DCA based algorithms for learning with sparsity in high dimensional setting and stochastical learning

Duy Nhat Phan

THÈSE

en vue de l'obtention du titre de

DOCTEUR DE L'UNIVERSITÉ DE LORRAINE

(arrêté ministériel du 7 Août 2006)

Spécialité Mathématiques appliquées

présentée par

Phan Duy Nhat

Titre de la thèse :

Algorithmes basés sur la programmation DC et DCA pour l'apprentissage avec la parcimonie et l'apprentissage stochastique en grande dimension

—

DCA based algorithms for learning with sparsity in high dimensional setting and stochastical learning

soutenue le 15 décembre 2016

Composition du Jury :

Rapporteurs	HEIN Matthias	*Professeur, Université Saarland*
	YASSINE Adnan	*Professeur, Université du Havre*
Examinateurs	CANU Stéphane	*Professeur, Laboratoire LITIS-INSA de Rouen*
	D'ASPREMONT Alexandre	*Professeur, École Normale Supérieure, Paris*
	GUERMEUR Yann	*Directeur de recherche, CNRS Loria*
	PHAM DINH Tao	*Professeur émérite, INSA de Rouen*
Directrice de thèse	LE THI Hoai An	*Professeur, Université de Lorraine*

Thèse préparée au sein de laboratoire
d'Informatique Théorique et Appliquée (LITA)
Université de Lorraine, Metz, France

Remerciements

Cette thèse a été réalisée au sein du Laboratoire d'Informatique Théorique et Appliquée (LITA) de l'Université de Lorraine, sous la direction de Madame le Professeur LE THI Hoai An, Directrice du LITA, Université de Lorraine.

Je souhaite en premier lieu exprimer ma profonde gratitude à Madame le Professeur LE THI Hoai An, pour la confiance qu'elle m'a accordée en acceptant d'encadrer cette thèse de doctorat. Elle a dirigé, avec une grande aisance, mes travaux de recherche avec une grande patience, rigueur et enthousiasme. Sa pédagogie et ses capacités dans la théorie fondamentale et appliquée sur un large spectre de domaines d'applications m'ont guidé sur le chemin de la réussite. Je lui suis très reconnaissant pour ses conseils, ses encouragements, ses soutiens permanents et ses aides très précieuses tant sur le plan professionnel que personnel pendant plus de trois ans. Sans eux, je ne serais certainement pas en mesure d'atteindre mon objectif. Je vais garder dans mon cœur tout le meilleur qu'elle m'a accordé. Je profite de cette riche expérience pour développer mes recherches dans l'avenir.

J'adresse respectueusement mes sincères remerciements à Monsieur PHAM DINH Tao, professeur à l'INSA de Rouen pour ses conseils, et son suivi dans mes travaux de recherche. Je voudrais lui exprimer toute ma reconnaissance pour les discussions approfondies très intéressantes que nous avons eues et pour m'avoir suggéré de nouvelles voies de recherche.

Je voudrais remercier vivement Monsieur HEIN Matthias, professeur à l'Université Saarland et Monsieur YASSINE Adnan, professeur à l'Université du Havre, de m'avoir fait l'honneur d'accepter d'être rapporteurs de ma thèse, et de participer à sa soutenance.

Je souhaite également remercier Monsieur CANU Stéphane, professeur à l'INSA de Rouen, Monsieur D'ASPREMONT Alexandre, professeur à l'École Normale Supérieure Paris, et Monsieur GUERMEUR Yann, directeur de recherche au Loria Nancy, d'avoir bien voulu accepter de juger mon travail.

Je tiens à remercier particulièrement Madame le Professeur LE THI Hoai Chau, pour sa confiance et sa recommandation, ce qui m'a donné une chance de rencontrer et travailler avec Madame le Professeur LE THI Hoai An.

Je remercie particulièrement le Docteur NGUYEN Manh Cuong pour les discussions intéressantes que nous avons eues lors de notre collaboration.

Mes remerciements s'adressent également au Gouvernement Vietnamien qui a financé mes études pendant trois ans. Je n'oublie pas de remercier l'Ecole supérieure de Pédagogie de Ho Chi Minh ville, Vietnam pour son soutien.

Un grand merci à mes collègues du LITA et à mes amis de Metz pour leur aide et leur soutien, ainsi que pour les moments agréables partagés lors de mon séjour en France. J'associe à ces remerciements tous les grands amis et collègues, dans l'ordre alphabétique, Tran Bach, Minh Tam, Hoai Minh, Vinh Thanh, Xuan Thanh, Bich Thuy, Minh Thuy, Tran Thuy, Anh Vu, ...

Je voudrais exprimer mes remerciements particuliers à ma femme, Thuy Ngoc, pour son soutien et sa patience. J'adresse toute mon affection à mes parents et à tous les membres de ma famille.

Enfin à tous ceux qui m'ont soutenu de près ou de loin, et à tous ceux qui m'ont incité même involontairement, à faire mieux, veuillez trouver ici le témoignage de ma profonde gratitude.

Résumé

De nos jours, avec l'abondance croissante de données de très grande taille, les problèmes de classification de grande dimension ont été mis en évidence comme un challenge dans la communauté d'apprentissage automatique et ont beaucoup attiré l'attention des chercheurs dans le domaine. Au cours des dernières années, les techniques d'apprentissage avec la parcimonie et l'optimisation stochastique se sont prouvées être efficaces pour ce type de problèmes. Dans cette thèse, nous nous concentrons sur le développement des méthodes d'optimisation pour résoudre certaines classes de problèmes concernant ces deux sujets. Nos méthodes sont basées sur la programmation DC (Difference of Convex functions) et DCA (DC Algorithm) étant reconnues comme des outils puissants d'optimisation non convexe.

La thèse est composée de trois parties. La première partie aborde le problème de la sélection des variables. La deuxième partie étudie le problème de la sélection de groupes de variables. La dernière partie de la thèse liée à l'apprentissage stochastique.

Dans la première partie, nous commençons par la sélection des variables dans le problème discriminant de Fisher (Chapitre 2) et le problème de scoring optimal (Chapitre 3), qui sont les deux approches différentes pour la classification supervisée dans l'espace de grande dimension, dans lequel le nombre de variables est beaucoup plus grand que le nombre d'observations. Poursuivant cette étude, nous étudions la structure du problème d'estimation de matrice de covariance parcimonieuse et fournissons les quatre algorithmes appropriés basés sur la programmation DC et DCA (Chapitre 4). Deux applications en finance et en classification sont étudiées pour illustrer l'efficacité de nos méthodes.

La deuxième partie étudie la $\ell_{p,0}$-régularisation pour la sélection de groupes de variables (Chapitre 5). En utilisant une approximation DC de la $\ell_{p,0}$-norme, nous prouvons que le problème approché, avec des paramètres appropriés, est équivalent au problème original. Considérant deux reformulations équivalentes du problème approché, nous développons différents algorithmes basés sur la programmation DC et DCA pour les résoudre. Comme applications, nous mettons en pratique nos méthodes pour la sélection de groupes de variables dans les problèmes de scoring optimal et d'estimation de multiples matrices de covariance.

Dans la troisième partie de la thèse, nous introduisons un DCA stochastique pour des problèmes d'estimation des paramètres à grande échelle (Chapitre 6) dans lesquelles la

fonction objectif est la somme d'une grande famille des fonctions non convexes. Comme une étude de cas, nous proposons un schéma DCA stochastique spécial pour le modèle log-linéaire incorporant des variables latentes.

Abstract

These days with the increasing abundance of data with high dimensionality, high dimensional classification problems have been highlighted as a challenge in machine learning community and have attracted a great deal of attention from researchers in the field. In recent years, sparse and stochastic learning techniques have been proven to be useful for this kind of problem. In this thesis, we focus on developing optimization approaches for solving some classes of optimization problems in these two topics. Our methods are based on DC (Difference of Convex functions) programming and DCA (DC Algorithms) which are well-known as one of the most powerful tools in optimization.

The thesis is composed of three parts. The first part tackles the issue of variable selection. The second part studies the problem of group variable selection. The final part of the thesis concerns the stochastic learning.

In the first part, we start with the variable selection in the Fisher's discriminant problem (Chapter 2) and the optimal scoring problem (Chapter 3), which are two different approaches for the supervised classification in the high dimensional setting, in which the number of features is much larger than the number of observations. Continuing this study, we study the structure of the sparse covariance matrix estimation problem and propose four appropriate DCA based algorithms (Chapter 4). Two applications in finance and classification are conducted to illustrate the efficiency of our methods.

The second part studies the $\ell_{p,0}$ regularization for the group variable selection (Chapter 5). Using a DC approximation of the $\ell_{p,0}$-norm, we indicate that the approximate problem is equivalent to the original problem with suitable parameters. Considering two equivalent reformulations of the approximate problem we develop DCA based algorithms to solve them. Regarding applications, we implement the proposed algorithms for group feature selection in optimal scoring problem and estimation problem of multiple covariance matrices.

In the third part of the thesis, we introduce a stochastic DCA for large scale parameter estimation problems (Chapter 6) in which the objective function is a large sum of nonconvex components. As an application, we propose a special stochastic DCA for the log-linear model incorporating latent variables.

PHAN Duy Nhat

Né le 08 Novembre, 1985 (Viet Nam)

Tél: 07 83 98 50 40

E-mail: nhatsp@gmail.com

Adresse personnelle: P5212, Res Univ Saulcy, Ile du Saulcy, 57010 Metz

Adresse professionnelle: Bureau E425, LITA – Université de Lorraine, Ile du Saulcy, 57045 Metz

Situation Actuelle

Depuis Septembre 2013	Doctorant au Laboratoire d'Informatique Théorique et Appliquée (LITA EA 3097) de l'Université de Lorraine. Encadré par Prof. Le Thi Hoai An. Sujet de thèse : **"Algorithmes basés sur la programmation DC et DCA pour l'apprentissage avec la parcimonie et l'apprentissage stochastique en grande dimension"**

Experience Professionnelle

09/2010–09/2013	Enseignant, Ecole supérieure de Pédagogie de Ho Chi Minh Ville, Vietnam.

Diplôme et Formation

2013 au present	Doctorant en Mathématiques appliquées. LITA–Université de Lorraine, Metz.
2009-2010	Master 2 en Mathématiques, Université de Strasbourg, France.
2008-2009	Master 1 en Mathématiques, Ecole normale supérieure de Hanoi, Vietnam.
2004–2008	Diplôme universitaire en Mathématique et Informatiques, Ecole supérieure de Pédagogie de Ho Chi Minh Ville, Vietnam.

Publications

Refereed international journal papers

[1] Hoai An Le Thi and Duy Nhat Phan. DC Programming and DCA for Sparse Fisher Linear Discriminant Analysis. Neural Computing and Applications (2016), doi: 10.1007/s00521-016-2216-9.

[2] Hoai An Le Thi and Duy Nhat Phan. DC Programming and DCA for Sparse Optimal Scoring Problem. Neurocomputing, 186: 170-181 (2016).

[3] Duy Nhat Phan and Hoai An Le Thi and Tao Pham Dinh. Sparse Covariance Matrix Estimation by DCA based Algorithms. *Submitted.*

[4] Hoai An Le Thi and Duy Nhat Phan. Efficient Nonconvex Group Variable Selection and Application to Group Sparse Optimal Scoring. *Submitted.*

[5] Hoai An Le Thi and Duy Nhat Phan. DC Programming and DCA for Bi-level Variable Selection in Estimation of Multiple Covariance Matrices. *In preparation.*

[6] Hoai An Le Thi and Duy Nhat Phan. Stochastic DCA and Application to Latent Log-Linear Model. *In preparation.*

Refereed papers in books / Refereed international conference papers

[1] Duy Nhat Phan, Manh Cuong Nguyen and Hoai An Le Thi. A DC Programming Approach for Sparse Linear Discriminant Analysis. Chapter in Advanced Computational Methods for Knowledge Engineering, Advances in Intelligent Systems and Computing, Volume 282, pp. 65-74, Springer (2014).

[2] Hoai An Le Thi and Duy Nhat Phan. A DC Programming Approach for Sparse Optimal Scoring. Chapter in Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, Volume 9078, pp. 435-446, Springer (2015).

[3] Duy Nhat Phan, Hoai An Le Thi and Tao Pham Dinh. A DC Programming Approach for Sparse Estimation of a Covariance Matrix. Chapter in Modelling Computation an Optimization in Information Systems and Management Sciences, Advances in Intelligent Systems and Computing, Volume 359, pp. 131-142, Springer (2015).

[4] Duy Nhat Phan and Hoai An Le Thi. Efficient Bi-level Variable Selection and Application to Estimation of Multiple Covariance Matrices. *Submitted.*

Communications in national / International conferences

[1] Duy Nhat Phan and Hoai An Le Thi. DC Programming and DCA for Sparse Discriminant Analysis. Presentation at the conference IFORS 2014, Barcelona, Spain, 13-18 July, 2014.

[2] Hoai An Le Thi and Duy Nhat Phan. Parameter Estimation for Latent Log-Linear Models as DC Programming. Presentation at the conference EURO 2015, Glasgow, UK, 12-15 July, 2015.

Contents

List of Figures

List of Tables

Notation

Throughout the thesis, we use uppercase letters to denote matrices, and lowercase letters for vectors or scalars. Vectors are also regarded as matrices with one column. The table below summarizes some of the notation used in the thesis.

$\mathbb{R}$	set of real numbers
$\mathbb{R}_+$	set of nonnegative real numbers
$\mathbb{R}_{++}$	set of positive real numbers
$\mathbb{R}^n$	set of real column vectors of size n
$\mathbb{R}^{m\times n}$	set of real matrices of size m - by - n
$\mathbb{R}^n_+$	set of nonnegative real column vectors of size n
$\mathbb{R}^n_{++}$	set of positive real column vectors of size n
I_n	identity matrix of size n
$\|\cdot\|_p$	ℓ_p-norm ($0 < p < \infty$), $\|x\|_p = (\sum_{i=1}^n \|x_i\|^p)^{1/p}$, $x \in \mathbb{R}^n$
$\|\cdot\|$	vector ℓ_2-norm/Euclidean norm, $\|x\| = (\sum_{i=1}^n \|x_i\|^2)^{1/2}$, $x \in \mathbb{R}^n$
	matrix ℓ_2-norm/spectral norm, $\|X\| = \max_{u\in\mathbb{R}^n, \|u\|=1} \|Xu\|$, $X \in \mathbb{R}^{m\times n}$
$\|\cdot\|_0$	ℓ_0-'norm', $\|x\|_0 = \|\{i : x_i \neq 0\}\|$, $\|X\|_0 = \|\{(i,j) : X_{ij} \neq 0\}\|$
$\|\cdot\|_F$	Frobenius norm, $\|X\|_F = (\sum_{i=1}^m \sum_{j=1}^n X_{ij}^2)^{1/2}$, $X \in \mathbb{R}^{m\times n}$
$\langle\ ,\ \rangle$	scalar product, $\langle X, Y\rangle = \sum_{i=1}^m \sum_{j=1}^n X_{ij}Y_{ij}$, $X, Y \in \mathbb{R}^{m\times n}$

X_{ij}	element located at the position (i,j) of X
X_{IJ}	submatrix of X with row (resp. column) indices in I (resp. J)
X^T	transpose of a matrix X, $(X^T)_{ij} = X_{ji}$
$\|X\|$	absolute of X, $\|X\|_{ij} = \|X_{ij}\|$ for all (i,j)
$\mathrm{sgn}(X)$	matrix of signs of X, $(\mathrm{sgn}(X))_{ij} = \mathrm{sgn}(X_{ij}) = -1$ if $X_{ij} < 0$, 1 if $X_{ij} > 0$, and 0 otherwise
$\mathrm{diag}(x)$	diagonal matrix whose the main diagonal is the vector x
$\lambda_{\min}(X)$	the smallest eigenvalue of X
$\lambda_{\max}(X)$	the largest eigenvalue of X
X^{-1}	inverse of the matrix X
$\det(X)$	determinant of the matrix X
$\mathrm{tr}(X)$	trace of the matrix $X \in \mathbb{R}^{n\times n}$, $\mathrm{tr}(X) = \sum_{i=1}^n X_{ii}$

$X \circ Y$	Hadamard product between matrices X and Y, $(X \circ Y)_{ij} = X_{ij}Y_{ij}$
$X \otimes Y$	Kronecker product between matrices X and Y
$X \succ 0$	X is symmetric positive definite
$X \preceq Y$	$Y - X$ is positive semi-definite matrix (all eigenvalues are nonnegative)
$(x)_+$	positive part of x, $(x)_+ = x$ if $x > 0$ and 0 otherwise
$\mathcal{S}(X, Y)$	soft-thresholding operator, $\mathcal{S}(X, Y)_{ij} = \operatorname{sgn}(X_{ij}) \left(\lvert X_{ij}\rvert - Y_{ij}\right)_+$

$\chi_C(\cdot)$	the indicator function of C, $\chi_C(x) = 0$ if $x \in C$ and $+\infty$ otherwise
$\nabla f(x)$	the gradient of f at x
$\nabla^2 f(x)$	the Hessian of f at x
$\partial f(x)$	the subdifferential of f at x

Introduction générale

Cadre général et nos motivations

L'émergence d'Internet ainsi que la croissance rapide de la science et de la technologie au cours de ces dernières années ont stimulé l'énorme volume des ressources d'informations disponibles, et il est encore en croissance à un rythme incroyablement rapide. Sans surprise, nous somme confrontés à l'immense quantité de données généralement appelées Big Data. Ainsi, les méthodes traditionnelles d'apprentissage et de fouille de données (Machine Learning and Data Mining - MLDM) deviennent inefficaces pour le traitement de ce genre de données. D'où, il est absolument nécessaire de développer des méthodes efficaces et robustes.

