulting approximate problem iteratively solve an ℓ_1-perturbed/reweighted-ℓ_1 problem which can be separated into independent sub-problems. This interesting feature makes our proposed approach very efficient in terms of computational complexity. Moreover, these DCA schemes converge after a finite number of iterations.

The superiority of our work to existing approaches for $\ell_{p,0}$ regularization (Eksioglu (2014), Hu et al. (2017)) are multiple. Firstly, while these works deal only with a very special loss function which is convex differentiable, we consider a general and large class of problems where the loss function is nonconvex and nondifferentiable (DC). Secondly, we can prove the equivalent between the original and approximate problems. Thirdly, the algorithm in Eksioglu (2014) requires the differentiability of the approximation term (in addition to the differentiability and the convexity of the loss function) while DCA works on nonconvex and nondifferentiable functions. Fourthly, the proposed DCAs enjoy several interesting convergence properties such as the finite convergence, the local optimality of the solutions, whereas no convergence property of the algorithm is available in Eksioglu (2014) and no convergence property was proved in Hu et al. (2017) for the $\ell_{p,0}$ regularization. Last but not least, the computational efficiency of DCA in solving convex subproblems by parallelization is a significant advantage (once again we note that the computation of proxi-

mal operators of discontinuous functions in the algorithm proposed in Hu et al. (2017) is very expensive or impossible when the loss function is nonconvex and nondifferentiable).

The rest of the paper is organized as follows. In Section 2, we present the approximate problems and show that these problems are equivalent to the original problem. We first give in Section 3 a brief introduction of DC programming and DCA for general DC programs and then show how to apply DCA to solve the approximate problems. The application of the proposed algorithms for group variable selection in optimal scoring is described in Section 4. The numerical experiments are reported in Section 5 and Section 6 concludes the paper.

Throughout the paper, for vectors $u, v \in \mathbb{R}^n$, the inner product of u and v is defined as $\langle u, v\rangle = \sum_{i=1}^n u_i v_i$. For every $p \geq 1$, the ℓ_p-norm of vector u is $\|u\|_p = (\sum_{i=1}^n |u_i|^p)^{\frac{1}{p}}$. In addition, if u is partitioned into J non-overlapping groups $u^1, ..., u^J$, we recall that the $\ell_{p,0}$-norm of u is defined as

$$\|u\|_{p,0} = \sum_{j=1}^{J} s(\|u^j\|_p),$$

where s is the step function defined above. Similarly, the $\ell_{p,1}$-norm of u is defined as

$$\|u\|_{p,1} = \sum_{j=1}^{J} \|u^j\|_p.$$

We denote the vector $(\|u^1\|_p, ..., \|u^J\|_p)$ by $|u_J|_p$. For $A \in \mathbb{R}^{n\times m}$, the Frobenius norm of A is given by $||A||_F = \sqrt{\sum_{i,j} A_{ij}^2}$.

2. DC approximate problems and the link with the original problem

By using the Capped-ℓ_1 function (Peleg & Meir, 2008), the approximate $\ell_{p,0}$-regularization of the $\ell_{p,0}$ is given by

$$\|x\|_{p,0} \approx \sum_{j=1}^{J} \eta_\alpha(\|x^j\|_p), \tag{13}$$

where $\eta_\alpha(t) = \min\{1, \alpha|t|\}$ and α is a tuning parameter such that $\eta_\alpha(t)$ approximates the step function $s(t)$ as α tends to $+\infty$. Since $\|x^j\|_\infty \leq \|x^j\|_p \leq \|x^j\|_1$

$\forall p \geq 1$ and η_α is increasing on $[0; +\infty)$, we get

$$\eta_\alpha(\|x^j\|_\infty) \leq \eta_\alpha(\|x^j\|_p) \leq \eta_\alpha(\|x^j\|_1) \leq s(\|x^j\|_p). \tag{14}$$

This shows that, with the same parameter α, the approximate $\ell_{1,0}$-norm of x is the closest to $\|x\|_{p,0}$.

We also note that, although $\|x\|_{p,0}$ does not change for any $p \geq 1$, the approximation of $\|x\|_{p,0}$ is different when p varies, and therefore they have the different property.

By using the approximation (13), the resulting approximate problem of (1) takes the form

$$\min \left\{ F_\alpha^p(x, y) := f(x, y) + \lambda \sum_{j=1}^{J} \eta_\alpha(\|x^j\|_p) : (x, y) \in K \right\}. \tag{15}$$

We also consider another equivalent form of the problem (15) as follows:

$$\min \left\{ F_\alpha(x, y, z) := f(x, y) + \lambda \sum_{j=1}^{J} \eta_\alpha(z_j) : (x, y, z) \in K_p \right\}, \tag{16}$$

where $K_p = \{(x, y, z) : (x, y) \in K, \|x^j\|_p \leq z_j, j = 1, ..., J\}$. Indeed, the problems (15) and (16) are equivalent in the following sense.

Proposition 1. *A point $(x^*, y^*) \in K$ is a global (resp. local) solution of the problem* (15) *if and only if $(x^*, y^*, |x_J^*|_p)$ is a global (resp. local) solution to the problem* (16)*. Moreover, if (x^*, y^*, z^*) is a global solution to* (16) *then (x^*, y^*) is a global solution to* (15)*.*

Proof. Since η_α is an increasing function on $[0, +\infty)$, we have

$$F_\alpha(x, y, z) \geq F_\alpha(x, y, |x_J|_p) = F_\alpha^p(x, y) \ \forall (x, y, z) \in K_p.$$

Then the conclusion on global solutions is trivial. The result on local solutions can be deduced from the following remarks. Let $a = \max_j \sqrt{d_j} \geq 1$, where d_j is the number of variables of the j-th group. If $1 \leq p < 2$, we have

$$|\|x^j\|_p - \|(x^j)^*\|_p| \leq \|x^j - (x^j)^*\|_p \leq d_j^{1/p-1/2} \|x^j - (x^j)^*\|_2. \tag{17}$$

Combining $d_j^{1/p-1/2} \leq a$ and (17), we obtain

$$|\|x^j\|_p - \|(x^j)^*\|_p| \leq a\|x^j - (x^j)^*\|_2 \ \forall j = 1, ..., J. \tag{18}$$

If $p \geq 2$, we have

$$|\|x^j\|_p - \|(x^j)^*\|_p| \leq \|x^j - (x^j)^*\|_p \leq \|x^j - (x^j)^*\|_2. \tag{19}$$

From (18)-(19) and $a \geq 1$, we have $\||x_J|_p - |x_J^*|_p\|_2 \leq a\|x - x^*\|_2 \ \forall p \geq 1$. Therefore, if $(x, y) \in B((x^*, y^*), \delta/\sqrt{a^2+1})$ then $(x, y, |x_J|_p) \in B((x^*, y^*, |x^*|_p), \delta)$. Moreover, if $(x, y, z) \in B((x^*, y^*, z^*), \delta)$ then $(x, y) \in B((x^*, y^*), \delta)$. The proof is then complete. □

Note that K is a compact polyhedral convex set. Hence, by Proposition 1, without the generality, we can make the following assumption.

Assumption 1. *K_p is a compact convex set.*

Proposition 2. *Under the assumption 1, with $p = +\infty$, there exists α_0 such that the approximate problem* (15) *is equivalent to the original problem* (1) *for all $\alpha > \alpha_0$.*

Proof. Since the ℓ_∞-norm is polyhedral convex function, $K_\infty = \{(x, y, z) : (x, y) \in K, \|x^j\|_\infty \leq z_j, j = 1, ..., J\}$ is also a compact polyhedral convex set. We notice that the problem (16) is an approximate of the problem below including the ℓ_0 penalty on the vector z.

$$\min \left\{ f(x, y) + \lambda \sum_{j=1}^{J} s(z_j) : (x, y, z) \in K_\infty \right\}. \tag{20}$$

Le Thi et al. (2015) has shown that there exists α_0 such that the problems (16) and (20) are equivalent for all $\alpha > \alpha_0$. Moreover, we have

$$f(x, y) + \lambda \sum_{j=1}^{J} s(z_j) \geq f(x, y) + \lambda \sum_{j=1}^{J} s(\|x^j\|_\infty) \ \forall (x, y, z) \in K_\infty.$$

It easily follows that the problems (20) and (1) are equivalent. By Proposition 1, we have the equivalence between the problems (16) and (15). Hence the approximate problem (15) is equivalent to the original problem (1). Proposition 2 is then proved. □

For $p = +\infty$, we have proved that the approximate problem (15) is equivalent to the problem (1) for all $\alpha > \alpha_0$. We now show the general result for all $p \geq 1$.

Proposition 3. *Under the assumption 1, there exists α_0 such that the approximate problem* (15) *is equivalent to the original problem* (1) *for all $\alpha > \alpha_0$ and $p \geq 1$.*

Proof. From the inequality (14), we have

$$\eta_\alpha(\|x^j\|_\infty) \leq \eta_\alpha(\|x^j\|_p) \leq s(\|x^j\|_p). \tag{21}$$

Hence we obtain

$$F_\alpha^\infty(x,y) \leq F_\alpha^p(x,y) \leq f(x,y) + \lambda \sum_{j=1}^{J} s(\|x^j\|_\infty) \ \forall (x,y) \in K. \tag{22}$$

For $p = +\infty$, by Proposition 2, the problems (15) and (1) are equivalent for all $\alpha > \alpha_0$. Hence let (x^*, y^*) be a common optimal solution, we have

$$\begin{aligned} F_\alpha^\infty(x^*,y^*) = F_\alpha^p(x^*,y^*) &= f(x^*,y^*) + \lambda \sum_{j=1}^{J} s(\|(x^j)^*\|_\infty) \\ &= f(x^*,y^*) + \lambda \sum_{j=1}^{J} s(\|(x^j)^*\|_p). \end{aligned}$$

Combining this and (22), for all $(x,y) \in K$, we have

$$\begin{aligned} f(x^*,y^*) + \lambda \sum_{j=1}^{J} s(\|(x^j)^*\|_p) = F_\alpha^p(x^*,y^*) &\leq F_\alpha^p(x,y) \\ &\leq f(x,y) + \lambda \sum_{j=1}^{J} s(\|x^j\|_p). \end{aligned}$$

Therefore, we can deduce that the approximate problem (15) is equivalent to the original problem (1). This completes the proof of Proposition 3. □

We are now going to develop DCA based algorithms for solving the nonconvex approximate problems (15) and (16).

3. Solution methods via DC programming and DCA

Before discussing DCA based algorithms for solving the problems (15) and (16), let us introduce briefly DC programming and DCA.

3.1. A brief introduction of DC programming and DCA

DC programming and DCA (see (Le Thi & Pham Dinh, 2005; Pham Dinh & Le Thi, 1997, 1998, 2014; Le Thi & Pham Dinh, 2018) and more references in Le Thi (Homepage)) constitute the backbone of smooth/nonsmooth nonconvex programming and global optimization. They address the problem of minimizing a DC function on the whole space $\mathbb{R}^n$ or on a closed convex set $C \in \mathbb{R}^n$. Generally speaking, a standard DC program takes the form:

$$\beta = \inf\{F(x) := G(x) - H(x) \,|\, x \in \mathbb{R}^n\} \quad (P_{dc}),$$

where G, H are lower semi-continuous proper convex functions on $\mathbb{R}^n$. Such a function F is called a DC function, and $G-H$ a DC decomposition of F while G and H are the DC components of F. When either G or H is a polyhedral convex function (say, a pointwise supremum of a finite collection of affine functions) (P_{dc}) is called a polyhedral DC program.

A convex constraint $x \in C$ can be incorporated in the objective function F by using the indicator function on C, denoted χ_C, which is defined by $\chi_C(x) = 0$ if $x \in C$, $+\infty$ otherwise. By the way, any convex constrained DC program can be written in the standard form (P_{dc}) by adding the indicator function of the convex constraint set to the first DC component G:

$$\inf\{F(x) := G(x) - H(x) : x \in C\ \} = \inf\{\chi_C(x) + G(x) - H(x) : x \in \mathbb{R}^n\}.$$

For a convex function θ, the subdifferential of θ at $x_0 \in \text{dom}\theta := \{x \in \mathbb{R}^n : \theta(x_0) < +\infty\}$, denoted by $\partial\theta(x_0)$, is defined by

$$\partial\theta(x_0) := \{y \in \mathbb{R}^n : \theta(x) \geq \theta(x_0) + \langle x - x_0, y\rangle, \forall x \in \mathbb{R}^n\}.$$

The subdifferential $\partial\theta(x_0)$ generalizes the derivative in the sense that θ is differentiable at x_0 if and only if $\partial\theta(x_0) \equiv \{\nabla_x\theta(x_0)\}$.

A point x^* is called a *critical point* of $G-H$, or a generalized Karush-Kuhn-Tucker point (KKT) of (P_{dc})) if

$$\partial H(x^*) \cap \partial G(x^*) \neq \emptyset. \tag{23}$$

The main idea of DCA is simple: each iteration l of DCA approximates the concave part $-H$ by its affine majorization (that corresponds to taking $y^l \in \partial H(x^l)$) and minimizes the resulting convex function.

Generic DCA scheme

Initialization: Let $x^0 \in \mathbb{R}^n$ be an initial guess, $l \leftarrow 0$.

Repeat

- Calculate $y^l \in \partial H(x^l)$
- Calculate $x^{l+1} \in \arg\min\{G(x) - \langle x, y^l \rangle : x \in \mathbb{R}^n\} \quad (P_l)$
- $l \leftarrow l + 1$

Until convergence of $\{x^l\}$.

The sequence $\{x^l\}$ generated by DCA enjoys the following properties (Le Thi & Pham Dinh, 2005; Pham Dinh & Le Thi, 1997):

(i) The sequence $\{G(x^l) - H(x^l)\}$ is decreasing, and DCA is a descent method without line search;

(ii) If $G(x^{l+1}) - H(x^{l+1}) = G(x^l) - H(x^l)$, then x^l is a critical point of $G - H$. In such a case, DCA terminates at l-th iteration. Otherwise, if the optimal value α of problem (P_{dc}) is finite and the infinite sequences $\{x^l\}$ is bounded then every limit point x^* of the sequences $\{x^l\}$ is a critical point of $G - H$. Thus DCA has a global convergence (i.e. DCA converges from an arbitrary initial point).

(iii) For polyhedral DC programs $\{x^l\}$ contains finitely many elements and DCA converges after a finite number of iterations.

For a complete study of DC programming and DCA the reader is referred to (Le Thi & Pham Dinh, 2005; Pham Dinh & Le Thi, 1997, 1998, 2014) and the references therein. It is worth pointing out that most of existing methods in convex/nonconvex programming are special versions of DCA via appropriate DC decompositions (see Le Thi & Pham Dinh (2018)). In recent years, numerous DCA based algorithms have been developed for successfully solving large-scale nonsmooth/nonconvex programs appearing in several application areas, espe-

cially in machine learning, communication system, biology, finance, etc. DCA has been proved to be a fast and scalable approach which is, thanks to the effect of DC decompositions, more efficient than related methods. For a comprehensible survey on thirty years of development of DCA, the reader is referred to the recent paper Le Thi & Pham Dinh (2018).

3.2. DCA for solving the first approximate problem

Firstly, we consider the approximate problem (15) and introduce a DCA scheme that includes algorithms of ℓ_1-perturbed/$\ell_{2,1}$-perturbed types. In the following proposition we show that the approximate $\ell_{p,0}$-regularization is a DC function.

Proposition 4. *The approximate $\ell_{p,0}$-regularization given by* (13) *is a DC function, with the DC decomposition $\sum_{j=1}^{J} \eta_\alpha(\|x^j\|_p) = \alpha\|x\|_{p,1} - \phi(x)$, where $\phi(x) = \sum_{j=1}^{J}(-1 + \max\{1, \alpha\|x^j\|_p\})$.*

Proof. Obviously $\alpha\|x\|_{p,1}$ is convex. We only need to show that ϕ is convex. Since the sum of convex functions is also convex, it is sufficient to show that the function $\varphi : \mathbb{R}^m \to \mathbb{R}$ given by $\varphi(y) = \max\{1, \alpha\|y\|_p\}$ is convex.
For any $\theta \in [0, 1]$ and $y, z \in \mathbb{R}^m$ we have

$$\begin{aligned}
\varphi(\theta y + (1-\theta)z) &= \max\{1, \alpha\|\theta y + (1-\theta)z\|_p\} \\
&\leq \max\{1, \alpha\theta\|y\|_p + \alpha(1-\theta)\|z\|_p\} \\
&= \max\{\theta + (1-\theta), \theta\alpha\|y\|_p + (1-\theta)\alpha\|z\|_p\} \\
&\leq \theta \max\{1, \alpha\|y\|_p\} + (1-\theta)\max\{1, \alpha\|z\|_p\} \\
&= \theta\varphi(y) + (1-\theta)\varphi(z)\}.
\end{aligned}$$

Hence φ is convex. □

Now, let g and h be DC components of the DC function f, say $f = g - h$. From Proposition 4, the objective function of the problem (15) can be written as a DC function:

$$F_\alpha^p(x, y) = G_1(x, y) - H_1(x, y), \tag{24}$$

where

$$G_1(x,y) = \chi_K(x,y) + g(x,y) + \lambda\alpha\|x\|_{p,1}$$
$$H_1(x,y) = h(x,y) + \lambda\sum_{j=1}^{J}(-1 + \max\{1, \alpha\|x^j\|_p\}),$$

Hence the objective function of the problem (15) can be rewritten as a DC function:

$$F_\alpha^p(x,y) = G_1(x,y) - H_1(x,y), \tag{25}$$

where

$$G_1(x,y) = \chi_K(x,y) + g(x,y) + \lambda\alpha\|x\|_{p,1}$$
$$H_1(x,y) = h(x,y) + \lambda\sum_{j=1}^{J} r(\|x^j\|_p),$$

and g, h are DC components of f, i.e., $f = g - h$. Hence a DC formulation of the problem (15) takes the form

$$\min_{(x,y)}\{G_1(x,y) - H_1(x,y)\}. \tag{26}$$

Following the generic DCA scheme, DCA for solving the problem (26) can be described as follows.

DCA1

Initialization: Choose an initial point (x^0, y^0), a tolerance τ and $l \leftarrow 0$.

repeat

1. Compute $(\bar{x}^l, \bar{y}^l) \in \partial h(x^l, y^l)$ and v^l by

$$v_j^l = \begin{cases} \lambda\alpha\partial\|(x^j)^l\|_p & \text{if} \quad \|(x^j)^l\|_p > 1/\alpha \\ 0 & \text{otherwise.} \end{cases}$$

2. Compute (x^{l+1}, y^{l+1}) by solving the problem:

$$\min_{(x,y)\in K}\left\{g(x,y) - \langle\bar{x}^l, x\rangle - \langle\bar{y}^l, y\rangle + \lambda\alpha\|x\|_{p,1} - \langle v^l, x\rangle\right\}. \tag{27}$$

3. $l \leftarrow l + 1$.

until Stopping criterion

Remark 1. *We see that the problem* (27) *has the form of an* $\ell_{p,1}$*-perturbed problem covering special cases such as the* $\ell_{2,1}$*-perturbed and* ℓ_1*-perturbed problem which can be found in many previous works (see e.g. (Le Thi et al., 2008, 2014a, 2015; Ong & Le Thi, 2013b)). Especially,* $p = 1$ *we have* $\|x\|_{1,1} \equiv \|x\|_1$ *and the problem* (27) *can be rewritten as follows.*

$$\min_{(x,y)\in K} \left\{ g(x,y) - \langle \bar{x}^l, x\rangle - \langle \bar{y}^l, y\rangle + \lambda\alpha\|x\|_1 - \langle v^l, x\rangle \right\}. \tag{28}$$

Thanks to the ℓ_1*-norm, if* $g(x, y)$ *is separable in its variables, so is the problem* (28)*. Hence, this problem is easier to solve. In many applications such as multi-task feature learning, group sparse PCA, group sparse optimal scoring, joint sparse compressed sensing, etc, it requires to estimate a row-wise sparse matrix* W*, meanwhile, its objective function is separable in columns. Therefore, the problem* (28) *can be separated into smaller sub-problems in these applications. Note, however, that the problem* (27) *is not separable if* $p \neq 1$*. Thus, we can say that the* $\ell_{1,0}$ *is the most interesting regularization for DCA.*

3.3. DCA for solving the second approximate problem

In the following, we introduce a DCA scheme for solving the problem (16) and indicate its connection with reweighted-$\ell_{p,1}$ procedure which includes reweighted-ℓ_1 and reweighted-$\ell_{2,1}$ as special cases. The problem (16) is a DC program of the form:

$$\min_{(x,y,z)} \{G_2(x,y,z) - H_2(x,y,z)\}, \tag{29}$$

where

$$G_2(x,y,z) = g(x,y) + \chi_{K_p}(x,y,z),$$
$$H_2(x,y,z) = h(x,y) + \lambda \sum_{j=1}^{J} (-\eta_\alpha)(z_j).$$

Let $(x^l, y^l, z^l) \in K_p$ be the current solution at iteration l. DCA applied to the DC program (29) updates $(x^{l+1}, y^{l+1}, z^{l+1}) \in K_p$ via two steps:

- Step 1: compute $(\bar{x}^l, \bar{y}^l) \in \partial h(x^l, y^l)$ and $v_j^l \in \lambda \partial(-\eta_\alpha)(z_j^l)\ \forall j = 1, ..., J$ by

$$v_j^l = \begin{cases} -\lambda\alpha & \text{if} \quad z_j^l \leq 1/\alpha \\ 0 & \text{otherwise.} \end{cases}$$

- Step 2: compute

$$(x^{l+1}, y^{l+1}, z^{l+1}) \in \arg\min_{(x,y,z)\in K_p} \left\{ g(x,y) - \langle \bar{x}^l, x\rangle - \langle \bar{y}^l, y\rangle - \langle v^l, z\rangle \right\}. \tag{30}$$

Since $v_j^l \leq 0\ \forall j = 1, ..., J$, the (30) is equivalent to

$$\begin{cases} (x^{l+1}, y^{l+1}) & = \arg\min_{(x,y)\in K} \left\{ g(x,y) - \langle \bar{x}^l, x\rangle - \langle \bar{y}^l, y\rangle + \sum_{j=1}^J (-v_j^l)\|x^j\|_p \right\}, \\ z_j^{l+1} & = \|(x^j)^{l+1}\|_p\ \forall j = 1, ..., J. \end{cases}$$

Hence DCA for solving the problem (16) can be described as follows.