Dans cette thèse, nous nous concentrons sur deux challenges en MLDM dans le contexte du big data: apprentissage avec la parcimonie sur des données de très grande dimensions et apprentissage stochastique sur une énorme quantité de données. L'apprentissage avec la parcimonie permet d'avoir une meilleure interprétation et de réduise "overfitting" en supprimant les redondantes variables. Pour la conception de modèles d'apprentissage, la modélisation parcimonieuse est basée sur la norme zéro (la norme zéro d'un vecteur est définie comme le nombre de ses termes non nulles). C'est la façon la plus naturelle pour aborder la sélection des variables en MLDM, mais le problème d'optimisation correspondant est NP-difficile. C'est pourquoi, dans ces travaux, les problèmes d'optimisation incluent des doubles difficultés. En effet, la première est de savoir comment traiter la norme zéro et la seconde est causée par la non-convexité des problèmes originaux. La difficulté de la norme zéro peut être surmontée par son approximation via une fonction DC (Difference of Convex functions). Le problème résultant est encore difficile, mais il possède des propriétés intéressantes et peut être ainsi résolu par les méthodes basées sur l'optimisation DC. En outre, lorsque les données possèdent certaines structures de groupe, nous sommes naturellement intéressés à la sélection de groupes importants de variables plutôt que des individus. La régularisation générale est proposée pour obtenir la parcimonie groupée. Egalement, la difficulté de cette régularisation peut être surmontée en utilisant une approximation DC appropriée, le problème résultant est donc un problème d'optimisation DC. Enfin, la présence de techniques d'optimisation stochastiques est justifié être efficaces en MLDM pour résoudre des problèmes avec un grand nombre de points de données d'entraînement. Mais le véritable challenge est des problèmes d'estimation des

paramètres non-convexes dans lesquelles la fonction objectif est la somme d'une grande famille des fonctions DC. Par conséquent, DCA stochastique est introduite afin d'obtenir un faible coût de calcul à chaque itération.

Sur le plan algorithmique, la thèse a proposé une approche unifiée, fondée sur la programmation DC et DCA, des outils puissant d'optimisation non convexe qui connaît un grand succès, au cours de deux dernières décennies, dans la résolution de nombreux problèmes d'application dans diverses domaines de sciences appliquées, en particulier en MLDM. De nombreuses expérimentations numériques sur différents types de données (biologie, image, finace, ...) réalisées dans cette thèse ont prouvé l'efficacité, la scalabilité, la rapidité des algorithmes proposés et leur supériorité par rapports aux méthodes standards.

La programmation DC et DCA considèrent le problème DC de la forme

$$\alpha = \inf\{f(x) := g(x) - h(x) \ : \ x \in \mathbb{R}^n\} \quad (P_{dc}),$$

où g et h sont des fonctions convexes définies sur $\mathbb{R}^n$ et à valeurs dans $\mathbb{R} \cup \{+\infty\}$, semi-continues inférieurement et propres. La fonction f est appelée fonction DC avec les composantes DC g et h, et $g - h$ est une décomposition DC de f. DCA est basé sur la dualité DC et des conditions d'optimalité locale. La construction de DCA implique les composantes DC g et h et non la fonction DC f elle-même. Or chaque fonction DC admet une infinité des décompositions DC qui influencent considérablement sur la qualité (la rapidité, l'efficacité, la globalité de la solution obtenue,...) de DCA. Ainsi, au point de vue algorithmique, la recherche d'une "bonne" décomposition DC et d'un "bon" point initial est très importante dans le développement de DCA pour la résolution d'un programme DC.

L'utilisation de la programmation DC et DCA dans cette thèse est justifiée par de multiple arguments (Pham Dinh and Le Thi, 2014):

- La programmation DC et DCA fournissent un cadre très riche pour les problèmes de MLDM: MLDM constituent *une mine des programmes DC* dont la résolution appropriée devrait recourir à la programmation DC et DCA. En effet la liste indicative (non exhaustive) des références dans Le Thi (Website) témoigne de la vitalité la puissance et la percée de cette approche dans la communauté de MLDM.
- DCA est une philosophie plutôt qu'un algorithme. Pour chaque problème, nous pouvons concevoir une famille d'algorithmes basés sur DCA. La flexibilité de DCA sur le choix des décomposition DC peut offrir des schémas DCA plus performants que des méthodes standard.
- L'analyse convexe fournit des outils puissants pour prouver la convergence de DCA dans un cadre général. Ainsi tous les algorithmes basés sur DCA bénéficient (au moins) des propriétés de convergence générales du schéma DCA générique qui ont été démontrées.
- DCA est une méthode efficace, rapide et scalable pour la programmation non convexe. A notre connaissance, DCA est l'un des rares algorithmes de la programmation non convexe, non différentiable qui peut résoudre des programmes DC de très grande dimension. La programmation DC et DCA ont été appliqués avec succès pour la modélisation DC et la résolution de nombreux et divers problèmes d'optimisation non convexes dans

différents domaines des sciences appliquées, en particulier en MLDM (voir par exemple la liste des références dans Le Thi (Website)).
Il est important de noter qu'avec les techniques de reformulation en programmation DC et les décompositions DC appropriées, on peut retrouver la plupart des algorithmes existants en programmation convexe/non convexe comme cas particuliers de DCA.

En particulier, pour la communauté de MLDM, les méthodes très connus comme Expectation–Maximisation (EM) (Dempster et al., 1977), Succesive Linear Approximation (SLA) (Bradley and Mangasarian, 1998), ConCave–Convex Procedure (CCCP) (Yuille and Rangarajan, 2003), Iterative Shrinkage-Thresholding Algorithms (ISTA) (Chambolle et al., 1998) sont des versions spéciaux de DCA.

Nos contributions

La thèse vise à développer de nouveaux modèles et méthodes pour cinq classes de problèmes difficiles et importants en MLDM: analyse discriminante linéaire de Fisher parcimonieuse, scoring optimal parcimonieuse, estimation de matrice de covariance parcimonieuse, sélection de groupes de variables et des applications dans scoring optimal et estimation de matrices de covariance, estimation des paramètres du modèle à variables latentes. Nous commençons par décrire brièvement les principales réalisations de la thèse.

Dans le premier temps, nous considérons le problème de la classification supervisée dans l'espace de grande dimension, dans lequel le nombre de variables est beaucoup plus grand que le nombre d'observations. Dans de nombreuses applications telles que la recherche d'information, la reconnaissance faciale et l'analyse des microarrays, nous rencontrons souvent ce genre de problème. Parmi plusieurs méthodes de classification dans la littérature, l'analyse discriminante linéaire (LDA) est considérée comme l'une des méthodes les plus populaires en raison de son avantage de la réduction de dimension. L'objectif principal de LDA est de trouver une transformation linéaire qui distingue au mieux les différentes classes. La classification est alors effectuée dans l'espace transformé en utilisant des mesures de distance. Il existe trois approches différentes pour aborder LDA, qui sont basées sur la résolution du modèle normal, problème discriminant de Fisher et le problème de scoring optimal, respectivement. Nous développons donc une nouvelle approche pour la sélection des variables pour LDA basée sur le problème discriminant de Fisher et la norme zéro. Pour aborder la norme zéro, nous étudions les approches d'approximation DC. Parmi plusieurs fonctions induisant de la parcimonie existantes, nous utilisons le Capped-ℓ_1 et la fonction exponentielle concave par morceaux. Le choix du Capped-ℓ_1 est motivé par ses avantages tant sur le plan théorique qu'algorithmique. De plus, la fonction exponentielle concave par morceaux a été montrée pour être efficace via des résultats numériques dans de nombreux travaux. Les problèmes résultants sont formulés sous forme de programmes DC, puis quatre shémas de DCA sont proposés. Les résultats expérimentaux sur les deux données réelles et simulées démontrent l'efficacité des algorithmes proposés par rapport aux certaines méthodes standards.

Le deuxième problème abordé dans cette thèse est la sélection des variables dans le

scoring optimal en utilisant la régularisation $\ell_2+\ell_0$, appelé le problème de scoring optimal parcimonieuse (SOS). La résolution de SOS comprend des doubles difficultés. La première est la façon de traiter la norme zéro et la seconde est causée par la non-convexité du problème original de scoring optimal. La difficulté de la norme zéro est surmontée en utilisant deux approximations DC. Les problèmes d'optimisation résultants sont encore difficiles, mais ils possèdent des propriétés intéressantes: quand w_k est fixé, la solution optimale du problème par rapport à la variable θ_k peut être calculée explicitement. Pour chaque θ_k fixé, nous sommes confrontés à un programme DC par rapport à la variable w_k, donc nous sommes suggérés d'utiliser des schémas alternatifs basés sur DCA pour les résoudre. Nous prouvons que les principaux algorithmes convergent vers un point critique des problèmes approchés. Les performances des algorithmes proposés sont soigneusement examinées en les comparant avec sept méthodes standards sur tous les deux données simulées et réelles de grande dimension.

Toujours dans le cadre de la sélection des variables, nous considérons le troisième problème - estimation de matrice de covariance parcimonieuse (SCME). L'objectif est d'estimer une matrice de covariance parcimonieuse sur la base d'un échantillon de vecteurs Gaussiens. Beaucoup d'analyses statistiques de données de grande dimension exige l'estimation d'une matrice de covariance ou son inverse, telles que la gestion de portefeuille et de l'évaluation des risques, l'analyse de l'indépendance et des relations d'indépendance conditionnelle entre les composants dans les modèles graphiques, analyse en composantes principales, et ainsi de suite. Le Capped-ℓ_1 et la fonction exponentielle concave par morceaux sont choisis de nouveau pour la modélisation parcimonieuse, cependant nous sommes toujours confrontés à la difficulté de la non-convexité de la fonction log-vraisemblance négative. Ainsi, nous proposons deux formulations DC du problème approché SCME basé sur deux décompositions DC de sa fonction objectif. Le premier résultat est obtenu à partir d'une décomposition DC naturelle tandis que le second est introduit pour exploiter de beaux effets de décompositions DC. La complexité des deux schémas DCA correspondantes est sensiblement différente. Selon nos expériences numériques, le rapport de gain en temps de calcul entre les deux DCA est de 44 fois. En appliquant DCA à deux formulations DC avec deux approximations, nous avons alors quatre algorithmes basés sur DCA pour le problème approché SCME. Les résultats d'analyse de convergence spéciale de nos algorithmes sont fournis. En outre, nous considérons deux applications importantes du problème SCME dans nos expériences, qui sont respectivement l'analyse discriminante quadratique en utilisant des matrices de covariance parcimonieuses estimées par les algorithmes proposés et le problème d'optimisation de portefeuille.

Le quatrième problème abordé dans la thèse est la sélection de groupes de variables. La nécessité de sélectionner des groupes de variables se pose dans de nombreux domaines d'application tels que l'apprentissage, le statistique, la biologie computationnelle, le traitement du signal, et d'autres domaines connexes. Nous étudions la $\ell_{p,0}$-régularisation pour obtenir la parcimonie groupée. En utilisant une approximation DC approprié du $\ell_{p,0}$-norme, nous indiquons que le problème approché est équivalent au problème original avec les paramètres appropriés. En considérant deux reformulations équivalentes du problème approché, nous développons des algorithmes basés sur DCA pour les résoudre. Lorsque $p = 1$ (resp. $p = 2$), nos algorithmes comprennent un algorithme de ℓ_1 perturbé (resp.

algorithme de $\ell_{2,1}$ perturbé) et un algorithme de ℓ_1 repondéré (resp. algorithme de $\ell_{2,1}$ repondéré). Il se trouve que, parmi les $\ell_{p,0}$-régularisations, $\ell_{1,0}$ est la régularisation la plus intéressante avec plusieurs avantages dans tous les deux aspects théoriques et computationels. En ce qui concerne les applications, nous appliquons les algorithmes proposés à la sélection de groupes de variables dans les problèmes de scoring optimal et estimation de matrices de covariance. Dans la première application, la parcimonie est obtenue en utilisant la $\ell_{p,0}$-régularisation qui sélectionne les mêmes variables dans tous les vecteurs discriminants. Les vecteurs discriminants parcimonieux résultant fournissent une représentation de faible dimension plus interprétable des données. Dans la seconde application, les matrices de covariance partagent certaines structures communes telles que les emplacements ou les poids des éléments non nuls, nous combinons la ℓ_0-norme et la $\ell_{p,0}$-norme pour obtenir la parcimonie sur chaque matrice de covariance et à travers de multiples matrices de covariance, respectivement.

Finalement, nous analysons et appliquons le technique stochastique basé sur la programmation DC et DCA aux problèmes d'estimation des paramètres à grande échelle dans lesquelles la fonction objectif est la somme d'une grande famille des fonctions DC. A chaque itération, nous utilisons seulement un petit sous-ensemble des fonctions DC et exécutons une itération du programme DC correspondante. Comme une application, nous étudions la structure du modèle log-linéaire latent et proposons un schéma DCA stochastique spécial dans lequel la solution à chaque sous-problème convexe peut être explicitement calculée. Nous également étudions la programmation DC et DCA pour résoudre directement le modèle log-linéaire latent. Les expériences numériques montrent que nos algorithmes proposés offrent une bonne performance.

Organisation de la thèse

La thèse se compose en sept chapitres. Le premier chapitre décrit de manière succincte la programmation DC et DCA. Il présente les outils théoriques et algorithmiques servant des références aux autres chapitres. Les cinq chapitres suivants sont divisés en trois parties: la première partie (Chapitres 2, 3 et 4) aborde le problème de la sélection des variables. Plus précisément, nous présentons les approches basées sur DCA pour la sélection des variables dans le problème discriminant de Fisher (Chapitre 2), le problème de scoring optimal (Chapitre 3), et le dernier chapitre de cette partie (Chapitre 4) est consacré au problème d'estimation de matrice de covariance parcimonieuse. Ensuite, la deuxième partie (Chapitre 5), nous étudions le problème de sélection de groupes de variables et des applications dans les problèmes de scoring optimal et d'estimation de multiples matrices de covariance. La dernière partie traite un modèle à variables latentes en introduisant un DCA stochastique appliqué au modèle log-linéaire incorporant des variables latentes (Chapitre 6). Chapitre 7 fournit les conclusions et les perspectives de nos travaux.

Chapter 1

DC programming and DCA

This chapter summarizes some basis concepts and results that will be the groundwork of the thesis.

DC programming and DCA, which constitute the backbone of nonconvex programming and global optimization, were introduced by Pham Dinh Tao in their preliminary form in 1985. Important developments and improvements on both theoretical and computational aspects have been completed since 1993 throughout the joint works of Le Thi Hoai An and Pham Dinh Tao. In this section, we present some basic properties of convex analysis and DC optimization and DC Algorithm that computational methods of this thesis are based on. The materials of this section are extracted from (Le Thi, 1994; Pham Dinh and Le Thi, 1997; Le Thi and Pham Dinh, 2005).

Throughout this section, X denotes the Euclidean space $\mathbb{R}^n$ and $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ is the set of extended real numbers.

1.1 Fundamental convex analysis

A subset C of X is said to be *convex* if $(1-\lambda)x+\lambda y \in C$ whenever $x, y \in C$ and $\lambda \in [0,1]$.

Let f be a function whose values are in $\overline{\mathbb{R}}$ and whose domain is a subset S of X. The set

$$\{(x,t) : x \in S, t \in R, f(x) \leq t\}$$

is called the *epigraph* of f and is denoted by epi$f$.

We define f to be a *convex function* on S if epi$f$ is convex set in $X \times \mathbb{R}$. This is equivalent to that S is convex and

$$f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y), \quad \forall x, y \in S, \forall \lambda \in [0,1]$$

The function f is *strictly convex* if the inequality above holds strictly whenever x and y are distinct in S and $0 < \lambda < 1$.

The *effective domain* of a convex function f on S, denoted by $\mathrm{dom}f$, is the projection on X of the epigraph of f

$$\mathrm{dom}f = \{x : \exists t \in \mathbb{R}, (x,t) \in \mathrm{epi}f\} = \{x \mid f(x) < +\infty\}$$

and it is convex.

The convex function f is called *proper* if $\mathrm{dom}f \neq \emptyset$ and $f(x) > -\infty$ for all $x \in S$.

The function f is said to be *lower semi-continuous* at a point x of S if

$$f(x) \leq \liminf_{y \to x} f(y)$$

Denote by $\Gamma_0(X)$ the set of all proper lower semi-continuous convex function on X.

Let $\rho \geq 0$ and C be a convex subset of X. One says that a function $\theta : C \mapsto \mathbb{R} \cup \{+\infty\}$ is ρ–convex if

$$\theta[\lambda x + (1-\lambda)y] \leq \lambda\theta(x) + (1-\lambda)\theta(y) - \frac{\lambda(1-\lambda)}{2}\rho\|x-y\|^2$$

for all $x, y \in C$ and $\lambda \in (0,1)$. It amounts to say that $\theta - (\rho/2)\|.\|^2$ is convex on C. The modulus of strong convexity of θ on C, denoted by $\rho(\theta, C)$ or $\rho(\theta)$ if $C = X$, is given by

$$\rho(\theta, C) = \sup\{\rho \geq 0 : \theta - (\rho/2)\|.\|^2 \text{ is convex on } C\}$$

One say that θ is *strongly convex* on C if $\rho(\theta, c) > 0$.

A vector y is said to be a *subgradient* of a convex function f at a point x^0 if

$$f(x) \geq f(x^0) + \langle x - x^0, y\rangle, \qquad \forall x \in X$$

The set of all subgradients of f at x^0 is called the *subdifferential* of f at x^0 and is denoted by $\partial f(x^0)$. If $\partial f(x)$ is not empty, f is said to be *subdifferentiable* at x.

We also have notations

$$\mathrm{dom}\ \partial f = \{x \in X : \partial f(x) \neq \emptyset\} \ \text{ and } \ \mathrm{range}\ \partial f(x) = \cup\{\partial f(x) : x \in \mathrm{dom}\ \partial f\}$$

Proposition 1.1 *Let f be a proper convex function. Then*

1. $\mathrm{ri}(\mathrm{dom}f) \subset \mathrm{dom}\ \partial f \subset \mathrm{dom}f$
   where $\mathrm{ri}(\mathrm{dom}f)$ *stands for the* relative interior *of* $\mathrm{dom}f$.
2. *If f has a unique subgradient at x, then f is differentiable at x, and $\partial f(x) = \{\nabla f(x)\}$.*
3. $x_0 \in \mathrm{argmin}\{f(x) : x \in X\}$ *if and only if* $0 \in \partial f(x_0)$.

Conjugates of convex functions

The *conjugate* of a function $f : X \mapsto \overline{\mathbb{R}}$ is the function $f^* : X \mapsto \overline{\mathbb{R}}$ defined by

$$f^*(y) = \sup_{x \in X}\{\langle x, y\rangle - f(x)\}$$

Proposition 1.2 *Let $f \in \Gamma_0(X)$. Then we have*

1. *$f^* \in \Gamma_0(X)$ and $f^{**} = f$.*
2. *$f(x) + f^*(y) \geq \langle x, y\rangle$, for any $x, y \in X$.*
   Equality holds if and only if $y \in \partial f(x) \Leftrightarrow x \in \partial f^*(y)$.

Polyhedral Functions

A *polyhedral* set is a closed convex set having form

$$C = \{x \in X : \langle x, b_i\rangle \leq \beta_i,\ \forall i = 1, \ldots, m\},$$

where $b_i \in X$ and $\beta_i \in \mathbb{R}$ for all $i = 1, \ldots, m$.

A function $f \in \Gamma_0(X)$ is said to be *polyhedral* if

$$f(x) = \max\{\langle a_i, x\rangle - \alpha_i : i = 1, \ldots, k\} + \chi_C(x), \quad \forall x \in X \tag{1.1}$$

where $a_i \in X, \alpha_i \in \mathbb{R}$ for $i = 1, \ldots, k$ and C is a nonempty polyhedral set. Notation χ_C stands for *indicator function* of C and is defined by $\chi_C(x) = 0$ if $x \in C$, and $+\infty$ otherwise. It is clear that $\operatorname{dom} f = C$.

Proposition 1.3 *Let f be a polyhedral convex function, and $x \in \operatorname{dom} f$. Then we have*

1. *f is subdifferentiable at x, and $\partial f(x)$ is a polyhedral convex set. In particular, if f is defined by (1.1) with $C = X$ then*

$$\partial f(x) = \operatorname{co}\{a_i : i \in I(x)\}$$

   where $I(x) = \{i \in \{1, \ldots, k\} : \langle a_i, x\rangle - \alpha_i = f(x)\}$.

2. *The conjugate f^* is a polyhedral convex function. Moreover, if $C = X$ then*

$$\operatorname{dom} f^* = \operatorname{co}\{a_i : i = 1, \ldots, k\}$$
$$f^*(y) = \inf\left\{\sum_{i=1}^k \lambda_i \alpha_i \Big| \sum_{i=1}^k \lambda_i a_i = y, \sum_{i=1}^k \lambda_i = 1, \lambda_i \geq 0, \forall i = 1, \ldots, k\right\}$$

   In particular,

$$f^*(a_i) = \alpha_i, \qquad \forall i = 1, \ldots, k$$

Difference of convex (DC) functions

A function f is called DC function on X if it has the form

$$f(x) = g(x) - h(x), \quad x \in X$$

where g and h belong to $\Gamma_0(X)$. One says that $g-h$ is a *DC decomposition* of f and g, h are its *DC components*. If g and h are in addition finite on all of X then one says that $f = g-h$ is finite DC function on X. The set of DC functions (resp. finite DC functions) on X is denoted by $\mathcal{DC}(X)$ (resp. $\mathcal{DC}_f(X)$).

Remark 1.1 *Give a DC function f having a DC decomposition $f = g - h$. Then for every $\theta \in \Gamma_0(X)$ finite on the whole X, $f = (g+\theta)-(h+\theta)$ is another DC decomposition of f. Thus, a DC function f has finitely many DC decompositions.*

1.2 DC optimization

General DC program

In the sequel, we use the convention $+\infty - (+\infty) = +\infty$.

For $g, h \in \Gamma_0(X)$, a general *DC program* is that of the form

$$(P) \qquad \alpha = \inf\{f(x) = g(x) - h(x) : x \in X\}$$

and its dual counterpart

$$(D) \qquad \alpha^* = \inf\{h^*(y) - g^*(y) : y \in X\}$$

There is a perfect symmetry between primal and dual programs (P) and (D): the dual program to (D) is exactly (P), moreover, $\alpha = \alpha^*$.

Remark 1.2 *Let C be a nonempty closed convex set. Then, the constrained problem*

$$\inf\{f(x) = g(x) - h(x) : x \in C\}$$

can be transformed into an unconstrained DC program by using the indicator function χ_C, i.e.,

$$\inf\{f(x) = \phi(x) - h(x) : x \in X\}$$

where $\phi := g + \chi_C$ is in $\Gamma_0(X)$.

We will always keep the following assumption that is deduced from the finiteness of α

$$\operatorname{dom} g \subset \operatorname{dom} h \quad \text{and} \quad \operatorname{dom} h^* \subset \operatorname{dom} g^*. \tag{1.2}$$

Polyhedral DC program

In problem (P), if one of the DC components g and h is polyhedral function, we call (P) *polyhedral DC program*. This is an important class of DC optimization. It is often encountered in practice and has worthy properties.

Consider problem (P) where h is a polyhedral convex function given by

$$h(x) = \max\{\langle a_i, x\rangle - \alpha_i : i = 1, \ldots, k\}$$

By Proposition 1.3, the dual problem (D) has the form

$$\begin{aligned} \alpha^* &= \inf\{h^*(y) - g^*(y) : y \in X\} \\ &= \inf\{h^*(y) - g^*(y) : y \in \operatorname{co}\{a_i : i = 1, \ldots, k\}\} \\ &= \inf\{\alpha_i - g^*(a_i) : i = 1, \ldots, k\} \end{aligned}$$

Note that, if g is polyhedral convex and h is not, then by considering the dual problem (D) we have the similar formulation as above since g^* is polyhedral.

Optimality conditions for DC optimization

A point x^* is said to be a *local minimizer* of $g - h$ if $x^* \in \operatorname{dom} g \cap \operatorname{dom} h$ (so, $(g - h)(x^*)$ is finite) and there is a neighborhood U of x^* such that

$$g(x) - h(x) \geq g(x^*) - h(x^*), \qquad \forall x \in U. \tag{1.3}$$

A point x^* is said to be a *critical point* of $g - h$ if it verifies the generalized Kuhn–Tucker condition

$$\partial g(x^*) \cap \partial h(x^*) \neq \emptyset \tag{1.4}$$

Let $\mathcal{P}$ and $\mathcal{D}$ denote the solution sets of problems (P) and (D) respectively, and let

$$\mathcal{P}_\ell = \{x^* \in X : \partial h(x^*) \subset \partial g(x^*)\}, \quad \mathcal{D}_\ell = \{y^* \in X : \partial g^*(y^*) \subset \partial h^*(y^*)\}$$

Below, we present some fundamental results on DC programming (Pham Dinh and Le Thi, 1997).

Theorem 1.1 i) *Transportation of global minimizers:* $\cup\{\partial h(x) : x \in \mathcal{P}\} \subset \mathcal{D} \subset \operatorname{dom} h^*$. *The first inclusion becomes equality if* g^* *is subdifferentiable in* $\mathcal{D}$. *In this case* $\mathcal{D} \subset (\operatorname{dom} \partial g^* \cap \operatorname{dom} \partial h^*)$.

ii) *Necessary local optimality: if* x^* *is a local minimizer of* $g - h$, *then* $x^* \in \mathcal{P}_\ell$.

iii) *Sufficient local optimality: Let* x^* *is a critical point of* $g - h$ *and* $y^* \in \partial g(x^*) \cap \partial h(x^*)$. *Let* U *be a neighborhood of* x^* *such that* $(U \cap \operatorname{dom} g) \subset \operatorname{dom} \partial h$. *If for any* $x \in U \cap \operatorname{dom} g$, *there is* $y \in \partial h(x)$ *such that* $h^*(y) - g^*(y) \geq h^*(y^*) - g^*(y^*)$, *then* x^* *is a local minimizer of* $g - h$. *More precisely,*

$$g(x) - h(x) \geq g(x^*) - h(x^*), \qquad \forall x \in U \cap \operatorname{dom} g$$

iv) *Transportation of local minimizers: Let* $x^* \in \operatorname{dom}\partial h$ *be a local minimizer of* $g-h$. *Let* $y^* \in \partial h(x^*)$ *and a neighborhood* U *of* x^* *such that* $g(x)-h(x) \geq g(x^*)-h(x^*)$, $\forall x \in U \cap \operatorname{dom} g$. *If*

$$y^* \in \operatorname{int}(\operatorname{dom} g^*) \quad \text{and} \quad \partial g^*(y^*) \subset U$$

then y^* *is a local minimizer of* $h^* - g^*$.

Remark 1.3 *a) By the symmetry of the DC duality, these results have their corresponding dual part. For example, if* y *is a local minimizer of* $h^* - g^*$, *then* $y \in \mathcal{D}_\ell$.

b) The properties i), iii) and their dual parts indicate that there is no gap between the problems (P) *and* (D). *They show that globally/locally solving the primal problem* (P) *implies globally/locally solving the dual problem* (D) *and vice–versa. Thus, it is useful if one of them is easier to solve than the other.*

c) The necessary local optimality condition $\partial h^*(x^*) \subset \partial g^*(x^*)$ *is also sufficient for many important classes programs, for example (Le Thi and Pham Dinh, 2005), if* h *is polyhedral convex, or when* f *is locally convex at* x^*, *i.e. there exists a convex neighborhood* U *of* x^* *such that* f *is finite and convex on* U. *We know that a polyhedral convex function is almost everywhere differentiable, that is it is differentiable everywhere except on a set of measure zero. Thus, if* h *is a polyhedral convex function, then a critical point of* $g - h$ *is almost always a local solution to* (P).

d) If f *is actually convex on* X, *we call* (P) *a* "false" DC program. *In addition, if* $\operatorname{ri}(\operatorname{dom} g) \cap \operatorname{ri}(\operatorname{dom} h) \neq \emptyset$ *and* $x^0 \in \operatorname{dom} g$ *such that* g *is continuous at* x^0, *then* $0 \in \partial f(x^0) \Leftrightarrow \partial h(x^0) \subset \partial g(x^0)$ *(Le Thi and Pham Dinh, 2005). Thus, in this case, the local optimality is also sufficient for the global optimality. Consequently, if in addition* h *is differentiable, a critical point is also a global solution.*

1.3 DC Algorithm (DCA)

The DCA consists in the construction of the two sequences $\{x^k\}$ and $\{y^k\}$ (candidates for being primal and dual solutions, respectively) which are easy to calculate and satisfy the following properties:

i) The sequences $(g-h)(x^k)$ and $(h^*-g^*)(y^k)$ are decreasing.

ii) Their corresponding limits x^∞ and y^∞ satisfy the local optimality condition $(x^\infty, y^\infty) \in \mathcal{P}_\ell \times \mathcal{D}_\ell$ or are critical points of $g-h$ and h^*-g^*, respectively.

From a given point $x^0 \in \operatorname{dom} g$, the DCA generates these sequences by the scheme

$$y^k \in \partial h(x^k) = \arg\min\{h^*(y) - \langle y, x^k\rangle : y \in X\} \tag{1.5a}$$

$$x^{k+1} \in \partial g^*(y^k) = \arg\min\{g(x) - \langle x, y^k\rangle : x \in X\}. \tag{1.5b}$$

The interpretation of the above scheme is simple. At iteration k of DCA, we replace the second component h in the primal DC program by its affine minorant

$$h_k(x) = h(x^k) + \langle x - x^k, y^k\rangle, \tag{1.6}$$

where $y^k \in \partial h(x^k)$. Then the original DC program reduces to the *convex program*

$$(P_k) \qquad \alpha_k = \inf\{f_k(x) := g(x) - h_k(x) : x \in X\}$$

that is equivalent to (1.5a). It is easy to see that f_k is a majorant of f at x^k. Similarly, by replacing $g*$ with its affine minorant

$$g_k^*(y) = g^*(y^{k-1}) + \langle y - y^{k-1}, x^k \rangle, \tag{1.7}$$

where $x^k \in \partial g^*(y^{k-1})$, we lead to the convex problem

$$(D_k) \qquad \inf\{h^*(y) - g_k^*(y) : y \in X\}$$

whose solution set is $\partial h(x^k)$.

Convergence properties of DCA

Theorem 1.2 *Suppose that the sequences $\{x^k\}$ and $\{y^k\}$ are generated by DCA. Then we have*

i) The sequences $\{g(x^k) - h(x^k)\}$ and $\{h^*(y^k) - g^*(y^k)\}$ are decreasing and

- *$g(x^{k+1}) - h(x^{k+1}) = g(x^k) - h(x^k)$ iff $y^k \in \partial g(x^k) \cap \partial h(x^k), y^k \in \partial g(x^{k+1}) \cap \partial h(x^{k+1})$ and $[\rho(g) + \rho(h)]\|x^{k+1} - x^k\| = 0$. Moreover if g or h are strictly convex then $x^k = x^{k+1}$.*

  In such a case DCA terminates at the k^{th} iteration (finite convergence of DCA)
- *$h^*(y^{k+1}) - g^*(y^{k+1}) = h^*(y^k) - g^*(y^k)$ iff $x^{k+1} \in \partial g^*(y^k) \cap \partial h^*(y^k), x^{k+1} \in \partial g^*(y^{k+1}) \cap \partial h^*(y^{k+1})$ and $[\rho(g^*) + \rho(h^*)]\|y^{k+1} - y^k\| = 0$. Moreover if g^* or h^* are strictly convex, then $y^{k+1} = y^k$.*

  In such a case DCA terminates at the k^{th} iteration (finite convergence of DCA).

ii) If $\rho(g) + \rho(h) > 0$ (resp. $\rho(g^*) + \rho(h^*) > 0$)) then the series $\{\|x^{k+1} - x^k\|^2$ (resp. $\{\|y^{k+1} - y^k\|^2\}$ converges.

iii) If the optimal value α of problem (P) is finite and the infinite sequences $\{x^k\}$ and $\{y^k\}$ are bounded then every limit point x^* (resp. y^*) of the sequence $\{x^k\}$ (resp. $\{y^k\}$) is a critical point of $g - h$ (resp. $h^* - g^*$).

iv) DCA has a linear convergence for general DC programs.

v) In polyhedral DC programs, the sequences $\{x^k\}$ and $\{y^k\}$ contain finitely many elements and DCA has a finite convergence. Especially, if h is polyhedral convex and h is differentiable at x^*, then x^8 is a local minimizer of (P).

DCA's distinctive feature relies upon the fact that DCA deals with the convex DC components g and h but not with the DC function f itself. DCA is one of the rare algorithms

for nonconvex nonsmooth programming. Moreover, a DC function f *has infinitely many DC decompositions which have crucial implications for the qualities* (convergence speed, robustness, efficiency, globality of computed solutions,...) of DCA. For a given DC program, the choice of *optimal* DC decompositions is still open. Of course, this depends strongly on the very specific structure of the problem being considered. In order to tackle the large-scale setting, one tries in practice to choose g and h such that sequences $\{x^k\}$ and $\{y^k\}$ can be easily calculated, *i.e.,* either they are in an explicit form or their computations are inexpensive. Very often in practice, the solution of (D_k) to compute the sequence $\{y^k\}$ is explicit because the calculation of a subgradient of h is explicitly obtained by using the usual rules for calculating subdifferential of convex functions. But the solution of the convex program (P_k), if not explicit, should be achieved by efficient algorithms well-adapted to its special structure, in order to handle the large-scale setting.