DCA2

Initialization: Choose an initial point (x^0, y^0), a tolerance τ and $l \leftarrow 0$.

repeat

1. Compute $(\bar{x}^l, \bar{y}^l) \in \partial h(x^l, y^l)$ and v^l by

$$v_j^l = \begin{cases} -\lambda\alpha & \text{if} \quad \|(x^j)^l\|_p \leq 1/\alpha \\ 0 & \text{otherwise.} \end{cases}$$

2. Compute (x^{l+1}, y^{l+1}) by solving the problem:

$$\min_{(x,y)\in K} \left\{ g(x,y) - \langle \bar{x}^l, x\rangle - \langle \bar{y}^l, y\rangle + \sum_{j=1}^J (-v_j^l)\|x^j\|_p \right\}. \tag{31}$$

3. $l \leftarrow l + 1$.

until Stopping criterion

Remark 2. *If the function f is convex, we can choose DC components of f as $g = f$ and $h = 0$. Then $(\bar{x}^l, \bar{y}^l) = 0\ \forall l$. In this case, the problem in step 2 becomes*

$$\min_{(x,y)\in K} \left\{ f(x,y) + \sum_{j=1}^{J} (-v_j^l) \|x^j\|_p \right\}. \tag{32}$$

We see that (32) has the form of an $\ell_{p,1}$-regularization problem but with different weights on groups x^j. Hence DCA2 iteratively solves the weighted-$\ell_{p,1}$ problem (32) with weights $(-v_j^l)$ being updated at each iteration l.

For $p = 1$, (32) becomes a group-weighted-ℓ_1 problem. Moreover, if $f(x, y)$ is separable, problem (32) can be separated into independent sub-problems. This is the case in many applications. In addition, the $\ell_{1,0}$-regularization can also promote sparsity within each group.

For $p = 2$, (32) becomes a weighted-$\ell_{2,1}$ problem and the DCA in this case is reweighted-$\ell_{2,1}$ algorithm. Note that the adaptive group Lasso proposed in Wei & Huang (2010) is also of this type. However, the adaptive group Lasso only processes in one step and heuristically computes weights by $(-v_j) = 1/\|\bar{x}^j\|_2$ where $\bar{x}$ is an initial estimate of the solution.

4. Application to group variable selection in optimal scoring problem

In this section, we consider the problem of group variable selection in linear discriminant analysis (LDA). Instead of directly considering the Fisher formulation (6), we are interested in the optimal scoring interpretation of LDA (Hastie et al., 1994, 1995). The rationality of the optimal scoring method derives from the fact that LDA can also be reformulated as a multiple regression problem via optimal scoring. Generally, the problem can be formulated as follows.

Let $\{(x_i, y_i) : i = 1, ..., n\}$ be a set of labeled training data with observation vector $x_i \in \mathbb{R}^d$ and label $y_i \in \{1, ..., C\}$. The data matrix is denoted by $X = [x_1; ...; x_n] \in \mathbb{R}^{n\times d}$. We assume that the features have been standardized to have mean 0 and variance 1. Let $Y \in \mathbb{R}^{n\times C}$ with $Y_{ik} = 1$ if x_i belongs to the k-th class and 0 otherwise. To find the linear transformation $W \in \mathbb{R}^{d\times L}$, that maps

the data in the d-dimensional space to the L-dimensional space ($L \leq C-1$), in which the between-class variance is maximized while the within-class variance is minimized, the optimal scoring criterion solves the problem

$$\begin{aligned} \min_{\Theta \in \mathbb{R}^{C\times L}, W\in\mathbb{R}^{d\times L}} \quad & \left\{ \frac{1}{2n} ||Y\Theta - XW||_F^2 \right\} \\ \text{subject to} \quad & \frac{1}{n}\Theta^T Y^T Y \Theta = I_L, \end{aligned} \tag{33}$$

where I_L is the $L \times L$ identity matrix. Hastie et al. (1994) proposed an algorithm for solving the problem (33) described as below.

(1) Choose a score matrix Θ_0 such that $\frac{1}{n}\Theta_0^T Y^T Y \Theta_0 = I_L$.

(2) Compute $\hat{W}$ by solving the following problem

$$\min_{W\in\mathbb{R}^{d\times L}} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 \right\}. \tag{34}$$

(3) Compute eigenvector matrix V and corresponding eigenvalues $\lambda_1, ..., \lambda_L$ of $\Theta_0^T Y^T X \hat{W}$.

(4) Compute solution $\Theta^* = \Theta_0 V$ and $W^* = \hat{W} V$.

The classification rule is to assign a new observation x to class y if

$$y = \arg\max_k \|(x - u_k)^T W^* D\|_2^2,$$

where D is a diagonal matrix with k-th diagonal term $D_{kk} = 1/\sqrt{\lambda_k^2(1-\lambda_k^2)}$ and u_k is the mean vector of class k.

In high-dimensional settings, there are many irrelevant and/or redundant features. In addition, the classification rule involves a linear combination of the features. Hence one difficulty of the LDA is data interpretation. The most suitable approach to overcome this difficulty is feature selection. The resulting sparse discriminant vectors can provide a more interpretable low-dimensional representation of data. Recently, Clemmensen et al. (2011) has used an individual variable selection approach by imposing the $\ell_2 + \ell_1$ penalty on each discriminant vector. In Le Thi & Phan (2016b), the authors have developed alternating schemes based on DCA for solving the sparse optimal scoring problem using the $\ell_2 + \ell_0$-regularization. These methods successively find sparse

discriminant vectors. Leng (2008); Merchante et al. (2012) have used the $\ell_{2,1}$-norm which is a convex approximation of the $\ell_{2,0}$-norm in order to select the same features in all discriminant vectors.

In the optimal scoring problem, the feature j is selected if at least a component in the row j of W is nonzero and vice versa. Therefore, it is reasonable to consider rows of W as groups. Denote by $W_{j:}$ $(j = 1, \ldots, d)$ and $W_{:k}$ $(k = 1, \ldots, L)$ the j-th row and the k-th column of W, respectively. In the step 2, using $\ell_{p,0}$-regularization for the problem (34) leads us to consider the group variable selection problem in optimal scoring.

$$\min_{W \in \mathbb{R}^{d \times L}} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \lambda \|W\|_{p,0} \right\}, \tag{35}$$

where $\lambda \geq 0$ is a tuning parameter, and the $\ell_{p,0}$-norm of W is defined by $\|W\|_{p,0} = \sum_{j=1}^{d} s(\|W_{j:}\|_p)$, i.e., we consider W as a vector by concatenating its rows which are regarded as groups.

In the appendix Appendix A, we will prove that the optimal solution set of the problem (35) is bounded. Hence, without loss of generality we can only consider the problem (35) on a box $K = [-M, M]^{d \times L}$ for sufficient large M. The resulting problem is

$$\min_{W \in K} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \lambda \|W\|_{p,0} \right\}. \tag{36}$$

Observe that the problem (36) is a special case of (1) where the function f is given by

$$f(W) = \frac{1}{2n} ||Y\Theta_0 - XW||_F^2,$$

and the corresponding approximate problem takes the form:

$$\min_{W \in K} \left\{ F^p(W) := \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \lambda \sum_{j=1}^{d} \eta_\alpha(\|W_{j:}\|_p) \right\}. \tag{37}$$

By Proposition 1, this problem is equivalent to

$$\min_{(W,z) \in K_p} \left\{ F(W, z) := \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \lambda \sum_{j=1}^{d} \eta_\alpha(z_j) \right\}, \tag{38}$$

where $K_p = \{(W, z) : W \in K, \|W_{j:}\|_p \leq z_j, j = 1, \ldots, d\}$.

By Proposition 3, there exists α_0 such that the approximate problem (37) is equivalent to the original problem (35) for all $\alpha > \alpha_0$.

Here, f is a convex quadratic function, and DC components of f are taken as $g = f$ and $h = 0$. Hence a DC formulation of the problem (37) is

$$\min_{W} \{F^p(W) := G_1(W) - H_1(W)\}, \tag{39}$$

where

$$G_1(W) = \frac{1}{2n}||Y\Theta_0 - XW||_F^2 + \lambda\alpha\|W\|_{p,1} + \chi_K(W),$$
$$H_1(W) = \lambda \sum_{j=1}^{d} r(\|W_{j:}\|_p).$$

The problem (38) is also a DC program defined by

$$\min_{(W,z)} \{F(W, z) := G_2(W, z) - H_2(W, z)\}, \tag{40}$$

where

$$G_2(W, z) = \frac{1}{2n}||Y\Theta_0 - XW||_F^2 + \chi_{K_p}(W, z),$$
$$H_2(W, z) = \lambda \sum_{j=1}^{d} (-\eta_\alpha)(z_j).$$

According to DCA1 with $p = 1$, at each iteration l, $\ell_{1,0}$(DCA1) computes $V^l \in \partial\lambda \sum_{j=1}^{d} r(\|W_{j:}^l\|_1)$, and then computes W^{l+1} by solving the problem:

$$\min_{W \in K} \left\{ \frac{1}{2n}||Y\Theta_0 - XW||_F^2 + \lambda\alpha\|W\|_{1,1} - \sum_{j=1}^{d} \langle V_{j:}^l, W_{j:} \rangle \right\}. \tag{41}$$

The subgradient V^l is computed by

$$V_{jk}^l = \begin{cases} \text{sgn}(W_{jk}^l)\lambda\alpha & \text{if} \quad \|W_{j:}^l\|_1 > 1/\alpha \\ 0 & \text{otherwise,} \end{cases}, k = 1, ..., L.$$

The problem (41) can be separated into L independent sub-problems of the same form:

$$\min_{W_{:k} \in [-M,M]^d} \left\{ f_k(W_{:k}) := \frac{1}{2n}||(Y\Theta_0)_{:k} - XW_{:k}||_2^2 + \lambda\alpha\|W_{:k}\|_1 - \langle V_{:k}^l, W_{:k} \rangle \right\}, \tag{42}$$

where $(Y\Theta_0)_{:k}$ denotes the k-th column of $Y\Theta_0$. Hence for solving the problem (41), we can solve L sub-problems (42) in parallel. This is an interesting property of the $\ell_{1,0}$-regularization. For solving problem (42), we use the coordinate descent method (Friedman et al., 2007). We choose a component W_{jk} to minimize, and consider the other components $\hat{W}_{mk}, m \neq j$ as fixed. The resulting optimization problem is

$$\min_{W_{jk}\in[-M,M]} \left\{ \frac{1}{2n}\sum_{i=1}^{n}[(Y\Theta_0)_{ik} - \sum_{m\neq j} X_{im}\hat{W}_{mk} - X_{ij}W_{jk}]^2 + \lambda\alpha|W_{jk}| - V_{jk}^l W_{jk} \right\}.$$

This problem can be rewritten as follows.

$$\min_{W_{jk}\in[-M,M]} \left\{ \frac{1}{2}\|W_{jk}\|_2^2 - R_j W_{jk} + n\lambda\alpha|W_{jk}| - nV_{jk}^l W_{jk} \right\}, \tag{43}$$

where $R_j = \sum_{i=1}^{n} X_{ij}[(Y\Theta_0)_{ik} - \sum_{m\neq j} X_{im}\hat{W}_{mk}]$. The optimal solution to the convex sub-problem (43) can be explicitly computed. Indeed, we have the following lemma.