1.4 Special DCA and proximal operator

The general scheme of DCA requires to solve a sequence of the subproblems of the form (1.5b) that might be not easy to solve. The design of an efficient DCA for a concrete problem should be based is a special structure. How to exploit the nice effect of DC decomposition is a crucial question to be studied for each DC program.

Before closing this chapter let us discuss the a special DC decomposition that can be very efficiently in many practical problems. Consider the convex constrained DC program of the form

$$\min\{f(x) = g_1(x) + g_2(x) - h(x) : x \in C\}, \tag{1.8}$$

where $C \subset \mathbb{R}^n$ is a convex set and g_1, g_2, h are convex functions.

We assume that there exists a nonnegative number ρ such that the function $\frac{\rho}{2}\|x\|^2 - g_2(x)$ is convex. In many practical problems that ρ exists and can be computed according to the properties of the function g_2. For instance, when g_2 is twice continuously differentiable. We now write the problem (1.8) in the form of DC program with the following decomposition:

$$\bar{g} = \chi_C(x) + g_1(x) + \frac{\rho}{2}\|x\|^2,$$
$$\bar{h} = \frac{\rho}{2}\|x\|^2 - g_2(x) + h(x).$$

The DCA applied to the problem (1.8) with above decomposition can be described as follows:
SDCA (Special DCA): Let $x^0 \in \mathbb{R}^n$ and set $l \leftarrow 0$.
Repeat
1. Calculate $y^l \in \partial\left(\frac{\rho}{2}\|.\|^2 - g_2(.) + h(.)\right)(x^l)$
2. Calculate x^{l+1} by solving the convex problem

$$\min\{\chi_C(x) + g_1(x) + \frac{\rho}{2}\|x\|^2 - \langle x, y^l\rangle : x \in \mathbb{R}^n\} \tag{1.9}$$

i.e., $x^{l+1} = \text{prox}_\rho^{\chi_C + g_1}(y^l/\rho)$.
3. $l \leftarrow l + 1$
Until convergence of $\{x^l\}$.
Here, prox_ρ^φ stands for the proximal operator associated to φ defined by

$$\text{prox}_\rho^\varphi(t) = \arg\min_x \{\varphi(x) + \frac{\rho}{2}\|x - t\|^2\}.$$

Remark 1.4 *i) When $g_1 \equiv 0$, the proximal operator $prox_\rho^{\chi_C}(y^l/\rho)$ in the step 3 reduces to the orthogonal projection $P_C(y^l/\rho)$ of y^l/ρ on C. For certain cases of C, for example, box and ball, Algorithm SDCA is greatly less expensive than other algorithms, because the orthogonal projection on C in these cases is given in explicit form (see Le Thi et al. (2014b); Pham Dinh and Le Thi (1998); Le Thi and Pham Dinh (1998)).*

ii) In many application problems with sparsity-inducing norms, we will have $g_1(x) = \lambda\|x\|_1$. Hence, the proximal operator $prox_\rho^{\chi_C + \lambda\|.\|_1}(y^l/\rho)$ can be computed by an inexpensive algorithm. Especially, if $C = \mathbb{R}^n$, x^{l+1} has a closed form:

$$x^{l+1} = \mathcal{S}\left(y^l/\rho, \lambda/\rho\right), \tag{1.10}$$

where $\mathcal{S}$ is a soft-thresholding operator.

iii) From Prop. 1 in Le Thi et al. (2014b), we obtain that $\frac{\rho}{2}\|x\|^2 - g_2(x)$ is convex if $\rho \geq \max\{0, \lambda_{\max}(H_{g_2}(x))\}$ for all $x \in C$, where $\lambda_{\max}(H_{g_2}(x))$ denotes the largest eigenvalue of the Hessian matrix $H_{g_2}(x)$ of g_2 at x.

iv) In practice, when g_2 is differentiable and the computation of its gradient is not difficult, and the proximal operator $prox_\rho^{\chi_C + g_1}(y^l/\rho)$ can be inexpensively determined, the use of SDCA is highly recommended.

Part I

Variable Selection and Classification

Chapter 2

Sparse Fisher Linear Discriminant Analysis

Abstract: *We consider the supervised pattern classification in the high dimensional setting, in which the number of features is much larger than the number of observations. We present a novel approach to the sparse Fisher linear discriminant problem using the ℓ_0-norm. The resulting optimization problem is nonconvex, discontinuous and very hard to solve. We overcome the discontinuity by using appropriate approximations to the ℓ_0-norm such that the resulting problems can be formulated as DC (Difference of Convex functions) programs to which DC programming and DC Algorithms (DCA) are investigated. The experimental results on both simulated and real datasets demonstrate the efficiency of the proposed algorithms compared to some state-of-the-art methods.*

2.1 Introduction

The problem of classifying observations into Q classes ($Q \geq 2$) has drawn considerable attention from researchers in machine learning, as it has been applied in many fields such as information retrieval or face recognition. In the literature, several notions have been used to formalize this classification problem. Let X be an $n \times p$ data matrix with observations x_i $(i = 1, ..., n)$ on the rows and features on the columns. Denote n_i the number of observations in the class C_i. We assume that the features have been standardized to have mean 0 and variance 1. To obtain an optimal classification rule, we need to know the class posterior probabilities $\Pr(k|x)$. We suppose that $f_k(x)$ is the class-conditional density in the class k, and let π_k be the prior probability of the class k, with

1. This chapter is published under the titles:
[1] Hoai An Le Thi and Duy Nhat Phan. DC Programming and DCA for Sparse Fisher Linear Discriminant Analysis. *Neural Computing and Applications* (2016), doi: 10.1007/s00521-016-2216-9.
[2] Duy Nhat Phan, Manh Cuong Nguyen and Hoai An Le Thi. A DC Programming Approach for Sparse Linear Discriminant Analysis. *Advanced Computational Methods for Knowledge Engineering, Advances in Intelligent Systems and Computing*, Volume 282, pp. 65-74, Springer (2014).

$\sum_{k=1}^{Q} \pi_k = 1$. Linear discriminant analysis (LDA) performs classification by assuming that the data within each class are normal distributed.

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_w|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_k)^T\Sigma_w^{-1}(x-\mu_k)\right\}, \tag{2.1}$$

where μ_k is the mean vector of the k-th class and Σ_w is the common within-class covariance matrix. In practice μ_k, π_k and Σ_w are unknown, but they can be estimated from the training data by $\mu_k = 1/n_k \sum_{x_i \in C_k} x_i, \pi_k = n_k/n$ and

$$\Sigma_w = \frac{1}{n}\sum_{k=1}^{Q}\sum_{x_i \in C_k}(x_i-\mu_k)(x_i-\mu_k)^T. \tag{2.2}$$

Various approaches have been presented for solving the classification problem. Among these methods, Linear Discriminant Analysis (LDA) which was first introduced in Fisher (1936) is regarded as one of the most popular methods because of its advantage of dimension reduction. The primary purpose of LDA is to find a linear transformation that best discriminates between classes. The classification is then performed in the transformed space using some distance metrics. There are three different approaches to tackle LDA, which are based on solving the normal model, the Fisher's discriminant problem and the optimal scoring problem, respectively (Hastie et al., 1995; Mardia et al., 1979; Hastie et al., 2009).

For the first approach, the LDA classification rule is obtained by using Bayes's rule to estimate the most likely class for a new observation, i.e., the predicted class for a new observation x is

$$\arg\max_k Pr(k|x), \tag{2.3}$$

where a simple application of the Bayes's theorem gives us

$$Pr(k|x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{Q} f_l(x)\pi_l}. \tag{2.4}$$

In comparing two classes k and l, it is sufficient to look at the log-ratio:

$$\begin{aligned}\ln\frac{Pr(k|x)}{Pr(l|x)} &= \ln\frac{f_k(x)}{f_l(x)} + \ln\frac{\pi_k}{\pi_l} \\ &= x^T\Sigma_w^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_w^{-1}\mu_k + \ln\pi_k - \left(x^T\Sigma_w^{-1}\mu_l - \frac{1}{2}\mu_l^T\Sigma_w^{-1}\mu_l + \ln\pi_l\right).\end{aligned} \tag{2.5}$$

Therefore, the decision rule (2.3) is equivalent to

$$\arg\max_k \left\{x^T\Sigma_w^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_w^{-1}\mu_k + \ln n_k\right\}. \tag{2.6}$$

With the second approach, the main purpose of this method is seeking a low dimensional projection of the observations such that the between-class variance is large relative to

the within-class variance, i.e. we seek *discriminant vectors* $w_1, ..., w_{Q-1}$ that successively minimize

$$\min_{w_k \in \mathbb{R}^p} \left\{ -w_k^T \Sigma_b w_k : w_k^T \Sigma_w w_k = 1; \quad w_k^T \Sigma_w w_l = 0, l = 1, .., k-1 \right\}, \tag{2.7}$$

where $\Sigma_b = \frac{1}{n} \sum_{k=1}^{Q} n_k \mu_k \mu_k^T$ is a standard estimate of the between-class covariance matrix. The problem (2.7) is a generalized eigen problem which has at most $Q-1$ non trivial solutions, since Σ_b has rank at most $Q-1$, and hence at most $Q-1$ discriminant vectors. A classification rule is obtained by computing $Xw_1, ..., Xw_{Q-1}$ and assigning each observation to its nearest centroid in this transformation space. We can use only the first $K \leq Q-1$ discriminant vectors in order to perform reduced rank classification.

For the last one, the rationality of this method derives from the fact that LDA can also be re-formulated as a regression problem via optimal scoring. This approach was discussed in detail by Hastie et al. (1995). Let $Y \in \mathbb{R}^{n \times Q}$ with $Y_{ik} = 1$ if $x_i \in C_k$ and 0 otherwise. To find K discriminant vectors $w_1, ..., w_K$, the optimal scoring criterion successively solves the problem

$$\begin{aligned} \min_{w_k, \theta_k} \quad & \left\{ ||Y\theta_k - Xw_k||_2^2 \right\} \\ \text{subject to} \quad & \frac{1}{n} \theta_k^T Y^T Y \theta_k = 1; \quad \theta_k^T Y^T Y \theta_l = 0, l = 1, ..., k-1, \end{aligned} \tag{2.8}$$

where θ_k is a Q-vector of scores.

In this chapter, we are interested in the Fisher's discriminant problem. To solve the generalized eigen problem (2.7), it typically requires the within-class covariance matrix Σ_w to be nonsingular. However, this requirement is difficult to satisfy when the dimensionality is high, because the matrix Σ_w is likely to be singular. In fact, in many applications such as information retrieval, face recognition and microarray analysis, we often encounter problems having a small number of observations but a very large number of features. In such cases, the classical LDA includes two great challenges. The first challenge is the singularity of the within-class covariance matrix of the features and the second one is the difficulty in interpreting the classification rule.

Numerous methods have been proposed to overcome the first challenge (see e.g. (Hastie et al., 1995; Bickel and Levina, 2004; Krzanowski et al., 1995; Xu et al., 2009)). These approaches use positive definite estimates of the within-class covariance matrix to deal with the singularity issue. Thus the problem (2.7) becomes

$$\min_{w_k \in \mathbb{R}^p} \left\{ -w_k^T \Sigma_b w_k : w_k^T \tilde{\Sigma}_w w_k = 1; \quad w_k^T \tilde{\Sigma}_w w_l = 0, l = 1, ..., k-1 \right\}, \tag{2.9}$$

where $\tilde{\Sigma}_w$ is a positive definite estimate for the within-class covariance matrix. The problem (2.9) is equivalent to the following problem (see (Witten and Tibshirani, 2011))

$$\min_{w_k \in \mathbb{R}^p} \left\{ -w_k^T \Sigma_b^k w_k : w_k^T \tilde{\Sigma}_w w_k \leq 1 \right\}, \tag{2.10}$$

where

$$\Sigma_b^k = \frac{1}{n} X^T Y (Y^T Y)^{-\frac{1}{2}} P_k^{\perp} (Y^T Y)^{-\frac{1}{2}} Y^T X. \tag{2.11}$$

Here $Y \in \mathbb{R}^{n \times Q}$ with $Y_{ij} = 1$ if $i \in C_j$ and 0 otherwise, $P_1^{\perp} = I_p$ (identity matrix), and $P_k^{\perp} (k > 1)$ is an orthogonal projection matrix into the orthogonal space of the space generated by $\{(Y^T Y)^{-\frac{1}{2}} Y^T X w_l : l = 1, ..., k-1\}$.

For the second challenge, the most suitable approach is feature selection. A sparse classifier leads to easier model interpretation and may reduce overfitting of the training data. In the literature, there exist a number of works to extend LDA to the high-dimensional setting in such a way that the resulting classifier involves a sparse linear combination of the features. We here refer to the notable approaches. One of them is based on soft-thresholding in order to obtain a sparse classifier (see e.g. Tibshirani et al. (2002, 2003); Guo et al. (2007); Shao et al. (2011)). Several authors use the ℓ_1-norm to deal with sparsity. More precisely, the ℓ_1-regularization is added to the objective function of the optimal scoring problem (2.8) (see e.g. (Grosenick et al., 2008; Leng, 2008; Clemmensen et al., 2011)), and/or the Fisher's discriminant problem (2.10) (Trendafilov and Jolliffe, 2007; Wu et al., 2009; Witten and Tibshirani, 2011). In particular, Witten and Tibshirani (2011) proposed a biconvex formulation closely related to the sparse principal components analysis proposal of Witten et al. (2009). Mai et al. (2012); Cai and Liu (2011) developed direct approaches for sparse discriminant analysis. Mai and Zou (2013) showed the connection between and the equivalence of three sparse discriminant analysis methods proposed in Wu et al. (2009), Clemmensen et al. (2011) and Mai et al. (2012).

A natural way to deal with feature selection in machine learning is using the ℓ_0-norm in the regularization term for the problem (2.10). As a result, we propose the sparse Fisher linear discriminant (SFLD) problem

$$\min_{w_k \in \mathbb{R}^p} \left\{ -w_k^T \Sigma_b^k w_k + \lambda_k ||w_k||_0 : w_k^T \tilde{\Sigma}_w w_k \leq 1 \right\}, \tag{2.12}$$

where $||w_k||_0$ denotes the ℓ_0-norm of w_k, i.e. the number of non-zero elements of vector w_k, and λ_k is a nonnegative tuning parameter.

Solving (2.12) is a formidable challenge since it is nonconvex, discontinuous and NP-hard. Optimization methods involving the ℓ_0-norm can be divided into three categories according to the way to treat the ℓ_0-norm: convex approximation, nonconvex approximation and nonconvex exact reformulation. We refer to Le Thi et al. (2015) for an excellent review on exact/approximation approaches to deal with the ℓ_0-norm. The best known and widely used convex approximation of ℓ_0-norm is ℓ_1-norm called Lasso (Tibshirani, 1996). For the problem (2.12), Witten and Tibshirani (2011) replaced the ℓ_0-norm with the ℓ_1-norm and applied the minorization-maximization (MM) approach for solving the resulting problem. This algorithm is in fact a version of difference of convex functions algorithm (DCA). DC approximation approaches for the ℓ_0-norm have been studied extensively on both theoretical and practical aspects for the problem of feature selection in SVM (see e.g. (Le Thi et al., 2008, 2009, 2015; Collobert et al., 2006; Neumann et al., 2005; Ong and Le Thi, 2013b)), and linear regression (see e.g (Chen et al., 2010; Gasso

et al., 2009)). These works add the ℓ_0-norm to a convex function and they only have a difficulty in treating the ℓ_0-norm. In this chapter, solving (2.12) includes double difficulties. The first is how to treat the ℓ_0-norm and the second is caused by the non-convexity of the original Fisher's discriminant problem. To tackle the ℓ_0-norm we investigate DC approximation approaches. Among several existing sparse inducing functions we are using the piecewise linear function (called Capped-ℓ_1) and the piecewise exponential concave function introduced respectively in Peleg and Meir (2008) and Bradley and Mangasarian (1998). This choice is motivated by the fact that the Capped-ℓ_1 has been proved theoretically in Le Thi et al. (2015) to be the tightest approximation while the piecewise exponential function has been showed to be efficient via the numerical results in numerous works (see e.g. (Bradley and Mangasarian, 1998; Le Thi et al., 2008, 2015, 2014c; Ong and Le Thi, 2013b)). The resulting problems are formulated as DC programs, and then DCA are applied. We propose two DCA schemes for two different formulations of a common model to both the approximation functions.

The rest of chapter is organized as follows. In Section 2.2, we illustrate how to apply DCA to solve the problem (2.12). The numerical experiments are reported in Section 2.3. Finally, the conclusions are given in Section 2.4.

We are now going to present solution methods based on DC programming and DCA for solving the SFLD problem (2.12).

2.2 Solution methods via DC programming and DCA

2.2.1 DC approximations of ℓ_0-norm

In this chapter, we consider two DC approximations of the ℓ_0-norm. For an $\alpha > 0$, let $\eta_{\alpha,1}$ and $\eta_{\alpha,2}$ be the functions given by

$$\eta_{\alpha,1}(x) = 1 - e^{-\alpha|x|}, \quad \forall x \in \mathbb{R}, \tag{2.13}$$

and

$$\eta_{\alpha,2}(x) = \min\{1, \alpha|x|\}, \quad \forall x \in \mathbb{R}, \tag{2.14}$$

respectively. Their graphs are illustrated in Figure 2.1.