Lemma 1. *The solution to the problem* (43) *takes the form*

$$\hat{W}_{jk} = \mathbf{proj}_{[-M,M]}\left(\mathcal{S}\left(R_j + nV_{jk}^l, n\lambda\alpha\right)\right), \tag{44}$$

where $\mathbf{proj}_X(.)$ *denotes projection on* X *and* $\mathcal{S}$ *is the soft-thresholding operator defined by*

$$\mathcal{S}(z,t) = \begin{cases} z - t & \text{if } z > 0 \text{ and } t < |z|, \\ z + t & \text{if } z < 0 \text{ and } t < |z|, \\ 0 & \text{if } t \geq |z|. \end{cases}$$

The proof of Lemma 1 is given in Appendix Appendix B. The update (44) is repeated for $j = 1, ..., d, 1, ...$ until

$$\|\hat{W}_{:k}^h - \hat{W}_{:k}^{h-1}\|_2 \leq \epsilon(\|\hat{W}_{:k}^{h-1}\|_2 + 1),$$

or

$$|f_k(\hat{W}_{:k}^h) - f_k(\hat{W}_{:k}^{h-1})| \leq \epsilon(|f_k(\hat{W}_{:k}^{h-1})| + 1),$$

where $\hat{W}_{:k}^h$ is the solution obtained at the h-th iteration and ϵ is a tolerance sufficient small.

Similarly, each iteration of $\ell_{1,0}$(DCA2) consists in computing $v_j^l = -\lambda\alpha$ if $\|W_{j:}^l\|_1 \leq 1/\alpha$ and 0 otherwise $\forall j = 1, ..., d$, and solving the problem:

$$\min_{W \in K} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \sum_{j=1}^{d} (-v_j^l) \|W_{j:}\|_1 \right\}. \tag{45}$$

The problem (45) can also be separated into L independent sub-problems of the same form,

$$\min_{W_{:k} \in [-M,M]^d} \left\{ \frac{1}{2n} ||(Y\Theta_0)_{:k} - XW_{:k}||_2^2 + \sum_{j=1}^{d} (-v_j^l) |W_{jk}| \right\}, \tag{46}$$

for which the coordinate descent method can be used.

The following result is a consequence of the convergence properties of DCA for DC polyhedral programs.

Theorem 1. *The algorithm $\ell_{1,0}$(DCA1) (resp. $\ell_{1,0}$(DCA2)) terminates after a finite number of iterations, and the solution W^* (resp. $(W^*, |W_d^*|_1)$) given by $\ell_{1,0}$(DCA1) (resp. $\ell_{1,0}$(DCA2)) is a critical point of the problem* (37) *(resp.* (38)*). Furthermore, if $\|W_{j:}^*\|_1 \neq \frac{1}{\alpha}$ $\forall j = 1, ..., d$, then W^* and $(W^*, |W_d^*|_1)$ are local solutions to* (37) *and* (38)*, respectively.*

Proof. Since the ℓ_1-norm and the function $\sum_{j=1}^{d} -\eta_\alpha(z_j)$ are polyhedral convex, the second DC component H_1 (resp. H_2) in (26) (resp. (29)) is polyhedral convex. Therefore, both (26) and (29) are polyhedral DC programs. According to the convergence property of polyhedral DC programs, $\ell_{1,0}$(DCA1) (resp. $\ell_{1,0}$(DCA2)) generates a sequence $\{W^l\}$ (resp. $\{(W^l, |W_d^l|_1)\}$) that converges to a critical point W^* (resp. $(W^*, |W_d^*|_1)$) after a finite number of iterations.

If $\|W_{j:}^*\|_1 \neq \frac{1}{\alpha}$ $\forall j = 1, ..., d$, then the second DC component H_1 (resp. H_2) is differentiable at W^* (resp. $(W^*, |W_d^*|_1)$). Hence, $\partial H_1(W^*) \subset \partial G_1(W^*)$ (resp. $\partial H_2(W^*, |W_d^*|_1) \subset \partial G_2(W^*, |W_d^*|_1)$). This is necessary local optimality condition for DC programs and also sufficient for polyhedral ones. It follows that W^* and $(W^*, |W_d^*|_1)$ are local solutions to (37) and (38), respectively. □

According to DCA1 with $p = 2$, $\ell_{2,0}$(DCA1) iteratively computes V^l by

$$V_{j:}^l = \begin{cases} \frac{\lambda\alpha}{\|W_{j:}^l\|_2} W_{j:}^l & \text{if} \quad \|W_{j:}^l\|_2 > 1/\alpha \\ 0 & \text{otherwise,} \end{cases}$$

and then solves the problem

$$\min_{W \in K} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \lambda\alpha \sum_{j=1}^{d} \|W_{j:}\|_2 - \sum_{j=1}^{d} \langle V_{j:}^l, W_{j:} \rangle \right\}. \tag{47}$$

Since $f(W)$ is not separable with respect to rows, the convex problem (47) cannot be separated into independent sub-problems as the previous cases. Here we apply a block coordinate descent algorithm for solving (47). We choose a row $W_{j:}$ to minimize, and consider the other rows $\hat{W}_{m:}, m \neq j$ as fixed. The resulting problem is

$$\min_{W_{j:} \in [-M,M]^L} \left\{ \frac{1}{2n} \sum_{i=1}^{n} ||r_{(-j,i)} - X_{ij} W_{j:}||_2^2 + \lambda\alpha \|W_{j:}\|_2 - \langle V_{j:}^l, W_{j:} \rangle \right\}, \tag{48}$$

where $r_{(-j,i)} = (Y\Theta_0)_{i:} - \sum_{m \neq j} X_{im} \hat{W}_{m:}$. The problem (48) can be rewritten as follows.

$$\min_{W_{j:} \in [-M,M]^L} \left\{ \frac{1}{2} \|W_{j:}\|_2^2 - \langle R_j, W_{j:} \rangle + n\lambda\alpha \|W_{j:}\|_2 \right\}, \tag{49}$$

where $R_j = \sum_{i=1}^{n} X_{ij} r_{(-j,i)} + nV_{j:}^l$. We notice that, without the constraint $W_{j:} \in [-M, M]^L$, the problem (49) becomes to the following unconstrained problem.

$$\min_{W_{j:}} \left\{ \frac{1}{2} \|W_{j:}\|_2^2 - \langle R_j, W_{j:} \rangle + n\lambda\alpha \|W_{j:}\|_2 \right\}. \tag{50}$$

The optimal solution to this problem is given by

$$\hat{W}_{j:} = (\|R_j\|_2 - n\lambda\alpha)_+ \frac{R_j}{\|R_j\|_2}, \tag{51}$$

where the positive part $(x)_+ = \max(x, 0)$. Clearly, if $\hat{W}_{j:} \in [-M, M]^L$, then the unconstrained solution $\hat{W}_{j:}$ is the optimal solution to the constrained problem (49). In practice, we realize that this is often the case, meaning the problem (49) can be efficiently solved. However, when $\hat{W}_{j:} \notin [-M, M]^L$, we use the

alternating direction method of multipliers (Boyd et al., 2011) for solving the problem (49) provided in Appendix Appendix C.

According to $\ell_{2,0}$(DCA2), at each iteration l, we have to compute $v_j^l = -\lambda\alpha$ if $\|W_{j:}^l\|_2 \leq 1/\alpha$ and 0 otherwise $\forall j = 1, ..., d$, and then compute W^{l+1} by solving the problem:

$$\min_{W \in K} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \sum_{j=1}^{d} (-v_j^l) \|W_{j:}\|_2 \right\}. \tag{52}$$

The convex sub-problem (52) is also solved by a block coordinate descent algorithm.

Theorem 2. *a) The sequence generated by $\ell_{2,0}$(DCA1) has at least one limit point and every limit point of this sequence is a critical point of the problem* (37).

b) The algorithm $\ell_{2,0}$(DCA2) terminates after a finite number of iterations, and the solution $(W^*, |W_d^*|_2)$ given by $\ell_{2,0}$(DCA2) is a critical point of the problem (38). *Furthermore, if $\|W_{j:}^*\|_2 \neq \frac{1}{\alpha}\ \forall j = 1, ..., d$, then $(W^*, |W_d^*|_2)$ is a local solution to* (38).

Proof. a) (a) is a consequence of convergence properties of general DC programs and the facts that the objective function of (37) is bounded below by 0 and the sequence generated by $\ell_{2,0}$(DCA1) is bounded.

b) For $p = 2$, (29) is also a polyhedral DC program. Hence $\ell_{2,0}$(DCA2) generates a sequence $\{(W^l, |W_d^l|_2)\}$ that converges to a critical point $(W^*, |W_d^*|_2)$ after a finite number of iterations. Furthermore, if $\|W_{j:}^*\|_2 \neq \frac{1}{\alpha}\ \forall j = 1, ..., d$, then the second DC component H_2 is differentiable at $(W^*, |W_d^*|_2)$. Hence, $\partial H_2(W^*, |W^*|_2) \subset \partial G_2(W^*, |W_d^*|_2)$. This is necessary local optimality condition for DC programs and also sufficient for polyhedral ones. Therefore, $(W^*, |W_d^*|_2)$ is a local solution to (38). □

5. Numerical experiments

5.1. Comparative algorithms

We will compare the proposed algorithms ($\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2), $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2)) with the two methods using the $\ell_{2,1}$-regularization in optimal scoring (OSGL (Leng, 2008; Merchante et al., 2012) and multiclass support vector machines (MSVMGL (Blodel et al., 2013)), and the method using the $\ell_2 + \ell_0$-regularization (ADCA (Le Thi & Phan, 2016b)). We make a comparison between the four proposed algorithms to study the performance of the two regularizations as well as the two different DC decompositions.

5.1.1. Sparse optimal scoring using group lasso (OSGL):

OSGL uses the $\ell_{2,1}$-norm instead of the $\ell_{p,0}$-norm in the problem (35), that is

$$\min_{W \in \mathbb{R}^{d \times L}} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \lambda \|W\|_{2,1} \right\}. \tag{53}$$

In the experiments, this problem is solved by a block coordinate descent algorithm.

5.1.2. Sparse multiclass support vector machine using group lasso (MSVMGL):

Let W be a $d \times C$ matrix, where d and C represent the number of features and the number of classes, respectively. Denote by $W_{:k} \in \mathbb{R}^d$ the k-th column of W. In multiclass support vector machine, an observation x is classified to one of the C classes using the following rule:

$$y = \arg \max_{k \in \{1,\dots,C\}} W_{:k}^T x.$$

Given n training observations (x_i, y_i), $i = 1, ..., n$, the goal of Blodel et al. (2013) is to estimate a row-wise sparse solution W by solving the following problem.

$$\min \left\{ \sum_{i=1}^{n} \sum_{k \neq y_i} \max(1 - (W_{:y_i}^T x_i - W_{:k}^T x_i), 0)^2 + \lambda \|W\|_{2,1} \right\}. \tag{54}$$

In Blodel et al. (2013), the authors use the block coordinate descent method for solving this problem. The code is published at `https://github.com/mblondel/lightning`.