The first DC approximation of the ℓ_0-norm called the piecewise exponential concave function (Bradley and Mangasarian, 1998) is defined by

$$||w_k||_0 \approx \sum_{i=1}^{p} \eta_{\alpha,1}(w_{ki}). \tag{2.15}$$

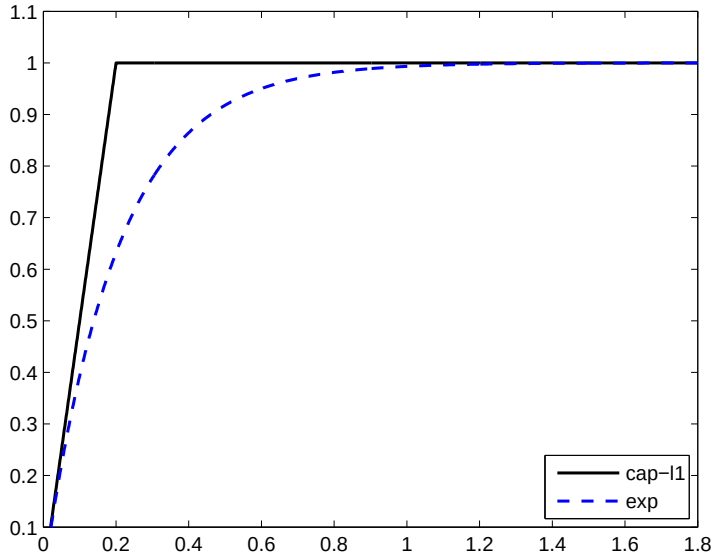

Figure 2.1: Graphs of approximation functions: Capped-ℓ_1 and exponential function

The second DC approximation of the ℓ_0-norm is the piecewise linear approximation proposed in Peleg and Meir (2008). It is described as follows.

$$||w_k||_0 \approx \sum_{i=1}^{p} \eta_{\alpha,2}(w_{ki}). \tag{2.16}$$

For simplification, we use the common notation η_α to design both $\eta_{\alpha,1}$ and $\eta_{\alpha,2}$. Then, the approximations (2.15) and (2.16) are rewritten as follows.

$$||w_k||_0 \approx \sum_{i=1}^{p} \eta_\alpha(w_{ki}). \tag{2.17}$$

Using the approximation (2.17), we can reformulate the SFLD problem (2.12) in the form

$$\min \left\{ F(w_k) = -w_k^T \Sigma_b^k w_k + \lambda_k \sum_{i=1}^{p} \eta_\alpha(w_{ki}) : w_k \in \Omega \right\}, \tag{2.18}$$

where $\Omega = \left\{ w_k \in \mathbb{R}^p : w_k^T \tilde{\Sigma}_w w_k \leq 1 \right\}$.

Note that $\eta_\alpha(w_{ki}) = \eta_\alpha(|w_{ki}|)\ \forall w_{ki} \in \mathbb{R}$ and η_α is increasing concave over $[0, +\infty]$, hence we get another equivalent form of (2.18)

$$\min \left\{ \bar{F}(w_k, z) = -w_k^T \Sigma_b^k w_k + \lambda_k \sum_{i=1}^{p} \eta_\alpha(z_i) : (w_k, z) \in \Omega_1 \right\}, \tag{2.19}$$

where $\Omega_1 = \{(w_k, z) : w_k \in \Omega, |w_{ki}| \leq z_i \quad \forall i = 1, ..., p\}$.

2.2.2 DCA for solving (2.18)

The approximation η_α can be presented as a DC function (Le Thi et al., 2008; Ong and Le Thi, 2013b)

$$\eta_\alpha(x) = g(x) - h(x), \tag{2.20}$$

where $g(x) = \alpha|x|$, $h(x) = -1 + \alpha|x| + e^{-\alpha|x|}$ if $\eta_\alpha = \eta_{\alpha,1}$, and $h(x) = -1 + \max\{1, \alpha|x|\}$ if $\eta_\alpha = \eta_{\alpha,2}$. Note that the first DC decomposition $g(x)$ is the same for the both $\eta_{\alpha,1}$ and $\eta_{\alpha,2}$.

Then, the problem (2.18) can be rewritten as follows.

$$\min\left\{F(w_k) = G_1(w_k) - H_1(w_k) : w_k \in \mathbb{R}^p\right\}. \tag{2.21}$$

where

$$G_1(w_k) := \chi_\Omega(w_k) + \lambda_k \sum_{i=1}^{p} g(w_{ki}), \tag{2.22}$$

and

$$H_1(w_k) := w_k^T \Sigma_b^k w_k + \lambda_k \sum_{i=1}^{p} h(w_{ki}) \tag{2.23}$$

are clearly convex functions. According to the generic DCA scheme, at each iteration l, we have to compute a subgradient $v^l \in \partial H_1(w_k^l)$ and then solve the convex program of the form (P_l), namely

$$\min\{G_1(w_k) - \langle v^l, w_k\rangle : w_k \in \mathbb{R}^p\}, \tag{2.24}$$

For $k = 1, , ..., K$, DCA for solving (2.21) can be described as below.

DCA1 (DCA for solving (2.21))

Initialization: Let τ be a tolerance sufficient small, set $l = 0$ and choose $w_k^0 \in \Omega$.
repeat

1. Compute $v^l \in \partial H_1(w_k^l)$
2. Solve the following convex problem to obtain w_k^{l+1}

$$\min\left\{\lambda_k \alpha ||w_k||_1 - \langle v^l, w_k\rangle : w_k^T \tilde{\Sigma}_w w_k \le 1\right\}. \tag{2.25}$$

3. $l \leftarrow l + 1$.

until $||w_k^l - w_k^{l-1}||_2 \le \tau\left(||w_k^{l-1}||_2 + 1\right)$ or $|F(w_k^l) - F(w_k^{l-1})| \le \tau\left(|F(w_k^{l-1})| + 1\right)$.

The implementation of DCA1 requires the computation of $v^l \in \partial H_1(w_k^l)$ in step 1, which depends on η_α. More precisely, when $\eta_\alpha = \eta_{\alpha,1}$, h is differentiable, so is h. Thus v^l is computed by

$$v_i^l = \begin{cases} 2\langle \Sigma_{bi}^k, w_k^l\rangle + \lambda_k \alpha(1 - e^{-\alpha w_{ki}^l}) & \text{if} \quad w_{ki}^l \ge 0 \\ 2\langle \Sigma_{bi}^k, w_k^l\rangle - \lambda_k \alpha(1 - e^{\alpha w_{ki}^l}) & \text{if} \quad w_{ki}^l < 0 \end{cases} \quad i = 1, ..., p, \tag{2.26}$$

where Σ_{bi}^k stands for the ith row of Σ_b^k. When $\eta_\alpha = \eta_{\alpha,2}$, v^l is calculated as follows.

$$v_i^l = \begin{cases} 2\langle \Sigma_{bi}^k, w_k^l \rangle + \text{sgn}(w_{ki})\lambda_k \alpha & \text{if} \quad \alpha |w_{ki}^l| \geq 1 \\ 2\langle \Sigma_{bi}^k, w_k^l \rangle & \text{otherwise} \end{cases} i = 1, ..., p, \tag{2.27}$$

where $\text{sgn}(w_{ki})$ is the sign of w_{ki}, i.e., $\text{sgn}(w_{ki}) = -1$ if $w_{ki} < 0$, 1 if $w_{ki} > 0$, and 0 otherwise.

Remark 2.1 *1. For solving the convex problem (2.25), we first solve the following convex problem.*

$$\hat{d} = \arg\min_{d \in \mathbb{R}^p} \left\{ d^T \tilde{\Sigma}_w d + \lambda_k \alpha ||d||_1 - \langle v^l, d \rangle \right\}. \tag{2.28}$$

Then, the solution to (2.25) is $\hat{w}_k = 0$ if $\hat{d} = 0$ and $\widehat{w}_k = \hat{d}/\sqrt{\hat{d}^T \tilde{\Sigma}_w \hat{d}}$ otherwise (see (Witten and Tibshirani, 2011)).

2. The problem (2.28) can be solved by using DCA (see (Le Thi, 2000)) with special DC components of its objective function as follows:

$$G(d) = \frac{\mu}{2}\|d\|_2^2 + \lambda_k \alpha \|d\|_1 - \langle v^l, d \rangle,$$
$$H(d) = \frac{\mu}{2}\|d\|_2^2 - d^T \tilde{\Sigma}_w d,$$

are convex functions when μ is larger than or equal to the largest eigenvalue of $\tilde{\Sigma}_w$. At each iteration of this DCA, we can explicitly compute the solution to its convex subproblem by a soft-thresholding. We also note that the coordinate descent approach (see (Friedman et al., 2007)) and the alternating direction method of multipliers (see (Boyd et al., 2011)) can deal with this problem.

3. In fact, when the diagonal estimate of the within-class covarince matrix is used, the solution to the problem (2.28) is explicitly computed.

4. We observe that the problem (2.25) has ℓ_1-perturbed form, hence this DCA scheme can be seen as ℓ_1-perturbed algorithm (see (Le Thi et al., 2015)).

Theorem 2.1 *(Convergence properties of DCA1)*
(i) DCA1 generates the sequence $\{w_k^l\}$ in Ω such that $\{F(w_k^l)\}$ is decreasing.
(ii) Every limit point w_k^* of the sequence $\{w_k^l\}$ is a critical point of the problem (2.21)

Proof : Observing that Ω is a compact set, (i) and (ii) are direct consequences of convergence properties of general DC programs. □

2.2.3 DCA for solving (2.19)

We are going to apply DCA to the problem (2.19). This problem can be written as a DC program

$$\min \left\{ \bar{F}(w_k, z) = G_2(w_k, z) - H_2(w_k, z) \right\}, \tag{2.29}$$

where

$$G_2(w_k, z) := \chi_{\Omega_1}(w_k, z), \tag{2.30}$$

and

$$H_2(w_k, z) := w_k^T \Sigma_b^k w_k + \lambda_k \sum_{i=1}^{p} (-\eta_\alpha)(z_i). \tag{2.31}$$

DCA applied to (2.29) consist of first calculating a subgradient

$$(v^l, \bar{z}^l) \in \partial H_2(w_k^l, z^l), \tag{2.32}$$

and then solving the following convex program at the each iteration.

$$(w_k^{l+1}, z^{l+1}) \in \arg\min \left\{ G_2(w_k, z) - \langle v^l, w_k \rangle - \langle \bar{z}^l, z \rangle \right\} \tag{2.33}$$

When $\eta_\alpha = \eta_{\alpha,1}$, $\bar{z}^l$ is calculated by

$$\bar{z}_i^l = -\lambda_k \alpha \exp(-\alpha z_i^l) \quad \forall i = 1, ..., p, \tag{2.34}$$

and when $\eta_\alpha = \eta_{\alpha,2}$, we have

$$\bar{z}_i^l = \begin{cases} -\lambda_k \alpha & \text{if} \quad z_i^l \leq 1/\alpha, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i = 1, ..., p. \tag{2.35}$$

We deduce from (2.34) and (2.35) that $\bar{z}_i^l \leq 0 \ \forall i = 1, ..., p$. Thus, the problem (2.33) is equivalent to

$$\begin{cases} w_k^{l+1} \in \arg\min\limits_{w_k \in \Omega} \left\{ -\langle v^l, w_k \rangle - \sum\limits_{i=1}^{p} \bar{z}_i^l |w_{ki}| \right\} \\ z_i^{l+1} = |w_{ki}^{l+1}| \quad \forall i. \end{cases} \tag{2.36}$$

Hence DCA applied on (2.29) is given by the algorithm below.

DCA2 (DCA for solving (2.29))

Initialization: Let τ be a tolerance sufficient small, set $l = 0$ and choose $(w_k^0, z^0) \in \Omega_1$.

repeat

1. Compute $v^l = 2\Sigma_b^k w_k^l$ and $\bar{z}_i^l \in \lambda_k \partial(-\eta_\alpha)(z_i^l)$ according to (2.34) and (2.35).
2. Solve the following convex problem to obtain w_k^{l+1}

$$\min \left\{ -\langle v^l, w_k \rangle - \sum_{i=1}^{p} \bar{z}_i^l |w_{ki}| : w_k^T \tilde{\Sigma}_w w_k \leq 1 \right\} \tag{2.37}$$

3. Compute $z_i^{l+1} = |w_{ki}^{l+1}| \quad \forall i = 1, ..., p.$
4. $l \leftarrow l + 1.$

until $||w_k^l - w_k^{l-1}||_2 \leq \tau\left(||w_k^{l-1}||_2 + 1\right)$ or $|\bar{F}(w_k^l, z^l) - \bar{F}(w_k^{l-1}, z^{l-1})| \leq \tau\left(|\bar{F}(w_k^{l-1}, z^{l-1})| + 1\right)$.

Remark 2.2 *The problem (2.37) has ℓ_1-reweighted form, hence this DCA scheme can be regarded as ℓ_1-reweighted algorithm.*

Theorem 2.2 *(Convergence properties of DCA2)*
(i) DCA2 generates the sequence $\{(w_k^l, z^l)\}$ in Ω_1 such that $\{\bar{F}(w_k^l, z^l)\}$ is decreasing.
(ii) Every limit point (w_k^*, z^*) of the sequence $\{(w_k^l, z^l)\}$ is a critical point of the problem (2.29)

Proof : The Theorem is direct consequences of convergence properties of general DC programs. □

2.3 Numerical experiments

We use the SFLD problem for supervised classification problems in high dimension. The SFLD problem transforms the set of labeled data points in the original space into a labeled set in a lower-dimensional space and selects relevant features. The classification rule is obtained by computing $Xw_1, ..., Xw_s$ and assigning each observation to its nearest centroid in this transformation space, i.e. the predicted class for a test observation x is

$$\arg\min_k ||x^T W - \mu_k^T W||_2^2 - 2\ln(n_k), \tag{2.38}$$

where the linear transformation $W = [w_1, ..., w_K]$ is computed by DCA1 or DCA2 and the second term is an adjustment term for unequal class sizes.

2.3.1 Comparative algorithms

We denote by DCA1-PiE and DCA1-Capped-ℓ_1 the DCA1 with $\eta_\alpha = \eta_{\alpha,1}$ and $\eta_\alpha = \eta_{\alpha,2}$, respectively. The DCA2 with $\eta_\alpha = \eta_{\alpha,1}$ and $\eta_\alpha = \eta_{\alpha,2}$ are denoted by DCA2-PiE and DCA2-Capped-ℓ_1, respectively. We will compare our proposed Algorithms with the methods proposed in Witten and Tibshirani (2011) (PLDA) and Guo et al. (2007) (RDA). We also compare with the method proposed in Mai et al. (2012) (DSDA) for the binary classification problems.

2.3.1.1 Penalized linear discriminant analysis (PLDA)

PLDA used the ℓ_1-norm instead of the ℓ_0-norm in the problem (2.12), that is

$$\max_{w_k \in \mathbb{R}^p} \{w_k^T \Sigma_b^k w_k - \lambda_k ||w_k||_1 : w_k^T \tilde{\Sigma}_w w_k \leq 1\}, \tag{2.39}$$

The problem (2.39) is nonconvex. Witten and Tibshirani (2011) used the MM approach for finding a local of this problem. In the experiments, the authors used the diagonal estimate $\text{diag}(\sigma_1^2, ..., \sigma_p^2)$ for $\tilde{\Sigma}_w$, where σ_i is the within-class standard deviation for feature i. The R package **penalizedLDA** is available from CRAN [2].

2.3.1.2 Shrunken centroids regularized discriminant analysis (RDA)

RDA is based on the same underlying model as LDA (see (Guo et al., 2007)) and it regularizes the within-class covariance matrix used by LDA

$$\tilde{\Sigma}_w = \alpha \Sigma_w + (1 - \alpha) I_p, \tag{2.40}$$

where $0 \leq \alpha \leq 1$. In order to perform feature selection, one can perform soft-thresholding of the quantity $\tilde{\Sigma}_w^{-1} \mu_k$. That is, we compute

$$\text{sgn}(\tilde{\Sigma}_w^{-1} \mu_k)(|\tilde{\Sigma}_w^{-1} \mu_k| - \delta)_+, \tag{2.41}$$

where δ is a nonnegative tuning parameter. The R package **rda** is available from CRAN.

2.3.1.3 Direct sparse discriminant analysis (DSDA)

Mai et al. (2012) developed DSDA for the binary classification setting. Let y_i be equal to n_1/n (resp. n_2/n) if the observation x_i belongs to class 1 (resp. class 2). The solution to DSDA is defined by

$$(\hat{\beta}, \hat{\beta}_0) = \arg \min_{(\beta, \beta_0)} \sum_{i=1}^{n} (y_i - \beta_0 - X\beta)^2 + \lambda \|\beta\|_1, \tag{2.42}$$

where λ is a tuning parameter. Then the classification rule is to assign x to class 2 if

$$[x - (\hat{\mu}_1 + \hat{\mu}_2)]^T \hat{\beta} + \hat{\beta}^T \hat{\Sigma} \hat{\beta} \left[(\hat{\mu}_2 - \hat{\mu}_1)^T \hat{\beta} \right]^{-1} \ln(n_2/n_1) > 0,$$

where $\hat{\mu}_1$ and $\hat{\mu}_2$ are the sample mean vectors of classes 1 and 2, respectively, and $\hat{\Sigma}$ is the pooled sample covariance matrix.