5.1.3. Alternating scheme based on DCA for solving sparse optimal scoring (ADCA):

Le Thi & Phan (2016b) uses an $\ell_2 + \ell_0$ regularization for the optimal scoring problem in order to select features in each discriminant vector. Using the Capped-ℓ_1 approximation of the ℓ_0-norm, the authors seek sparse discriminant vectors $W_{:1}, ..., W_{:L}$ that successively solve the following problem

$$\begin{aligned} \min_{W_{:k}, \Theta_{:k}} \quad & \frac{1}{2n}||Y\Theta_{:k} - XW_{:k}||_2^2 + \lambda \left[\frac{1-\gamma}{2}||W_{:k}||_2^2 + \gamma \sum_{i=1}^{d} \eta_\alpha(w_{ki}) \right] \\ \text{subject to} \quad & \frac{1}{n}\Theta_{:k}^T Y^T Y \Theta_{:k} = 1; \quad \Theta_{:k}^T Y^T Y \Theta_{:l} = 0, l = 1, ..., k-1. \end{aligned} \tag{55}$$

Here $\gamma \in [0, 1]$ and $\lambda \geq 0$ are tuning parameters. Le Thi & Phan (2016b) developed alternating schemes based on DCA for solving this problem.

5.2. Datasets

We evaluate the performance of comparative algorithms on two synthetic datasets and a collection of real world datasets. The datasets are generated in the following ways.

For the first setup S1, we generate a three classes classification problem. Each class is assumed to have a multivariate normal distribution $N(\mu_k, \Sigma)$, $k = 1, 2, 3$ with dimension $q = 500$. All elements on the main diagonal of covariance matrix Σ are equal to 1 and all other elements are equal to 0.6. The first 35 components of μ_1 are 0.7, $\mu_{2j} = 0.7$ if $36 \leq j \leq 70$ and $\mu_{3j} = 0.7$ if $71 \leq j \leq 105$ and 0 otherwise. For each class, we generate 100 training samples, 100 tuning samples and 500 test samples.

For the last setup S2, we generate a three-class classification problem as follows: $i \in C_k$ then $X_{ij} \sim N((k-1)/2, 1)$ if $j \leq 100$, $k = 1, 2, 3$ and $X_{ij} \sim N(0, 1)$ otherwise, where $N(\mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2. A total of 300 training samples, 300 tuning samples and 1500 test samples are generated with equal probabilities for each class.

The real world datasets consist of seven real microarray gene expression datasets, and four face image datasets. The number of classes ranges from 3 to

10. All the datasets are preprocessed by normalizing each dimension of the data to zero mean and unit variance. The detailed information of these datasets is summarized in Table 1.

SRBCT (Khan et al., 2001) can be downloaded at `http://research.nhgri.nih.gov/microarray/Supplement/`.

The high-dimensional dataset consists of multi-spectral imaging of three penicillium species: melanoconodium, polonicum and venetum. This data is studied in Clemmensen et al. (2007).

The ALL/AML dataset was studied in the seminal paper of Golub et al. (1999). It is available at `http://www-genome.wi.mit.edu`. The authors studied the data for binary classification between AML and ALL. However, it can be treated as a three-class dataset (B-cell, T-cell and AML).

Leukemia microarray dataset is used in Yeoh et al. (2002) and available at `http://datam.i2r.a-star.edu.sg/datasets/krbd/`.

MLL-leukemia dataset can be downloaded at `http://research.dfci.harvard.edu/korsmeyer/Supp_pub/Supp_Armstrong_Main.html`.

The six remaining datasets are downloaded at the ASU feature selection repository `http://featureselection.asu.edu/`.

5.3. Experimental setups

All algorithms are implemented in the R 3.0.2, and performed on a PC Intel i7 CPU3770, 3.40 GHz of 8GB RAM.

In our experiments, we use the cross-validation scheme to validate the performance of various classifiers. Each real dataset is split into a training set and a test set respectively containing 2/3 and 1/3 of the observations. The test set is used to measure the accuracy of various classifiers given by the training procedure. This process is repeated 10 times, each with a random choice of training set and test set. The reported computational results are the average results over 10 runs with different training sets.

The stop tolerance of DCA is $\tau = 10^{-5}$ while the stop tolerance of the coordinate descent method is $\epsilon = 10^{-4}$. The starting point of DCA is zero. The

Table 1: Real datasets used in experiments.

Dataset	#Features	#Samples	#Classes
SRBCT (SRB)	2308	83	4
Pencillium (PEN)	3754	36	3
ALL/AML (ALL)	7129	72	3
Leukemia (LEU)	12558	248	6
MLL-Leukemia (MLL)	12582	72	3
CLL-SUB-111 (CLL)	11340	111	3
TOX-171 (TOX)	5748	171	4
AR10P (AR1)	2400	130	10
PIX10P (PIX)	10000	100	10
PIE10P (PIE)	2420	210	10
ORL10P (ORL)	10304	100	10

bound M is set to 10^3.

The tuning parameters λ and L (number of used discriminant vectors) were chosen by using the tuning sets for synthetic data and via 5-fold cross-validation procedure on the training sets for real data, from the sets of candidate values in the range $[\lambda_{\min}, \lambda_{\max}]$ for λ and $[1, C-1]$ for L. Observing that larger λ is, sparser the solution would be, we randomly select a synthetic data set (S1 in our experiment) and choose $\lambda_{\min}$ and $\lambda_{\max}$ as follows. We start with a quite small value λ so that all variable groups are selected, and increase this value until the obtaining of the "sparsest" solution (i.e., when the number of non-zero groups is no more change). Then $\lambda_{\min}$ (resp. $\lambda_{\max}$) is taken as the largest value so that all groups are selected (resp. the smallest value corresponding to the "sparsest" solution). By this way, the sets of values of λ and L in our experiment are given by

$$\Lambda = \{0.002, 0.004, 0.006, 0.008, 0.01, 0.014, 0.016, 0.018, 0.02, 0.024, 0.028, 0.032\},$$

and $\boldsymbol{L} = \{1, ..., C-1\}$, respectively.

Table 2: Numerical results on S1 with different values of α

α	Accuracy (%)	FS (#)	FS (%)	Training time (s)
1	99.86	84.7	16.94	0.17
5	100	75.4	15.08	0.08
10	99.4	74.1	14.2	0.079
25	99.86	70.81	14.16	0.07
50	99.6	63.57	12.71	0.07
125	99.73	59.62	11.8	0.09

For studying the influence of the approximate parameter α in the Capped-ℓ_1 function, we randomly choose an algorithm ($\ell_{1,0}$(DCA1)) and a data set (S1). For each value of α in $\{1, 5, 10, 25, 50, 125\}$, we use the validation sets to select λ and L from Λ and $\boldsymbol{L}$. We report the results in the Table 2.

We observe from the Table that the obtained results are similar: the accuracy varies from 99.4% to 100% while the sparsity is in the range $[11.8\%, 16.94\%]$. Hence, we set $\alpha = 5$ which has been showed to be efficient via the numerical results in several works Bradley & Mangasarian (1998); Ong & Le Thi (2013a); Le Thi et al. (2015, 2014b); Le Thi & Phan (2016b,a) (and gives the accuracy 100% in our experiment). By this way, we avoid performing tuning parameter selection on a three-dimensional grid.

5.4. Numerical results on synthetic data

In this experiment, we generate training, tuning, and test sets in the same manner as described in Sect. 5.2. The tuning sets are used to choose the parameters λ and the number of discriminant vectors used L, while the test sets are used to measure the accuracy of various classifiers trained on the training sets. We perform 10 trials for each experimental setting.

The experimental results on synthetic data are given in Table 3. In this table, the average number and percentage of selected features (FS (#) and FS (%)), the average percentage of accuracy of classifiers, as well as the average

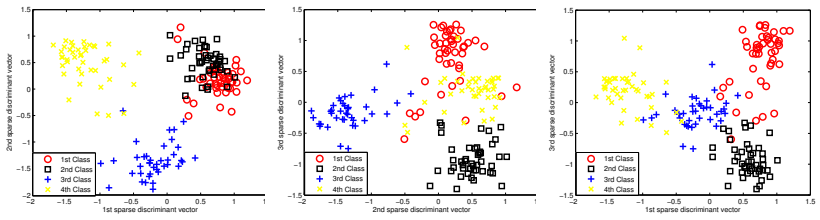

Figure 1: The TOX dataset was projected onto the first three sparse discriminant vectors. The samples in each class are shown by using a distinct symbol.

training time (standard deviations) in second over 10 trials are reported.

We observe from Table 3, in terms of both the accuracy and the feature selection, the four proposed algorithms give better results than OSGL, MSVMGL and ADCA. More precisely, $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) are comparable and better than the other algorithms. Comparing among the four regularizations, the best and the second best performing methods with respect to the accuracy of classifier and the sparsity are the $\ell_{1,0}$-regularization and the $\ell_{2,0}$-regularization, respectively.

The training time of all the algorithms on the synthetic data is quite small. We also see that $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) are faster than the other algorithms.

5.5. Numerical results on real datasets

The computational results (the average results on 10 runs with different training sets) given by $\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2), $\ell_{2,0}$(DCA1), $\ell_{2,0}$(DCA2), OSGL, MSVMGL and ADCA are reported in Tables 4-5. We are interested in the efficiency (the sparsity and the accuracy of classifiers), the number of used discriminant vectors L, as well as the rapidity of these algorithms. The discriminant vectors can be used to visualize the datasets such as in Figure 1. Observing from Tables 4-5, we have the following comments.

Sparsity: The classifiers obtained by $\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2), $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) are sparser than those obtained by OSGL, MSVMGL and ADCA on 9/11 datasets. ADCA and OSGL are slightly better than $\ell_{1,0}$(DCA1),

Table 3: Comparative results of $\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2), $\ell_{2,0}$(DCA1), $\ell_{2,0}$(DCA2), OSGL, MSVMGL and ADCA on the synthetic data in terms of the average number and percentage of selected features (FS (#) and FS (%)), the average percentage of accuracy of classifiers, and the average training time (standard deviations) in second over 10 trials. Bold fonts indicate the best results in each column.

		Accuracy (%)	FS (#)	FS (%)	Training time (s)
S1	$\ell_{1,0}$(DCA1)	**100** (0)	**75.4** (3.2)	**15.08** (0.64)	0.08 (0.021)
	$\ell_{1,0}$(DCA2)	**100** (0)	76.3 (2.83)	15.26 (0.56)	**0.05** (0.007)
	$\ell_{2,0}$(DCA1)	**100** (0)	99.2 (12.27)	19.85 (2.45)	0.19 (0.053)
	$\ell_{2,0}$(DCA2)	**100** (0)	102.9 (1.79)	20.8 (0.35)	0.27 (0.049)
	OSGL	99.98 (0.03)	231 (9.23)	46.2 (1.84)	0.53 (0.068)
	MSVMGL	98.26 (2.41)	238.6 (10.43)	47.72 (2.08)	2.16 (0.62)
	ADCA	**100** (0)	107.3 (1.55)	21.46 (0.31)	1.32 (0.17)
S2	$\ell_{1,0}$(DCA1)	**98.97** (0.29)	**97.9** (6.4)	**19.58** (1.28)	**0.02** (0.009)
	$\ell_{1,0}$(DCA2)	98.73 (0.3)	101.4 (8.93)	20.28 (1.78)	**0.02** (0.007)
	$\ell_{2,0}$(DCA1)	97.2 (0.15)	116.38 (7.82)	23.27 (1.56)	0.03 (0.004)
	$\ell_{2,0}$(DCA2)	96.73 (0.53)	100.5 (4.35)	20.1 (0.89)	0.06 (0.011)
	OSGL	93.3 (1.46)	125.3 (5.55)	25.06 (1.11)	0.06 (0.006)
	MSVMGL	91.95 (1.72)	128.7 (7.26)	25.74 (1.45)	1.05 (0.82)
	ADCA	97.09 (0.56)	116.6 (5.4)	23.32 (1.08)	0.37 (0.06)

Table 4: Comparative results of $\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2), $\ell_{2,0}$(DCA1), $\ell_{2,0}$(DCA2), OSGL, MSVMGL and ADCA in terms of the average number of selected features and its standard deviation (upper row), and the average percentage of selected features and its standard deviation (lower row) over 10 training/test set splits. Bold fonts indicate the best results in each row.

	$\ell_{1,0}$(DCA1)	$\ell_{1,0}$(DCA2)	$\ell_{2,0}$(DCA1)	$\ell_{2,0}$(DCA2)	OSGL	MSVMGL	ADCA
SRB	42.8 (13.14)	**35.1** (16.07)	90.24 (37.15)	87.1 (42.34)	100.3 (4.24)	123.3 (23.11)	70.6 (6)
	1.85 (0.56)	**1.52** (0.69)	3.9 (1.6)	3.77 (1.83)	4.34 (0.18)	5.34 (1)	3.05 (0.26)
PEN	7.7 (4.37)	**3.5** (0.7)	7.4 (4.22)	14.8 (3.76)	41.6 (3.97)	48.3 (5.07)	22.0 (4.41)
	0.2 (0.11)	**0.09** (0.01)	0.19 (0.11)	0.49 (0.1)	1.1 (0.1)	1.28 (0.13)	0.59 (0.12)
ALL	31.9 (12.69)	**31.8** (14.57)	52.5 (8.12)	70.8 (11.8)	83.8 (5.75)	366.5 (40.16)	63.1 (11.4)
	0.45 (0.17)	**0.44** (0.2)	0.73 (0.11)	0.99 (0.16)	1.17 (0.08)	5.14 (0.56)	0.88 (0.16)
LEU	88.6 (40.32)	97.1 (50.11)	50.1 (18.95)	109.4 (49.25)	243.9 (11.1)	2485.1 (1051.35)	**37.9** (5.65)
	0.7 (0.32)	0.77 (0.39)	0.39 (0.15)	0.87 (0.39)	1.94 (0.09)	19.78 (8.37)	**0.3** (0.05)
MLL	35.1 (10.02)	**32.4** (11.11)	56.9 (5.38)	51.4 (17.89)	126.6 (6.85)	162.43 (95.62)	132.6 (15.09)
	0.27 (0.07)	**0.25** (0.08)	0.45 (0.04)	0.41 (0.14)	1.01 (0.05)	1.29 (0.76)	1.05 (0.12)
CLL	40.1 (21.66)	**35.6** (30.6)	79.3 (23.07)	83.3 (64.47)	103.8 (5)	80.7 (7.78)	110.6 (25.84)
	0.35 (0.19)	**0.31** (0.26)	0.69 (0.2)	0.73 (0.56)	0.91 (0.04)	0.71 (0.06)	0.97 (0.22)
TOX	**129.9** (23.15)	132.2 (27.42)	164.6 (24.31)	226.7 (64.11)	269.8 (12.53)	262 (7.81)	275.3 (35.37)
	2.25 (0.4)	2.29 (0.47)	2.86 (0.42)	3.94 (1.11)	4.69 (0.21)	4.55 (0.13)	4.78 (0.61)
AR1	168.3 (49.77)	**88.8** (15.18)	342.1 (141.43)	231.6 (79.31)	351 (9.26)	302.8 (18.09)	363.7 (53.41)
	7.01 (2.07)	**3.7** (0.63)	14.29 (5.89)	9.65 (3.3)	14.63 (0.38)	12.61 (0.75)	15.15 (2.22)
PIX	116.1 (20.22)	**19.9** (9.12)	200.9 (18.57)	435.7 (92.31)	281.9 (16.17)	209.2 (20.96)	435.3 (73.3)
	1.16 (0.2)	**0.19** (0.09)	2.01 (0.18)	4.35 (0.92)	2.81 (0.16)	2.09 (0.2)	4.35 (0.73)
PIE	91 (19.31)	**30.5** (9.38)	201.9 (13.97)	237.1 (18.95)	281.3 (8.4)	65.3 (3.65)	206.3 (59.96)
	3.76 (0.79)	**1.2** (0.38)	8.34 (0.57)	9.79 (0.78)	11.62 (0.35)	2.69 (0.15)	8.52 (2.47)
ORL	327.5 (73.49)	254.4 (101.79)	315.9 (18.2)	276.8 (35.53)	**236.8** (8.35)	353.1 (27.27)	630.3 (87.53)
	3.17 (0.71)	2.46 (0.98)	3.06 (0.17)	2.68 (0.34)	**2.29** (0.08)	3.42 (0.26)	6.11 (0.84)

Table 5: Comparative results in terms of the average of percentage of accuracy of classifiers and its standard deviation (first row) over 10 training/test set splits, the number of used discriminant vectors L (second row), and the average training time (in seconds) and its standard deviation (third row). Bold fonts indicate the best results in each row.

	$\ell_{1,0}$(DCA1)	$\ell_{1,0}$(DCA2)	$\ell_{2,0}$(DCA1)	$\ell_{2,0}$(DCA2)	OSGL	MSVMGL	ADCA
SRB	**100** (0)	**100** (0)	99.1 (1.56)	99.64 (1.12)	99.61 (1.21)	97.47 (4.54)	99.64 (1.12)
	3	3	3	3	3	-	3
	0.94	**0.93** (0.01)	30.81 (4.72)	25.48 (6.59)	15.17 (0.18)	71.93 (9.17)	5.13 (1.01)
PEN	**100** (0)	**100** (0)	**100** (0)	**100** (0)	**100** (0)	**100** (0)	**100** (0)
	2	2	2	2	2	-	2
	1.76 (0.03)	2.63 (0.016)	13.27 (8.07)	12.66 (8.21)	18.52 (7.94)	111.08 (52.06)	7.08 (1.88)
ALL	95.89 (3.31)	96.24 (3.07)	95.37 (5.32)	95.76 (7.1)	95.34(2.42)	94.6 (3.43)	**97.06**(2.03)
	2	2	2	2	2	-	2
	8.66 (0.06)	12.83 (0.06)	108.42 (31.32)	95.24 (21.17)	83.58 (10.89)	76.31 (12.08)	48.8 (9.87)
LEU	**98.38** (1.29)	96.84 (0.85)	98.07 (1.18)	95.97 (1.55)	97.09 (1.64)	74.27 (5.47)	95.52 (2.49)
	5	5	5	5	5	-	5
	47.14 (0.88)	56.93 (1.3)	349.45 (107.35)	128.32 (108.93)	193.92 (62.45)	74.54 (191.96)	106.05 (30.65)
MLL	**98.76**(1.98)	97.12 (1.99)	95.86 (3.35)	94.87 (5.68)	97.08 (2.87)	88.67 (2.62)	97.92 (2.19)
	2	2	2	2	2	-	2
	28.44 (1.4)	**28.14** (0.47)	406.58 (89.77)	337.22 (99.36)	252.2 (17.28)	318.75 (65.91)	163.18 (51.28)
CLL	**86.47** (3.26)	85.99 (1.59)	84.28 (3.49)	77.31 (5.4)	78.88 (4.59)	64.72 (5.13)	83.99 (2.12)
	2	2	2	2	2	-	2
	21.98 (1.04)	32.72 (1.66)	304.79 (196.73)	310.78 (78.83)	279.57 (63.55)	164.45 (15.08)	301.13 (39.66)
TOX	91.92 (1.69)	92.45 (1.44)	94.73 (1.84)	91.92 (3.11)	92.45 (3.51)	92.98 (5.72)	**96.31** (2.4)
	2	2	2	2	2	-	2
	8.21 (0.7)	**8.1** (0.17)	88.91 (22.4)	36.12 (19.34)	83.76 (12.54)	339.39 (24.82)	109.74 (17.36)
AR1	97.24 (1.5)	96.57 (1.13)	**98.38** (1.54)	96.67 (2.49)	97.5 (2.65)	94.56 (3.17)	**98.38** (1.11)
	9	9	9	9	9	-	9
	3.23 (0.26)	3.94 (0.27)	203.22 (48.48)	152.53 (31.36)	140.64 (5.51)	376.05 (69.81)	57.91 (4.79)
PIX	98.51 (1.57)	98.51 (1.57)	95.1 (5.64)	**99.14** (1.92)	97.84 (3.3)	96.47 (3.34)	98.57 (1.51)
	9	9	9	9	9	-	9
	79.08 (1.98)	**78.11** (0.66)	3539.9 (283.71)	3998.36 (800.78)	2531.92 (256.66)	4271.62 (137.98)	1283.61 (135.65)
PIE	**100** (0)	**100** (0)	99 (1.35)	99.71 (0.9)	**100** (0)	98.57 (1.16)	**100** (0)
	9	9	9	9	9	-	9
	3.23 (0.09)	**3.11** (0.04)	103.94 (18.29)	86.96 (6.93)	63.84 (3.73)	23.36 (4.28)	54.14 (11.5)
ORL	**98.84** (1.49)	98.81 (1.54)	96.21 (4.76)	94.66 (3.77)	98.77 (1.58)	94.92 (3.51)	98.51 (1.48)
	9	9	9	9	9	-	9
	85.09 (2.66)	99.79 (5.05)	3581.92 (1111.72)	2788.62 (524.3)	1085.26 (122.8)	4118.03 (271.39)	744.63 (184.85)

$\ell_{1,0}$(DCA2), $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) on the two datasets LEU and ORL, respectively. $\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2), $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) respectively select from 0.2% to 7.01%, 0.09% to 3.7%, 0.19% to 14.29% and 0.41% to 9.79% of features while ADCA, OSGL and MSVMGL select from 0.3% to 15.15%, from 0.91% to 14.63% and from 0.71% to 19.78% of features, respectively. Comparing among the four regularizations, we see that the $\ell_{1,0}$-regularization and the $\ell_{2,0}$-regularization are respectively the best and second best in terms of the sparsity. We also observe that DCA1 and DCA2 are comparable. More precisely, $\ell_{1,0}$(DCA2) is better than $\ell_{1,0}$(DCA1) on 9/11 datatsets while $\ell_{2,0}$(DCA2) is better than $\ell_{2,0}$(DCA1) on 4/11 datasets.