2. http://cran.r-project.org/

2.3.2 Datasets

We evaluate the performance of DCA1-PiE, DCA1-Capped-ℓ_1, DCA2-PiE and DCA2-Capped-ℓ_1 on three synthetic datasets and a collection of real world datasets. The description of three synthetic datasets are summarized in Table 2.1 and they are generated as follows (see (Witten and Tibshirani, 2011)):

For the first setup S1, we generate a four classes classification problem. Each class is assumed to have a multivariate normal distribution $N(\mu_k, I)$, $k = 1, 2, 3, 4$ with dimension of $p = 500$. The first 25 components of μ_1 are 0.7, $\mu_{2j} = 0.7$ if $26 \leq j \leq 50$, $\mu_{3j} = 0.7$ if $51 \leq j \leq 75$, $\mu_{4j} = 0.7$ if $71 \leq j \leq 100$ and 0 otherwise. For each class, we generate 100 training samples, 100 tuning samples and 500 test samples.

The second simulation setup S2 includes two classes of multivariate normal distributions $N(\mu_1, \Sigma)$ and $N(\mu_2, \Sigma)$, each of dimension $p = 500$. The components of μ_1 are assumed to be 0 and for μ_2, $\mu_{2j} = 0.6$ if $j \leq 200$ and 0 otherwise. The covariance matrix Σ is the block diagonal matrix with five blocks of dimension 100×100 whose element (j, j') is $0.6^{|j-j'|}$. For each class, 50 training samples, 50 tuning samples and 500 test samples are generated.

For the last setup S3, we generate a four-class classification problem as follows: $i \in C_k$ then $X_{ij} \sim N((k-1)/3, 1)$ if $j \leq 100$, $k = 1, 2, 3, 4$ 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 100 training samples, 100 tuning samples and 1000 test samples are generated with equal probabilities for each class.

Table 2.1: Synthetic datasets used in experiments.

Datasets	#Features	#Train	#Test	#Classes
Simulation 1 (S1)	500	400	2000	4
Simulation 2 (S2)	500	100	1000	2
Simulation 3 (S3)	500	100	1000	4

The real world datasets consist of two real datasets from UCI Machine Learning Repository and NIPS 2003 Feature Selection Challenge (Internet Advertisement and Gisette), seven real microarray gene expression datasets, and one dataset for handwritten character recognition (MNIST). All the datasets are pre-processed by normalizing each dimension of the data to zero mean and unit variance. The detailed information of these datasets is summarized in Table 2.2.

The Colon Tumor dataset of intensities of 2000 genes in 22 normal and 40 tumor colon tissues. It is published in Alon et al. (1999) and available at http://genomics-pubs.princeton.edu/oncology/.

SRBCT (Khan et al., 2001) is the dataset of small, round blue cell tumors of childhood and can be downloaded at http://research.nhgri.nih.gov/microarray/Supplement/.

The training and test set consist of 83 samples spanning four classes.

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

Lung Cancer dataset, there are 181 tissue samples (31 MPM and 150 ADCA). Each sample is described by 12533 genes. It is used in Gordon et al. (2002) and can be downloaded at http://datam.i2r.a-star.edu.sg/datasets/krbd/.

Leukemia microarray dataset is published in Yeoh et al. (2002) and available at http://datam.i2r.a-star.edu.sg/datasets/krbd/. The data consisted of 12558 gene expression measurements for 248 samples belong to six cancer classes.

Nakayama data consisting of 86 samples from 5 types of soft tissue tumors, each with 22283 gene expression measurements (Nakayama et al., 2007). This data is available at Gene Expression Omnibus.

Sun data consisting of 180 samples and 54613 expression measurements (see (Sun et al., 2006)). The samples are belong to four classes. It is available at Gene Expression Omnibus.

MNIST dataset is available at http://yann.lecun.com/exdb/mnist/. The training and test sets consist of 60000 and 10000 images of size 28×28 pixels, respectively.

Table 2.2: Real datasets used in experiments.

Datasets	#Features	#Samples	#Classes
Internet Advertisement (ADV)	1558	3279	2
Colon Tumor (COL)	2000	62	2
SRBCT (SRB)	2308	83	4
Pencillium (PEN)	3754	36	3
Gisette (GIS)	5000	7000	2
Lung Cancer (LUN)	12533	181	2
Leukemia (LEU)	12558	248	6
Nakayama (NAK)	22283	86	5
Sun (SUN)	54613	180	4
MNIST (MNI)	784	60000/10000	10

2.3.3 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.

We use the same diagonal estimate $\tilde{\Sigma}_w$ as PLDA. Its advantage for the data in which the number of features is much larger than the number of observations has been indicated in

Bickel and Levina (2004). Hence the solution $\hat{d}$ to the problem (2.28) is explicitly defined by

$$\hat{d}_i = \frac{1}{2\sigma_i^2} S(v_i^l, \lambda_k \alpha) \quad \forall i = 1, ..., d, \tag{2.43}$$

where S is the soft-thresholding operator defined as

$$S(z,t) = \begin{cases} z - t & \text{if } z > 0 \text{ and } t < |z|, \\ z + t & \text{if } z < 0 \text{ and } t < |z|, \\ 0 & \text{if } t \geq |z|. \end{cases} \tag{2.44}$$

The value of λ_k is taken as $\lambda_k = \lambda.\lambda_1^k$, where λ_1^k is the largest eigenvalue of Σ_b^k. This can avoid penalizing each discriminant vector more than the previous discriminant vectors, since the objective value of problem (2.12) without the zero-norm is equal to the largest eigenvalue of Σ_b^k when the diagonal estimate of the within-class covariance matrix is used. From which it follows that an eigenvector of the matrix Σ_b^k corresponding to the largest eigenvalue is also a good starting point w_k^0 of DCA, which seems to be natural (see (Witten and Tibshirani, 2011)).

The values of parameters λ and K (the number of used discriminant vectors) are chosen through a sixfold cross-validation procedure on training set from a set of candidates. The approximation parameter in (2.17) is fixed $\alpha = 5$ as suggested in Bradley and Mangasarian (1998). Concerning the parameter α, from the theoretical point of view, the larger α is, the better approximation of the ℓ_0-norm is. However, when we tried with larger α (up to 100), the result is not improved. The stop tolerance of DCA is $\tau = 10^{-6}$. We select relevant features as follows: feature i is deleted if $|w_{ki}| < 10^{-6}$ for all $k = 1, ..., K$.

2.3.4 Numerical results on synthetic data

In this experiment, we generate training, tuning, and test sets in the same manner as described in Sect. 2.3.2. The tuning sets are used to choose the parameters λ and the number of discriminant vectors used K, 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. DSDA is only tested on the 2-class synthetic dataset S2.

The experimental results on synthetic data are given in Table 2.3. In this table, the average of percentage of selected features and its standard deviation (FS), the average of percentage of accuracy of classifiers and its standard deviation (ACC), the average of CPU time in second and its standard deviation (CPU) over 10 trials, as well as the number of discriminant vector used (K) are reported.

We observe from Table 2.3, in terms of feature selection, the DCA based algorithms give better results than PLDA, RDA and DSDA. DCA1-Capped-ℓ_1 gives the best results on 2/3 synthetic datasets. On average of 3 synthetic datasets, DCA1-PiE, DCA1-Capped-ℓ_1, DCA2-PiE and DCA2-Capped-ℓ_1 respectively select 41.6%, 41.17%, 35.86% and 42.48% of features while PLDA and RDA respectively select 46.93% and 49.45% of features.

Table 2.3: Comparative results of DCA1-PiE, DCA1-Capped-ℓ_1, DCA2-PiE, DCA2-Capped-ℓ_1, PLDA, RDA and DSDA on synthetic data. Bold fonts indicate the best results in each row.

		DCA1-PiE	DCA1-Capped-ℓ_1	DCA2-PiE	DCA2-Capped-ℓ_1	PLDA	RDA	DSDA
ACC	S1	**96.89 ± 0.035**	96.66 ± 0.38	96.52 ± 0.28	96.45 ± 0.4	96.8 ± 0.39	96.34 ± 0.64	-
	S2	**98.1 ± 0.38**	97.9 ± 0.44	**98.1 ±0.38**	98.03 ± 0.23	98 ± 0.43	94.8 ± 1.01	91.62 ± 1.12
	S3	**88.03 ± 1.61**	87.51 ± 1.73	88.03 ±0.1.16	87.72 ± 1.12	87.31 ± 1.31	68.81 ± 1.6	-
Average		**94.34**	94.02	94.21	94.07	94.05	86.66	-
FS	S1	43.76 ± 12.44	52.48 ± 9.99	26.54 ± 0.77	47.64 ± 1.53	54.94 ± 7.03	**22.8 ± 0.95**	-
	S2	55.3 ± 2.68	**45.64 ± 1.78**	55.3 ± 2.68	51.76 ± 2.79	51.76 ± 2.79	79.68 ± 1.39	59.45 ± 10.29
	S3	25.74 ± 1.38	**25.4 ± 1.38**	25.74 ± 1.38	28.04 ± 1.62	34.1 ± 1.38	45.86 ± 1.79	-
Average		41.6	41.17	**35.86**	42.48	46.93	49.45	-
CPU	S1	0.022 ± 0.009	0.02 ± 0.008	0.06 ± 0.09	0.032 ± 0.01	**0.017 ± 0.01**	0.328 ± 0.015	-
	S2	**0.001 ± 0.003**	0.002 ± 0.006	0.004 ±0.006	0.006 ± 0.008	0.002 ± 0.006	0.03 ± 0.004	0.1 ± 0.02
	S3	0.004 ± 0.006	**0**	0.004 ± 0.006	0.01 ± 0.008	0.002 ± 0.006	0.034 ± 0.006	-
Average		0.009	**0.007**	0.023	0.016	**0.007**	1.131	-
K	S1	3	3	3	3	3	-	-
	S2	1	1	1	1	1	-	1
	S3	1	1	1	1	1	-	-

The DCA based algorithms not only provide a good performance in terms of feature selection, but also give a high accuracy of classifiers. DCA1-PiE gives the best accuracy of classifiers on all the three synthetic datasets. On average of 3 synthetic datasets, DCA1-PiE, DCA1-Capped-ℓ_1, DCA2-PiE, DCA2-Capped-ℓ_1, PLDA and RDA respectively give the accuracy of classifiers 94.34%, 94.02%, 94.21%, 94.07%, 94.05% and 86.66%. On the second synthetic dataset, the DCA based algorithms are also better than DSDA in terms of accuracy of classifiers as well as feature selection.

In Table 2.3, we also see that DCA1-Capped-ℓ_1 and PLDA are fastest on all the three synthetic datasets.

2.3.5 Numerical results on real datasets

For the experiments on the first nine real datasets, we use the cross-validation scheme to validate the performance of various classifiers. Each real dataset is split into a training set containing 2/3 of the samples and a test set containing 1/3 of the samples. This process is repeated 10 times, each with a random choice of training set and test set. The parameter λ and the number of discriminant vectors K which is used are chosen via 6-fold cross-validation.

The computational results given by DCA1-PiE, DCA1-Capped-ℓ_1, DCA2-PiE, DCA2-Capped-ℓ_1, PLDA, RDA and DSDA are reported in Tables 2.4-2.5. We are interested in the efficiency (the sparsity and the accuracy of classifiers) as well as the rapidity of the algorithms. We notice that DSDA is only tested on the 2-class datasets.

We observe from computational results that:

Sparsity: On all the datasets, the classifiers obtained by the DCA based algorithms are sparser than those obtained by PLDA, RDA and DSDA. DCA1-Capped-ℓ_1 is the best on 5 out of 9 datasets and DCA1-PiE is the best on 2 out of 9 datasets. The DCA based algorithms select from $0,09\%$ to 69.74% of features while PLDA and RDA choose from 15.36% to 100% of features. Overall, DCA1-Capped-ℓ_1 realizes a better trade-off between accuracy and sparsity than other algorithms. It suppresses considerably the number of features (up to 99.9%) while the correctness of classification is quite good (from 72.33% to 100%).

Accuracy of classifiers: In terms of the accuracy of classifiers, the DCA based algorithms attain better than PLDA, RDA and DSDA on 6/9 datasets (the gains vary from 0.05 to 11.45%). More specifically, DCA1-Capped-ℓ_1 is the best on 3/9 datasets, especially for the very large SUN data (54613 features), this approach only selects 9.02% of features but achieves the best accuracy of classifiers (72.33%). RDA is slightly better than the DCA based algorithms on the dataset NAK (the gains is 0.62%). However, RDA selects much more features than these approaches (the gain is 42.38% of features). On the two datasets ADV and GIS which the number of observations are larger than the number of features, RDA and DSDA are better than the DCA based algorithms. This can be explained

Table 2.4: Comparative results of DCA1-PiE, DCA1-Capped-ℓ_1, DCA2-PiE, DCA2-Capped-ℓ_1, PLDA, RDA and DSDA in terms of the average of percentage of accuracy of classifiers and its standard deviation (upper row), and the average of 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.

Datasets	DCA1-PiE	DCA1-Capped-ℓ_1	DCA2-PiE	DCA2-Capped-ℓ_1	PLDA	RDA	DSDA
ADV	94.17 ± 0.33	94.25± 0.29	94.17 ± 0.33	94.16 ± 0.28	94.23 ± 0.31	**96.47 ± 0.26**	96.1 ± 0.65
	25.84 ± 0.36	30.02 ± 1.13	25.84 ± 0.36	**22.52 ± 0.21**	33.18 ± 0.78	27.14 ± 0.92	26.56 ± 1.62
COL	81.1 ± 4.89	81.57 ± 5.08	81.09 ± 4.89	**82.52 ± 6.47**	78.78 ± 6.54	71.07 ± 7.5	81.48 ± 6.36
	0.12 ± 0.09	**0.1 ± 0.04**	0.16 ± 0.08	0.29 ± 0.1	21.7 ± 7.99	96.59 ± 0.45	1.11 ± 0.14
SRB	**99.62 ± 1.15**	99.27 ± 1.46	99.61 ± 1.15	98.92 ± 1.64	97.44 ± 3.66	99.29 ± 1.4	-
	9.84 ± 17.39	23.93 ± 29.23	14.28 ± 21.23	17.82 ± 24.44	57.4 ± 6.37	15.36 ± 0.4	-
PEN	**100 ± 0**	**100± 0**	**100± 0**	96.66 ± 4.08	**100± 0**	96.66 ± 2.52	-
	0.09 ± 0.01	61.59 ± 2.12	**0.09 ± 0.01**	5.18 ± 4.25	63.47 ± 3.02	94.28 ± 0.03	-
GIS	86.92 ± 0.47	87.45 ± 0.63	86.54 ± 0.66	86.88 ± 0.66	86.88 ± 0.63	84.52 ± 0.66	**94.07 ± 0.33**
	28.51 ± 4.28	50.56 ± 5.18	**22.6 ± 0.17**	28.26 ± 0.21	28.27 ± 0.21	98.67 ± 0.16	35.31 ± 0.4
LUN	**99.34 ± 0.81**	**99.34 ±0.81**	**99.34 ± 0.81**	99.17 ± 0.83	99.17 ± 0.83	98.18 ± 1.38	94.86 ± 2.5
	20.01 ± 1.21	**12.31 ± 10.1**	20.81 ± 0.77	17.94 ± 5.98	20.86 ± 2.83	98.67 ± 0.12	20.82 ± 3.41
LEU	96.87 ± 1.16	96.99 ± 1.32	96.87 ± 1.21	**97.11 ± 1.33**	96.86 ± 1.33	97.06 ± 1.59	-
	29.36 ± 12.51	**28.21 ± 13.97**	35.55 ± 0.24	30.56 ± 1.57	35.21 ± 0.23	83.6 ± 0.06	-
NAK	87. 82 ± 6.18	89.23 ± 4.23	86.76 ± 7.08	87.1 ± 7.13	87.1 ± 7.13	**89.85 ± 4.61**	-
	69.05 ± 8.82	**57.61 ± 29.22**	69.33 ± 5.58	69.74 ± 6.13	68.4 ± 7.12	99.99 ± 0.006	-
SUN	70.83 ± 4.07	**72.33 ± 3.75**	69.66 ± 4.72	69.66 ± 4.72	69.66 ± 4.72	67.67 ± 4.53	-
	16.24 ± 14.33	**9.02 ± 13.83**	33.93 ± 1.21	33.17 ± 4.62	35.34 ± 0.81	100 ± 0	-

Table 2.5: Comparative results of DCA1-PiE, DCA1-Capped-ℓ_1, DCA2-PiE, DCA2-Capped-ℓ_1, PLDA, RDA and DSDA in terms of the average of CPU time in second and its standard deviation (upper row) over 10 training/test set splits, the number of discriminant vectors used K (the data is projected onto a K-dimensional space) (lower row). Bold fonts indicate the best results in each row.