Accuracy of classifiers: The proposed algorithms not only provide a good performance in terms of feature selection, but also give a high accuracy of classifiers. Their accuracy of classifiers are better than OSGL, MSVMGL and ADCA on most of the datasets. Comparing among the four regularizations, in terms of the accuracy, the $\ell_{1,0}$(DCA) is the best performing method while $\ell_{2,0}$(DCA) and ADCA are the second best performing methods. More precisely, $\ell_{1,0}$(DCA) provides the best results on 7/11 datasets while $\ell_{2,0}$(DCA) and ADCA respectively achieve the best results on 3/11 datasets and 5/11 datasets. Here, notably the results obtained with $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) are still superior to that of OSGL and MSVMGL on most of the datasets.

Training time: $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) are remarkable faster than the other algorithms (the ratios of gains are from 3 to 63 times). Especially, $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) run much faster than the other algorithms on the datasets with large number of features and large number of classes (face image datasets). This can be explained by the fact that $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) lead to the sequences of convex problems which are separated into independent sub-problems. Moreover, we use "foreach" and "doParallel" packages to solve multiple sub-problems in parallel by utilizing multiprocessor/multicore computer. We observe that OSGL runs faster than $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) on 8/11 datasets. This is because OSGL only solves a convex problem while $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) have to solve the sequences of convex sub-problems

of the same type as OSGL. The training time of ADCA is quite good: ADCA is faster than $\ell_{2,0}$(DCA1), $\ell_{2,0}$(DCA2), OSGL and MSVMGL on most of the datasets.

6. Conclusion

We have intensively studied DC programming and DCA for group variable selection problem including the $\ell_{p,0}$-norm in the objective function. DC approximation approaches have been investigated from both a theoretical and an algorithmic point of view. We have proved that the Capped-ℓ_1 approximate problem is equivalent to the original problem with a sufficiently large approximation parameter. Considering the two equivalent formulations of the approximate problem we have developed DCA schemes for solving them. When $p = 1$ and $p = 2$, the four DCA based algorithms can be viewed as an $\ell_{2,1}$-perturbed algorithm, reweighted-$\ell_{2,1}$ algorithm, ℓ_1-perturbed algorithm, reweighted-ℓ_1 algorithm with different weights on groups and the same weight on each group. Concerning the group variable selection in optimal scoring, among the four DCA schemes, three ($\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2) and $\ell_{2,0}$(DCA2)) have the interesting convergence properties: they almost always converge to local solutions after a finite number of iterations. We have also showed several useful properties of the $\ell_{1,0}$-regularization for group variable selection. The achieved solutions by nonconvex methods using the $\ell_{1,0}$-regularization are sparser than that of the others. At each iteration, $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) can solve independent convex sub-problems in parallel. Numerical experiments confirm the theoretical results: $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) have obtained the best performance in terms of accuracy of classifiers and feature selection, and have taken the shortest time for training. The second best performing methods are $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) which provide better results than the remaining three methods OSGL, MSVMGL and ADCA in terms of the sparsity and the accuracy.

For the future works, we plan to study group variable selection for other applications. We will also study other mixed-norm regularizations such as the

$\ell_{0,0}$-regularization for variable selection problem.

Appendix A. Bounded optimal solution set of the problem (35)

Let $\mathcal{P}$ be the optimal solution set of the problem (35). In the sequel, $X_{:1}, ..., X_{:d}$ denote the columns of the data matrix X. We will prove that $\mathcal{P}$ is bounded.

Lemma 2. *Assume that $\bar{W} \in \mathcal{P}$ and let $I = \{i : \bar{W}_{i:} \neq 0\}$. Then $\{X_{:i}\}_{i\in I}$ is linearly independent.*

Proof. We suppose that $\{X_{:i}\}_{i\in I}$ is not linearly independent. Therefore, there exists $i_0 \in I$ such that $X_{:i_0}$ can be represented by a linear combination of $\{X_{:i}\}_{i\in I\setminus\{i_0\}}$. That is, there exists $\delta = (\delta^i)_{i\in I\setminus\{i_0\}} \in \mathbb{R}^{|I|-1}$ such that

$$X_{:i_0} = \sum_{i\in I\setminus\{i_0\}} \delta^i X_{:i}. \tag{A.1}$$

Hence we have

$$\begin{aligned} Y\Theta_0 - X\bar{W} = Y\Theta_0 - \sum_{i=1}^{d} X_{:i}\bar{W}_{i:}^T &= Y\Theta_0 - \sum_{i\in I} X_{:i}\bar{W}_{i:}^T \\ &= Y\Theta_0 - \sum_{i\in I\setminus\{i_0\}} X_{:i}(\bar{W}_{i:} + \delta^i \bar{W}_{i_0:})^T. \end{aligned} \tag{A.2}$$

We define $\hat{W} \in \mathbb{R}^{d\times L}$ by

$$\hat{W}_{i:} = \begin{cases} \bar{W}_{i:} + \delta^i \bar{W}_{i_0:} & \text{if} \quad i \in I \setminus \{i_0\} \\ 0 & \text{otherwise.} \end{cases}$$

By construction of $\hat{W}$ and (A.2), we obtain

$$Y\Theta_0 - X\bar{W} = Y\Theta_0 - \sum_{i\in I\setminus\{i_0\}} X_{:i}(\bar{W}_{i:} + \delta^i \bar{W}_{i_0:})^T = Y\Theta_0 - X\hat{W}. \tag{A.3}$$

Moreover, we notice that $\sum_{i=1}^{d} s(\|\hat{W}_{i:}\|_p) \leq |I| - 1$ and $\sum_{i=1}^{d} s(\|\bar{W}_{i:}\|_p) = |I|$. Then

$$\begin{aligned} \frac{1}{2n}\|Y\Theta_0 - X\hat{W}\|_F^2 + \lambda\sum_{i=1}^{d} s(\|\hat{W}_{i:}\|_p) &\leq \frac{1}{2n}\|Y\Theta_0 - X\hat{W}\|_F^2 + \lambda|I| - \lambda \\ &< \frac{1}{2n}\|Y\Theta_0 - X\bar{W}\|_F^2 + \sum_{i=1}^{d} s(\|\bar{W}_{i:}\|_p). \end{aligned}$$

This contradicts the hypothesis that $\bar{W}$ is an optimal solution of the problem (35). The proof of Lemma 2 is then completed. □

Given $I \subset \{1, 2, ..., d\}$, X_I denotes the $n \times |I|$ matrix whose columns are $X_{:i}$, $i \in I$ and W_I denotes the $|I| \times L$ matrix whose rows is $W_{i:}$, $i \in I$. Let $I_W = \{i : W_{i:} \neq 0\}$ and λ_I denotes the smallest eigenvalue of $X_I^T X_I$. We denote

$$S = \{I \subset \{1, 2, ..., d\} : \{X_{:i}\}_{i \in I} \text{ is linearly independent}\}.$$

It follows that $\forall I \in S$, $X_I^T X_I \in \mathbb{R}^{|I| \times |I|}$ is positive definite, so $\lambda_I > 0$. Since S is a finite set, then we obtain

$$\lambda_0 = \min\{\lambda_I : I \in S\} > 0.$$

Proposition 5. *The optimal solution set of the problem* (35) *is bounded, i.e.,* $\forall\ W \in \mathcal{P}$ *then*

$$\|W\|_F \leq \frac{2\|Y\Theta_0\|_F}{\sqrt{\lambda_0}}.$$

Proof. By Lemma 2, if W is an optimal solution to the problem (35), then $I_W \in S$. We have

$$XW = \sum_{i \in I_W} X_{:i} W_{i:}^T = X_{I_W} W_{I_W}.$$

Hence

$$\|XW\|_F^2 = Tr(W_{I_W}^T X_{I_W}^T X_{I_W} W_{I_W}) \geq \lambda_{I_W} \|W_{I_W}\|_F^2 = \lambda_{I_W} \|W\|_F^2. \tag{A.4}$$

Since W is an optimal solution to the problem (35), we have

$$\frac{1}{2n}\|Y\Theta_0 - XW\|_F^2 + \lambda \sum_{i=1}^{d} s(\|W_{i:}\|_p) \leq \frac{1}{2n}\|Y\Theta_0\|_F^2.$$

Then, we get

$$\|Y\Theta_0 - XW\|_F \leq \|Y\Theta_0\|_F \Rightarrow \|XW\|_F \leq 2\|Y\Theta_0\|_F \tag{A.5}$$

Combining (A.4) and (A.5), it follows that

$$\sqrt{\lambda_{I_W}}\|W\|_F \leq 2\|Y\Theta_0\|_F. \tag{A.6}$$

This leads to

$$\sqrt{\lambda_0}\|W\|_F \leq 2\|Y\Theta_0\|_F \Leftrightarrow \|W\|_F \leq \frac{2\|Y\Theta_0\|_F}{\sqrt{\lambda_0}}.$$

The Proposition 5 has been proved. □

Appendix B. Proof of Lemma 1

Since the problem (43) is strongly convex, it has an unique solution. We need to show that $\hat{W}_{jk}$ given by (44) is the solution to the problem (43), i.e., there exist (μ^*, ν^*) such that $(\hat{W}_{jk}, \mu^*, \nu^*)$ satisfies the following KKT conditions of the problem (43).

$$\hat{W}_{jk} - R_j + n\lambda\alpha\partial|\hat{W}_{jk}| - nV_{jk}^l + \mu^* - \nu^* \ni 0, \tag{B.1}$$

$$\mu^*(\hat{W}_{jk} - M) = 0, \tag{B.2}$$

$$\nu^*(\hat{W}_{jk} + M) = 0, \tag{B.3}$$

$$\mu^*, \nu^* \geq 0, \tag{B.4}$$

$$\hat{W}_{jk} - M \leq 0, \tag{B.5}$$

$$\hat{W}_{jk} + M \geq 0. \tag{B.6}$$

Since $\hat{W}_{jk}$ is the projection of $\mathcal{S}\left(R_j + nV_{jk}^l, n\lambda\alpha\right)$ on the box $[-M, M]$, the inequalities (B.5) and (B.6) hold. We now consider case-by-case.

The first case $\hat{W}_{jk} > 0$, we choose $\nu^* = 0$ satisfied the conditions (B.3) and (B.4). It follows from $\hat{W}_{jk} > 0$ that $\mathcal{S}\left(R_j + nV_{jk}^l, n\lambda\alpha\right) = R_j + nV_{jk}^l - n\lambda\alpha > 0$. If $R_j + nV_{jk}^l - n\lambda\alpha < M$, then $\hat{W}_{jk} = R_j + nV_{jk}^l - n\lambda\alpha$. Hence we can choose $\mu^* = 0$ such that $(\hat{W}_{jk}, \mu^*, \nu^*)$ satisfies the conditions (B.1), (B.2) and (B.4). If $R_j + nV_{jk}^l - n\lambda\alpha \geq M$, then $\hat{W}_{jk} = M$. Hence $\mu^* = R_j + nV_{jk}^l - n\lambda\alpha - M \geq 0$ which satisfies the conditions (B.1), (B.2) and (B.4).

The second case $\hat{W}_{jk} < 0$, we choose $\mu^* = 0$ satisfied the conditions (B.2) and (B.4). It follows from $\hat{W}_{jk} < 0$ that $\mathcal{S}\left(R_j + nV_{jk}^l, n\lambda\alpha\right) = R_j + nV_{jk}^l + n\lambda\alpha < 0$. If $R_j + nV_{jk}^l + n\lambda\alpha > -M$, then $\hat{W}_{jk} = R_j + nV_{jk}^l + n\lambda\alpha$. Hence we can choose $\nu^* = 0$ such that $(\hat{W}_{jk}, \mu^*, \nu^*)$ satisfies the conditions (B.1), (B.3) and (B.4). If

$R_j+nV_{jk}^l+n\lambda\alpha \leq -M$, then $\hat{W}_{jk} = -M$. Hence $\nu^* = -R_j-nV_{jk}^l-n\lambda\alpha-M \geq 0$ which satisfies the conditions (B.1), (B.3) and (B.4).

The final case $\hat{W}_{jk} = 0$, we then have $|R_j+nV_{jk}^l| \leq n\lambda\alpha$. Let $(\mu^*, \nu^*) = (0, 0)$ which satisfies the conditions (B.2), (B.3) and (B.4). The condition (B.1) can be rewritten as follows.

$$-R_j + n\lambda\alpha[-1, 1] - nV_{jk}^l \ni 0. \tag{B.7}$$

It follows from $|R_j + nV_{jk}^l| \leq n\lambda\alpha$ that there exists $\kappa \in [-1, 1]$ such that $R_j + nV_{jk}^l = n\lambda\alpha\kappa$. This implies that the condition (B.7) holds. The proof is then complete.

Appendix C. Alternating direction method of multipliers for solving (49)

Initialization: Choose x^0, y^0, $\rho > 0$, $h \leftarrow 0$, and $\tau > 0$.

repeat

1. Compute $W_{j:}^{h+1}$ by

$$W_{j:}^{h+1} = \mathbf{proj}_{[-M,M]^L}\left(\frac{R_j - y^h + \rho x^h}{\rho + 1}\right).$$

2. Compute x^{h+1} by

$$x^{h+1} = \left(\|W_{j:}^{h+1} + y^h/\rho\|_2 - n\lambda\alpha/\rho\right)_+ \frac{W_{j:}^{h+1} + y^h/\rho}{\|W_{j:}^{h+1} + y^h/\rho\|_2}.$$

3. Compute $y^{h+1} = y^h + \rho(W_{j:}^{h+1} - x^{h+1})$.
4. $h \leftarrow h + 1$.

until $|\varphi(W_{j:}^h) - \varphi(W_{j:}^{h-1})| \leq \tau(|\varphi(W_{j:}^{h-1})| + 1)$.

Here φ denotes the objective function of the problem (49).

Acknowledgment

This research is funded by Foundation for Science and Technology Development of Ton Duc Thang University (FOSTECT), website: http://fostect.tdtu.edu.vn, under Grant FOSTECT.2017.BR.10

References

Argyriou, A., Evgeniou, T., & Pontil, M. (2008). Convex multi-task feature learning. *Machine Learing*, *73*, 243–272.

Bi, J., Xiong, T., Yu, S., Dundar, M., & Rao, R. B. (2008). An improved multi-task learning approach with applications in medical diagnosis. In *ECML PKDD* (pp. 117–132). volume 5211.

Blodel, M., Seki, K., & Uehara, K. (2013). Block coordinate descent algorithms for large-scale sparse multiclass classification. *Machine Learning*, *93*, 31–52.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundat. Trends Mach. Learn.*, *3*, 1–124.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In *Proceeding of international conference on machine learning ICML98*.

Calandriello, D., Lazaric, A., & Restelli, M. (2014). Sparse multi-task reinforcement learning. In *NIPS*.

Chen, J., & Huo, X. (2006). Theoretical results on sparse representations of multiple-measurement vectors. *IEEE Transactions on Signal Processing*, *54*, 4634–4643.

Clemmensen, L., Hansen, M., Ersboll, B., & Frisvad, J. (2007). A method for comparison of growth media in objective identification of penicillium based on multi-spectral imaging. *Journal of Microbiological Methods*, *69*, 249–255.

Clemmensen, L., Hastie, T., Witten, D., & Ersbll, B. (2011). Sparse discriminant analysis. *Technometrics*, *53*, 406–413.

Cotter, S. F., Rao, B. D., Engan, K., & Kreutz-Delgado, K. (2005). Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, *53*, 24772488.

Danaher, P., Wang, P., & Witten, D. M. (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *J. R. Statist. Soc. B*, *76*, 373–397.

Eksioglu, E. M. (2014). Group sparse rls algorithms. *International Journal of Adaptive Control and Signal Processing*, *28*, 1398–1412.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Stat. Ass.*, *96*, 1348–1360.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annal of Eugenics*, *7*, 179–188.

Friedman, J., Hastie, T., Hoefling, H., & Tibshirani, R. (2007). Pathwise coordinate optimization. *The Anals of Applied Statistics*, *1*, 302–332.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gasenbeek, M., Mesirov, J. P., Coler, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., & Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, *286*, 531–537.

Gu, Q., Li, Z., & Han, J. (2011). Linear discriminant dimensionality reduction. In *ECML PKDD* (pp. 549–564). volume 6911.

Hastie, T., Buja, A., & Tibshirani, R. (1995). Penalized discriminant analysis. *The annals of statistics*, *23*, 73–102.

Hastie, T., Tibshirani, R., & Buja, A. (1994). Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, *89*, 1255–1270.

Hu, Y., Li, C., Meng, K., Qin, J., & Yang, X. (2017). Group sparse optimization via lp,q regularization. *J. Mach. Learn. Res.*, *18*, 960–1011.

Huang, J., & Wei, S., F. Ma (2012). Semiparametric regression pursuit. *Statist Sinica*, *22*, 1403–1426.

Kha, Z., Shafait, F., & Mian, A. (2015). Joint group sparse pca for compressed hyperspectral imaging. *IEEE Trans. Ima. Process.*, *24*, 4934–4942.

Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schab, M., Antonescu, C. R., Peterson, C., & Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using expression profiling and artificial neural networks. *Nat. Med.*, *7*, 673–679.

Lan, X., Ma, A., Yuen, P., & Chellappa, R. (2015). Joint sparse representation robust feature-level fusion for multi-cue visual tracking. *IEEE Trans Image Process*, *24*, 5826–5841.

Le Thi, H. A., Le Hoai, M., Nguyen, V. V., & Pham Dinh, T. (2008). A DC Programming approach for Feature Selection in Support Vector Machines learning. *Journal of Advances in Data Analysis and Classification*, *2*, 259–278.

Le Thi, H. A., Le Hoai, M., & Pham Dinh, T. (2014a). Feature selection in machine learning: An exact penalty approachusing a difference of convex function algorithm. *Machine Learning*, .

Le Thi, H. A., & Pham Dinh, T. (2005). The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, *133*, 23–46.

Le Thi, H. A., & Pham Dinh, T. (2018). DC programming and DCA: thirty years of developments. *Mathematical Programming*, *169*, 5–68.

Le Thi, H. A., Pham Dinh, T., Le Hoai, M., & Vo Xuan, T. (2015). DC approximation approaches for sparse optimization. *European Journal of Operational Research*, *244*, 26–44.

Le Thi, H. A., & Phan, D. N. (2016a). DC programming and DCA for sparse fisher linear discriminant analysis. *Neural Computing and Applications*, . Doi:10.1007/s00521-016-2216-9.

Le Thi, H. A., & Phan, D. N. (2016b). DC programming and DCA for sparse optimal scoring problem. *Neurocomputing*, *186*, 170–181.

Le Thi, H. A., T. Vo Xuan, T., & T. Pham Dinh, T. (2014b). Feature selection for linear SVMs under uncertain data: robust optimization based on difference of convex functions Algorithms. *Neural Networks*, *59*, 36–50.

Le Thi (Homepage), H. A. (2005). DC programming and DCA: `http://www.lita.univ-lorraine.fr/~lethi/index.php/en/research/dc-programming-and-dca.html`.

Lee, S., Oh, M., & Kim, Y. (2016). Sparse optimization for nonconvex group penalized estimation. *Journal of Statistical Computation and Simulation*, *86*, 597–610.

Leng, C. (2008). Sparse optimal scoring for multiclass cancer diagnosis and biomarker detection using microarray data. *Computational Biology and Chemistry*, *32*, 417–425.

Liu, J., Ji, S., & Ye, J. (2009). Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *UAI*.

Merchante, L. F. S., Grandvalet, Y., & Govaert, G. (2012). An efficient approach to sparse linear discriminant analysis. In *ICML*.

Nie, F., Huang, H., Cai, X., & Ding, C. (2010). Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *NIPS*.

Obozinski, G., Taskar, B., & Jordan, M. (2006). *Multi-task feature selection*. Technical Report Department of Statistics, University of California, Berkeley.

Obozinski, G., Taskar, B., & Jordan, M. (2010). Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, *20*, 231–252.

Ong, C., & Le Thi, H. A. (2013a). Learning sparse classifiers with Difference of Convex functions Algorithms. *Optimization Methods and Software*, *28*, 830–854.

Ong, C. S., & Le Thi, H. A. (2013b). Learning sparse classifers with difference of convex cunctions algorithms. *Optimization Methods and Software*, *28*, 830–854.

Peleg, D., & Meir, R. (2008). A bilinear formulation for vector sparsity optimization. *Signal Processing*, *88*, 375–389.

Pham Dinh, T., & Le Thi, H. A. (1997). Convex analysis approach to D.C. programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, *22*, 289–355.

Pham Dinh, T., & Le Thi, H. A. (1998). A DC optimization algorithm for solving the trust-region subproblem. *SIAM Journal of Optimization*, *8*, 476–505.

Pham Dinh, T., & Le Thi, H. A. (2014). Recent advances in DC programming and DCA. *Transactions on Computational Collective Intelligence*, *8342*, 1–37.

Quattoni, A., Carreras, X., Collins, M., & Darrell, T. (2009). An efficient projection for $\ell_{\infty,1}$-regularization. In *ICML*.

Sun, L., Liu, J., Chen, J., & Ye, J. (2009). Efficient recovery of jointly sparse vectors. In *NIPS*.

Turlach, B. A., Venables, W. N., & Wright, S. J. (2005). Simultaneous variable selection. *Technometrics*, *47*, 349–363.

Wang, L., Chen, G., & Li, H. (2007). Group scad regression analysis for microarray time course gene expression data. *Bioinformatics*, *23*, 1486–1494.

Wei, F., & Huang, J. (2010). Consistent group selection in high-dimensional linear regression. *Bernoulli*, *16*, 1369–1384.

Wei, F., & Zhu, H. (2012). Group coordinate descent algorithms for nonconvex penalized regression. *Comput Statist Data Anal.*, *56*, 316–326.

Yeoh, E. J., Ross, M. E., Shurtleff, S. A., Williams, W. K., Patel, D., Mahfouz, R., Behm, F. G., Raimondi, S. C., Relling, M. V., Patel, A., & al. (2002). Classification, subtype discovery, and prediction of outcome in pediatric lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, *1*, 133–143.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. B*, *68*, 49–67.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.*, *38*, 894–942.

Zhang, H. H., Liu, Y., Wu, Y., & Zhu, J. (2008). Variable selection for the multicategory SVM via adaptive sup-norm regularization. *Electron. J. of Stat.*, *2*, 149–167.

Zhang, Y., Yeung, D. Y., & Xu, Q. (2010). Probabilistic multi-task feature selection. In *NIPS*.

Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical statistics*, *15*, 265–286.