Datasets	DCA1-PiE	DCA1-Capped-ℓ_1	DCA2-PiE	DCA2-Capped-ℓ_1	PLDA	RDA	DSDA
ADV	**0.001 ± 0.003**	0.003 ± 0.006	0.007 ± 0.007	0.008 ± 0.008	0.004 ± 0.007	58.24 ± 1.05	7.19 ± 0.32
	1	1	1	1	1	-	1
COL	0.024 ± 0.019	**0.001 ± 0.003**	0.033 ± 0.017	0.004 ± 0.006	0.01 ± 0.014	0.067 ± 0.009	0.25 ± 0.009
	1	1	1	1	1	-	1
SRB	0.141 ± 0.026	**0.048 ± 0.037**	0.169 ± 0.046	0.061 ± 0.04	0.079 ± 0.01	0.082 ± 0.008	-
	3	3	3	3	3	-	-
PEN	0.046 ± 0.018	0.056 ± 0.023	0.051 ± 0.021	**0.03 ± 0.01**	0.033 ± 0.01	0.109 ± 0.009	-
	2	2	2	2	2	-	-
GIS	**0.004 ± 0.006**	0.008 ± 0.012	0.014 ± 0.008	0.013 ± 0.016	0.044 ± 0.083	829.918± 63.815	158.64 ± 9.37
	1	1	1	1	1	-	1
LUN	0.12 ± 0.045	**0.036 ± 0.016**	0.095 ± 0.032	0.101 ± 0.052	0.063 ± 0.012	1.052 ± 0.036	19.02 ± 0.39
	1	1	1	1	1	-	1
LEU	0.39 ± 0.306	0.27 ± 0.089	0.248 ± 0.016	0.38 ± 0.021	**0.21 ± 0.013**	1.71± 0.085	-
	5	5	5	5	5	-	-
NAK	0.971 ± 0.19	**0.664 ± 0.246**	0.907 ± 0.19	0.716 ± 0.19	0.749 ± 0.128	1.185 ± 0.972	-
	4	4	4	4	4	-	-
SUN	1.527 ± 1.007	**0.738 ± 0.593**	1.173 ± 0.23	0.961 ± 0.197	0.996 ± 0.156	6.819 ± 3.843	-
	2	2	2	2	2	-	-

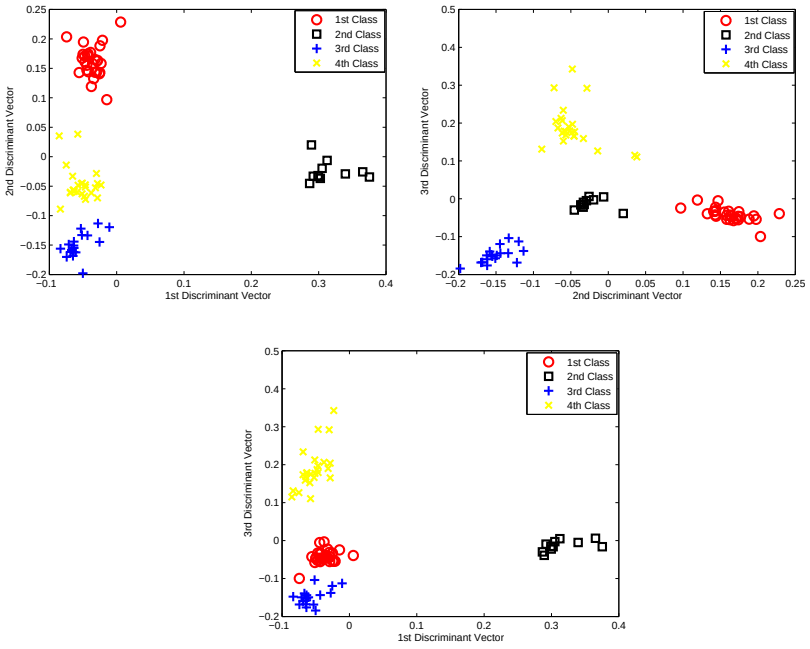

Figure 2.2: The SRBCT dataset was projected onto the first three sparse discriminant vectors. The samples in each class are shown by using a distinct symbol.

that these approaches use the sample covariance matrix which can be appropriate for these datasets. To sum up, when the number of features are much larger than the number of observations, our proposed methods are especially more efficient than other algorithms in classification of high dimensional data. Thus, the proposed methods are highly recommended for this type of data.

Training time and discriminant vectors: Training time and the numbers of discriminant vectors which are used are reported in Table 2.5. The DCA based algorithms and PLDA run very fast and they are comparable. RDA and DSDA are much slower than the DCA based algorithms (the ratios of gains are from 1.7 to 39660 times). The discriminant vectors can be used to visualize the datasets such as in Figures 2.2-2.3.

The computational results of the MNIST dataset are reported on Table 2.6. Notably in this dataset, RDA is not able to perform since the amount of RAM is insufficient. From Table 2.6, we observe that the DCA based algorithms outperform PLDA in terms of sparsity as well as accuracy of classifiers. As for the training time, all five algorithms run very fast (less than 0.2 s).

2.4 Conclusion

We have proposed efficient approaches for solving the Sparse Fisher Linear Discriminant problem using the ℓ_0-regularization. Among several sparse inducing functions of the

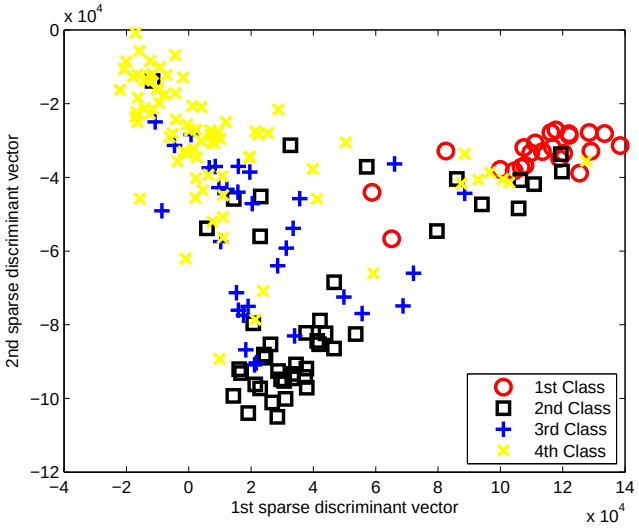

Figure 2.3: The Sun dataset was projected onto the first two sparse discriminant vectors. The samples in each class are shown by using a distinct symbol.

Table 2.6: Comparative results of MNIST dataset in terms of the number of selected features, the percentage of accuracy of classifiers on the test set, and training time in second. Bold fonts indicate the best results in each column.

Method	Selected features (%)	Accuracy of classifiers (%)	CPU in second
DCA1-PiE	**65.17**	**88.79**	0.09
DCA1-Capped-ℓ_1	65.94	88.77	0.15
DCA2-PiE	**65.17**	**88.79**	0.11
DCA2-Capped-ℓ_1	65.94	88.77	0.13
PLDA	80.09	81.11	**0.05**
RDA	-	-	-

ℓ_0-norm, we have used two appropriate approximations functions and reformulated the resulting problems as DC programs. Overall, four DCA based algorithms have been developed in order to exploit the nice effect of DC decompositions/DC formulations on the one part, and the efficiency of the approximate functions on the other part. The robustness and the effectiveness of our DCA based algorithms have been demonstrated through the computational results on both the simulated and real datasets. Their efficiency has been compared with three standard algorithms which use the ℓ_1-regularization (note again that this is the first work investigating ℓ_0-regularization for the SFLD problem).

The research developed in this chapter permitted us to highlight the following comments/recommendations:

1. *ℓ_1-regularization versus ℓ_0-regularization:* similar to several works using ℓ_0-regularization in learning with sparsity; once again, our work proved that ℓ_0-regularization produces much better sparsity than ℓ_1-regularization.
2. *ℓ_1-LDA (or more generally convex-LDA) versus ℓ_0-LDA:* unlike the convex regularization approach for several learning problems (for instance, the feature selection in SVM/linear regression, etc.) where convex regularizations result in convex optimization problems (which are so far easy to solve) and the convex-LDA problem is still nonconvex and then difficult. Hence, the quality of solutions (the sparsity and the accuracy of classifiers) depends on the efficiency of algorithms being investigated for these nonconvex programs. The same argument (both convex-LDA and ℓ_0-LDA are nonconvex) does not necessarily imply that ℓ_0-LDA algorithms are more time-consuming than the convex-LDA algorithms. Therefore, we recommend to use ℓ_0-LDA in high-dimensional data classification, not only when sparsity is significantly desired, but also when high accuracy is requested.
3. *PLDA, RDA and DSDA versus DCA:* in our numerical experiments the DCA based algorithms are most of the time better than PLDA, RDA and DSDA. This superiority comes mainly from the arguments mentioned in (1) and (2) above. In another hand, the estimation of the within-class covariance matrix used in each method ($\tilde{\Sigma}_w$) influences also on its efficiency. A more detailed comparative analysis can be summarized as follows:
   - *PLDA versus DCA:* these methods aim to solve the Fisher's discriminant problem but with two different regularizations to deal with sparsity: ℓ_1(PLDA) and ℓ_0(DCA). They use the same diagonal estimate matrix of the within-class covariance matrix $\tilde{\Sigma}_w$ in the model, and their iterations are based on the same idea (the MM method used in PLDA is a special version of the general DCA scheme). Thanks to ℓ_0-regularization, DCA always produce better sparsity than PLDA. By the nice effect of DC decompositions/DC formulations of the resulting nonconvex approximate problems, DCA give higher classification accuracy than PLDA. The training time of PLDA and DCA are comparable, but thanks again to the effect of DC decompositions, it is quite possible that DCA are faster than PLDA. Note also that, when the number of features is much larger than the number of observations, the diagonal estimate matrix $\tilde{\Sigma}_w$ used in our DCA (and in PDLA) is good, and therefore, DCA produce high classification accuracy. Moreover, with this diagonal estimate matrix $\tilde{\Sigma}_w$, DCA are explicitly computed at each iteration,

and by the way, they are very fast. Hence, the use of DCA for this type of data is highly recommended.

- *RDA versus DCA:* RDA does not solve the Fisher's discriminant problem but uses the classification rule (2.6) which requires the matrix inversion (compute directly $\tilde{\Sigma}_w^{-1}$). Such a procedure is very time-consuming when the size of $\tilde{\Sigma}_w$ (which is the number of features) is large. That is why DCA are always much faster than RDA, and the gain is more important in high-dimensional datasets. Note, however, that when the features are independent or when the number of sample is greater than the number of features, the estimate diagonal matrix used in DCA may not be appropriate, and then, it could happen that RDA produces better sparsity (e.g., the synthetic dataset S1 having independent features) or RDA gives better classification accuracy (e.g., the datasets ADV and GIS where the number of samples is greater than the number of features). For such types of data, the regularized matrix $\tilde{\Sigma}_w$ defined in (2.40) by RDA is more appropriate. For other types of data, DCA considerably outperform RDA on both sparsity and classification accuracy.
- *DSDA versus DCA:* for binary classification DSDA requires solving the problem (2.42), while DCA's iterations are explicitly defined in a very simple formulation. That is why DCA are faster than DSDA. By the same argument mentioned above concerning the efficiency of the estimate diagonal matrix used in DCA for datasets having the number of samples greater than the number of features, it could happen that DSDA gives higher classification accuracy than DCA (e.g., for the datasets ADV and GIS). As for sparsity, DCA are always better than DSDA which uses the ℓ_1-regularization.

4. *About the four versions of DCA for the SFLD problems:* DCA1-Capped-ℓ_1 is the best, most of the time, on both sparsity and classification accuracy, and it always realizes the best trade-off between sparsity and classification accuracy. This confirms once again the results developed in Le Thi et al. (2015): The Capped-ℓ_1 is the best nonconvex (DC) approximations, and the ℓ_1-perturbed algorithm (DCA1) is more efficient than the ℓ_1-reweighted algorithm (DCA2).

As a part of future work, we plan to study more extensive applications of the SFLD problem. We believe that the success of using DC approximation functions for the ℓ_0-norm motivates and opens up a new avenue for the sparse Linear Discriminant Analysis (LDA) problem. In particular, we intend to apply different DC approximation functions as well as further explore other models for the sparse LDA problem.

Chapter 3

Sparse Optimal Scoring Problem

Abstract: *Linear discriminant analysis (LDA) is a standard tool for classification and dimension reduction in many applications. However, the problem of high dimension is still a great challenge for the classical LDA. In this chapter we consider the supervised pattern classification in the high dimensional setting, in which the number of features is much larger than the number of observations and present a novel approach to the sparse optimal scoring problem using the zero-norm. The difficulty in treating the zero-norm is overcome by using appropriate continuous approximations such that the resulting problems are solved by alternating schemes based on DC (Difference of Convex functions) programming and DCA (DC Algorithms). The experimental results on both simulated and real datasets show the efficiency of the proposed algorithms compared to some state-of-the-art methods.*

3.1 Introduction

Among several classification methods in the literature the LDA based approach is regarded as one of the most popular and is known to be efficient for problems having a small number of observations but a very large number of features. There are three different approaches to tackle LDA, which are based on solving the normal model, the Fisher's discriminant problem and the optimal scoring problem, respectively (see Chapter 2 for more details). In this chapter we do not directly consider the Fisher's discriminant problem. Instead, we are interested in the optimal scoring interpretation of LDA and develop a optimal scoring based approach, named the sparse optimal scoring problem.

Let X be an $n \times p$ data matrix with observations x_i $(i = 1, ..., n)$ on the rows and features

1. This chapter is published under the titles:
[1] Hoai An Le Thi and Duy Nhat Phan. DC Programming and DCA for Sparse Optimal Scoring Problem. *Neurocomputing* 186: 170-181 (2016).
[2] Hoai An Le Thi and Duy Nhat Phan. A DC Programming Approach for Sparse Optimal Scoring. *Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science*, Volume 9078, pp. 435-446, Springer (2015).

on the columns. n_i is denoted as the number of observations in the cluster C_i, $i = 1, ..., Q$. We assume that the features have been standardized to have mean 0 and variance 1. Let $Y \in \mathbb{R}^{n \times Q}$ with $Y_{ik} = 1$ if $x_i \in C_k$ and 0 otherwise. To find the linear transformation W, the optimal scoring criterion successively solves the problem

$$\begin{aligned} \min_{w_k, \theta_k} \quad & \left\{ ||Y\theta_k - Xw_k||_2^2 \right\} \\ \text{subject to} \quad & \frac{1}{n}\theta_k^T Y^T Y \theta_k = 1; \quad \theta_k^T Y^T Y \theta_l = 0, l = 1, ..., k-1, \end{aligned} \tag{3.1}$$

where θ_k is a Q-vector of *scores*.

LDA often performs quite well in simple and low-dimensional setting and it is known to fail when the number of features p is larger than the number of observations n. However, in many applications such as information retrieval, face recognition and microarray analysis, we often encounter problems having a small number of observations but a very large number of features. In such cases, one difficulty of the classical LDA is interpretation of the classifier, since the classification rule involves a linear combination of all p features. To overcome this, the most suitable approach is feature selection. A sparse classifier leads to easier model interpretation and may reduce overfitting of the training data. In the literature, several authors use the ℓ_1-norm to deal with sparsity. More precisely, the ℓ_1-regularization is added to the objective function of the optimal scoring problem (3.1) (see e.g. (Grosenick et al., 2008; Leng, 2008; Clemmensen et al., 2011)). Clemmensen et al. (2011) replaced the ℓ_0-norm with the ℓ_1-norm and applied an alternating scheme for solving the resulting problem.

The most natural way to deal with feature selection in machine learning is using the ℓ_0-norm in the regularization term. Using $\ell_2 + \ell_0$ regularization for the optimal scoring problem (3.1) leads us to consider the sparse optimal scoring (SOS) problem defined by

$$\begin{aligned} \min_{w_k, \theta_k} \quad & \frac{1}{2n}||Y\theta_k - Xw_k||_2^2 + \lambda \left[\frac{1-\gamma}{2}||w_k||_2^2 + \gamma||w_k||_0 \right] \\ \text{subject to} \quad & \frac{1}{n}\theta_k^T Y^T Y \theta_k = 1; \quad \theta_k^T Y^T Y \theta_l = 0, l = 1, ..., k-1. \end{aligned} \tag{3.2}$$

Here $\gamma \in [0, 1]$ and $\lambda \geq 0$ are tuning parameters, and $||w_k||_0$ denotes the ℓ_0-norm of w_k, i.e. the number of non-zero elements of vector w_k.

In this chapter, solving (3.2) includes double difficulties. The first is how to treat the ℓ_0-norm and the second is caused by the non-convexity of the original optimal scoring problem. To tackle the ℓ_0-norm we investigate DC approximation approaches. As the previous chapter, we use two sparse inducing functions: the piecewise linear function (called Capped-ℓ_1) and the piecewise exponential concave function introduced respectively in Peleg and Meir (2008) and Bradley and Mangasarian (1998). Unfortunately, the resulting optimization problems are still difficult but they enjoy some interesting properties: when w_k is fixed the optimal solution of the problem with respect to the variable θ_k can be computed explicitly, while for each fixed θ_k we are faced on a DC program with respect to the variable w_k. We are then suggested to use alternating schemes based on DCA for solving them.

Our contributions are multiple. Using two DC approximations of the ℓ_0-norm and consider two DC formulations of each resulting approximate SOS problem we propose alternating schemes for solving the four approximate problems. To deal with DC programs w.r.t w_k in each step of the alternating algorithms, we investigate four DCA schemes. We prove that the main algorithms converge to a critical point of the approximate problems. The performance of the proposed algorithms are carefully examined in comparing with seven state of art methods on both simulated datasets and high-dimensional real datasets.

The rest of this chapter is organized as follows. In Section 3.2, we state the approximate problems and present the alternating schemes for solving them as well as the way to compute θ_k in these schemes. In Section 3.3 we show how to apply DCA on the non-convex subproblems to compute w_k in the alternating schemes. Section 3.4 is devoted to the description of the main algorithms and their convergence analysis. The numerical experiments are reported in Section 3.5 and Section 3.6 concludes the chapter.

3.2 Alternating schemes for the approximate sparse optimal scoring problems

3.2.1 Approximate sparse optimal scoring problems

The discontinuity of the ℓ_0-norm is overcome by using two DC approximations. For an $\alpha > 0$, let $\eta_{\alpha,1}$ and $\eta_{\alpha,2}$ be the functions given by

$$\eta_{\alpha,1}(x) = 1 - \exp(-\alpha|x|), \quad \forall x \in \mathbb{R},$$

and

$$\eta_{\alpha,2}(x) = \min\{1, \alpha|x|\}, \quad \forall x \in \mathbb{R},$$

The Capped-ℓ_1 approximation of the ℓ_0-norm is defined by (Peleg and Meir, 2008)

$$||w_k||_0 \approx \sum_{i=1}^{p} \eta_{\alpha,1}(w_{ki}), \tag{3.3}$$

and the piecewise exponential concave approximation (Bradley and Mangasarian, 1998) is

$$||w_k||_0 \approx \sum_{i=1}^{p} \eta_{\alpha,2}(w_{ki}). \tag{3.4}$$

For simplify the presentation, we use the common notation η_α to design both $\eta_{\alpha,1}$ and $\eta_{\alpha,2}$ and consider the resulting approximate problem of the SOS problem (3.2) in the form

$$\min_{(w_k,\theta_k)\in\mathbb{R}^p\times\Omega^k} \left\{ \frac{1}{2n}||Y\theta_k - Xw_k||_2^2 + \lambda \left[\frac{1-\gamma}{2}||w_k||_2^2 + \gamma \sum_{i=1}^{p} \eta_\alpha(w_{ki}) \right] \right\}, \tag{3.5}$$

where $\Omega^k = \{\theta_k \in \mathbb{R}^Q : \theta_k^T D\theta_k = 1; \theta_k^T D\theta_l = 0, l = 1, ..., k-1\}$ and $D = \frac{1}{n}Y^TY$.

Observing that $\eta_\alpha(w_{ki}) = \eta_\alpha(|w_{ki}|)\ \forall w_{ki} \in \mathbb{R}$ and η_α is increasing concave over $[0, +\infty]$, we can deduce another equivalent form of (3.5) (see (Le Thi et al., 2015) for more details)

$$\min_{(w_k,z_k)\in\Lambda^k,\theta_k\in\Omega^k} \left\{ \frac{1}{2n}||Y\theta_k - Xw_k||_2^2 + \lambda \left[\frac{1-\gamma}{2}||w_k||_2^2 + \gamma \sum_{i=1}^{p} \eta_\alpha(z_{ki}) \right] \right\}, \tag{3.6}$$

where $\Lambda^k = \{(w_k, z_k) \in \mathbb{R}^p \times \mathbb{R}_+^p : |w_{ki}| \leq z_{ki} \quad \forall i = 1, ..., p\}$. In the sequel we will investigate solution methods for solving the nonconvex problems (3.5) and (3.6).

For holding w_k (resp. (w_k, z_k)) fixed, we can find an explicit solution θ_k. However, for holding θ_k fixed, the resulting problems are still nonconvex. Hence we will investigate alternating schemes based on DC programming and DCA for solving (3.5) and (3.6).

3.2.2 Alternating schemes for solving the approximate SOS problems

The alternating scheme for solving the problem (3.5) consists of holding θ_k fixed and optimizing with respect to w_k, and then holding w_k fixed and optimizing with respect to θ_k. More precisely:

Starting with $w_k \in \mathbb{R}^p$ and $\theta_k \in \Omega^k$, at each iteration we perform two steps:
1. Fix θ_k and compute w_k by solving

$$\min_{w\in\mathbb{R}^p} \left\{ \frac{1}{2n}||Y\theta_k - Xw||_2^2 + \lambda \left[\frac{1-\gamma}{2}||w||_2^2 + \gamma \sum_{i=1}^{p} \eta_\alpha(w_i) \right] \right\}. \tag{3.7}$$

2. Fix w_k and compute θ_k by solving

$$\min_{\theta\in\Omega^k} \left\{ ||Y\theta - Xw_k||_2^2 \right\}. \tag{3.8}$$

Similarly, the alternating scheme for solving the problem (3.6) differs from the above scheme only on the step 1:

Starting with $(w_k, z_k) \in \Lambda^k$ and $\theta_k \in \Omega^k$, at each iteration we perform two steps:
1. Fix θ_k and compute w_k by solving

$$\min_{(w,z)\in\Lambda^k} \left\{ \frac{1}{2n}||Y\theta_k - Xw||_2^2 + \lambda \left[\frac{1-\gamma}{2}||w||_2^2 + \gamma \sum_{i=1}^{p} \eta_\alpha(z_i) \right] \right\}. \tag{3.9}$$

2. Fix w_k and compute θ_k by solving (3.8).
We will show below how to compute θ_k in the step 2 of these alternating schemes.

3.2.3 Compute θ_k in the alternating schemes

Let Q_{k-1} be the $Q \times (k-1)$ matrix whose columns are the previous $k-1$ solutions $\theta_1, ..., \theta_{k-1}$ consecutively. For solving (3.8), we state the following lemma.

Lemma 3.1 *The problem (3.8) has a unique solution* $\widehat{\theta}_k = s_k/\sqrt{s_k^T D s_k}$, *where* $s_k = (I - Q_{k-1}Q_{k-1}^T D)D^{-1}Y^T X w_k$.

Clemmensen et al. (2011) stated that $\widehat{\theta}_k$ solves the problem (3.8), but they do not give the proof. We will prove that this solution is unique.

Proof : We assume that θ_k is a solution of the problem (3.8). It follows that θ_k satisfies the following KKT conditions:

$$2nD\theta_k - Y^T X w_k + 2\lambda_1 D\theta_k + DQ_{k-1}\lambda_2 = 0, \tag{3.10}$$

$$\theta_k^T D\theta_k = 1, \tag{3.11}$$

$$\theta_k^T DQ_{k-1} = 0, \tag{3.12}$$

where $\lambda_1 \in \mathbb{R}$ and $\lambda_2 \in \mathbb{R}^{k-1}$ are Lagrange multipliers. Multiplying (3.10) by θ_k^T gives

$$n + \lambda_1 = \frac{1}{2}\theta_k^T Y^T X w_k. \tag{3.13}$$

On the other hand, substituting (3.13) into the objective function of the problem (3.8), we have

$$F_{w_k}(\theta_k) = ||Y\theta_k - Xw_k||^2 = n + w_k^T X^T X w_k - 4(n + \lambda_1). \tag{3.14}$$

Thus, we only need to consider $n + \lambda_1 > 0$. From (3.10), solving for θ_k leads to

$$\theta_k = \frac{1}{n+\lambda_1} D^{-1}(Y^T X w_k - \frac{1}{2} DQ_{k-1}\lambda_2). \tag{3.15}$$

The orthogonality constraints give

$$\theta_k^T DQ_{k-1} = 0 \Leftrightarrow w_k^T X^T Y Q_{k-1} - \frac{1}{2}\lambda_2^T Q_{k-1} DQ_{k-1} = 0$$
$$\Rightarrow \lambda_2 = 2Q_{k-1}^T Y^T X w_k.$$

Inserting this expression for λ_2 into equation (3.15) and simplifying gives

$$\theta_k = \frac{1}{n+\lambda_1}(I - Q_{k-1}Q_{k-1}^T D)D^{-1}Y^T X w_k = \frac{1}{n+\lambda_1} s_k, \tag{3.16}$$

where $s_k = (I - Q_{k-1}Q_{k-1}^T D)D^{-1}Y^T X w_k$. Finally, the constraint $\theta_k^T D\theta_k = 1$ gives $n + \lambda_1 = \sqrt{s_k^T D s_k}$, then $\widehat{\theta}_k = s_k/\sqrt{s_k^T D s_k}$ is unique solution of the problem (3.8). □

We are now going to develop DCA based algorithms for solving the subproblems in the step 1 of alternating schemes.

3.3 DCA based algorithms for solving nonconvex subproblems in alternating schemes

3.3.1 DC formulations and DCA based algorithms for nonconvex subproblems (3.7) and (3.9)

The approximation η_α can be expressed as a DC function:

$$\eta_\alpha(x) = g(x) - h(x), \tag{3.17}$$

where $g(x) = \alpha|x|$, $h(x) = -1 + \alpha|x| + \exp(-\alpha|x|)$ if $\eta_\alpha = \eta_{\alpha,1}$, and $h(x) = -1 + \max\{1, \alpha|x|\}$ if $\eta_\alpha = \eta_{\alpha,2}$.

Therefore, the objective function of the problem (3.7) can be rewritten as follows.

$$F_{\theta_k}(w) := G_1(w, \theta_k) - H_1(w), \tag{3.18}$$

where

$$G_1(w, \theta_k) := \frac{1}{2n}||Y\theta_k - Xw||_2^2 + \lambda\left[\frac{1-\gamma}{2}||w||_2^2 + \gamma\sum_{i=1}^{p} g(w_i)\right],$$

$$H_1(w) := \lambda\gamma\sum_{i=1}^{p} h(w_i),$$

are clearly convex functions. Hence a DC formulation of the problem (3.7) takes the form

$$\min_{w\in\mathbb{R}^p}\{G_1(w, \theta_k) - H_1(w)\}. \tag{3.19}$$

According to the generic DCA scheme, DCA applied on (3.19) consists of computing, at each iteration l, a subgradient $v^l \in \partial H_1(w^l)$ and solving the convex program of the form (P_l), namely

$$\min_{w\in\mathbb{R}^p}\{G_1(w, \theta_k) - \langle v^l, w\rangle\}. \tag{3.20}$$

The algorithm is described as follows.

DCA1

Initialization: Let τ be a tolerance sufficient small, set $l = 0$ and choose $w^0 \in \mathbb{R}^p$.
repeat

1. Compute $v^l \in \partial H_1(w^l)$.
2. Solve the following convex problem to obtain w^{l+1}

$$\min_{w\in\mathbb{R}^p}\left\{\frac{1}{2n}||Y\theta_k - Xw||_2^2 + \lambda\left[\frac{1-\gamma}{2}||w||_2^2 + \gamma\alpha||w||_1\right] - \langle v^l, w\rangle\right\}. \tag{3.21}$$

3. $l \leftarrow l + 1$.
until $\|w^l - w^{l-1}\|_2 \leq \tau(\|w^{l-1})\|_2 + 1)$ or $|F_{\theta_k}(w^l) - F_{\theta_k}(w^{l-1})| \leq \tau(|F_{\theta_k}(w^{l-1})| + 1)$.

We see that the problem (3.21) has the ℓ_1-perturbed form.

The implementation of DCA1 requires the computation of $v^l \in \partial H_1(w^l)$ in the step 1, which depends on η_α. More precisely, for $\eta_\alpha = \eta_{\alpha,1}$, v^l is computed by

$$v_i^l = \begin{cases} \lambda\gamma\alpha(1 - \exp(-\alpha w_i^l)) & \text{if} \quad w_i^l \geq 0 \\ -\lambda\gamma\alpha(1 - \exp(\alpha w_i^l)) & \text{if} \quad w_i^l < 0 \end{cases} i = 1, ..., p. \tag{3.22}$$

For $\eta_\alpha = \eta_{\alpha,2}$, v^l is calculated as follows.

$$v_i^l = \begin{cases} \text{sgn}(w_i^l)\lambda\gamma\alpha & \text{if} \quad \alpha|w_i^l| \geq 1 \\ 0 & \text{otherwise} \end{cases} i = 1, ..., p. \tag{3.23}$$

Remark 3.1 *For solving the convex problem (3.21), we use the coordinate descent method (Friedman et al., 2007). This method is known to be one of the most efficient algorithms for solving unconstrained convex problems whose objective function has the form: quadratic + ℓ_1. Its objective function is written as:*

$$f(w) := \frac{1}{2n}\sum_{i=1}^{n}\left[(Y\theta_k)_i - \sum_{m\neq j} X_{im}w_m - X_{ij}w_j\right]^2 + \frac{\lambda(1-\gamma)}{2}\sum_{m\neq j} w_m^2$$
$$+ \frac{\lambda(1-\gamma)}{2}w_j^2 + \lambda\gamma\alpha\sum_{m\neq j}|w_m| + \lambda\gamma\alpha|w_j| - \sum_{m\neq j} v_m^l w_m - v_j^l w_j.$$

Suppose that we have estimates $w_m = \tilde{w}_m$ for $m \neq j$, and we wish to partially optimize with respect to w_j. The coordinate-wise update has the form

$$w_j \longleftarrow \frac{\mathcal{S}\left(\frac{1}{n}\sum_{i=1}^{n} X_{ij}\left[(Y\theta)_i - \sum_{m\neq j} X_{im}\tilde{w}_m\right] + v_j^l, \lambda\gamma\alpha\right)}{1 + \lambda(1-\gamma)}, \tag{3.24}$$

where $\mathcal{S}(z,t)$ is the soft-thresholding operator with value

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

The update (3.24) is repeated for $j = 1, 2, ..., p, 1, 2, ...$ until

$$\|w^k - w^{k-1}\|_2 \leq \epsilon \text{ or } |f(w^k) - f(w^{k-1})| \leq \epsilon,$$

where w^k is the solution obtained at the k-th iteration.

For designing a DC formulation of (3.9) we observe that the function η_α is concave on $[0,+\infty]$ and then $-\eta_\alpha$ is convex on $[0,+\infty]$. Therefore a DC formulation of (3.9) can be

$$\min_{(w,z)\in\mathbb{R}^p\times\mathbb{R}^p} \left\{\bar{F}_{\theta_k}(w,z) := G_2(w,z,\theta_k) - H_2(w,z)\right\}, \tag{3.25}$$

where

$$\begin{aligned} G_2(w,z,\theta_k) &:= \frac{1}{2n}||Y\theta_k - Xw||_2^2 + \frac{\lambda(1-\gamma)}{2}||w||_2^2 + \chi_{\Lambda^k}(w,z), \\ H_2(w,z) &:= \lambda\gamma\sum_{i=1}^{p}(-\eta_\alpha)(z_i), \end{aligned}$$

are clearly convex functions. Here χ_{Λ^k} is the indicator function on Λ^k, that is

$$\chi_{\Lambda^k}(w,z) = \begin{cases} 0 & \text{if} \quad (w,z)\in\Lambda^k, \\ +\infty & \text{otherwise.} \end{cases}$$

Like DCA1, DCA applied on (3.25) consists of computing, at each iteration l, a subgradient $(v^l,\bar{z}^l)\in\partial H_2(w^l,z^l)$, and then solving the following convex program to obtain (w^{l+1},z^{l+1}):

$$\min_{(w,z)\in\mathbb{R}^p\times\mathbb{R}^p} \left\{G_2(w,z,\theta_k) - \langle v^l,w\rangle - \langle\bar{z}^l,z\rangle\right\}. \tag{3.26}$$

When $\eta_\alpha = \eta_{\alpha,1}$, $\bar{z}^l$ is calculated by

$$\bar{z}_i^l = -\exp(-\alpha z_i^l)\lambda\gamma\alpha \quad \forall i = 1,...,p, \tag{3.27}$$

and for $\eta_\alpha = \eta_{\alpha,2}$, we have

$$\bar{z}_i^l = \begin{cases} -\lambda\gamma\alpha & \text{if} \quad z_i^l \le 1/\alpha, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i = 1,...,p. \tag{3.28}$$

We deduce from (3.27) and (3.28) that $\bar{z}_i^l \le 0\ \forall i = 1,...,p$. Thus, the problem (3.26) is equivalent to

$$\begin{cases} w^{l+1} \in \arg\min\limits_{w\in\mathbb{R}^p}\left\{\frac{1}{2n}||Y\theta_k - Xw||_2^2 + \frac{\lambda(1-\gamma)}{2}||w||_2^2 - \sum\limits_{i=1}^{p}\bar{z}_i^l|w_i|\right\} \\ z_i^{l+1} = |w_i^{l+1}| \quad \forall i. \end{cases} \tag{3.29}$$

DCA for solving (3.25) is described as follow.

DCA2

Initialization: Let τ be a tolerance sufficient small, set $l=0$ and choose $w^0\in\mathbb{R}^p$, $z_i^0 = |w_i^0| \quad \forall i = 1,...,p$.

repeat

1. Compute $\bar{z}_i^l \in \lambda\gamma\partial(-\eta_\alpha)(z_i^l)$.

2. Solve the following convex problem to obtain w^{l+1}

$$\min_{w\in\mathbb{R}^p}\left\{\frac{1}{2n}||Y\theta_k - Xw||_2^2 + \frac{\lambda(1-\gamma)}{2}||w||_2^2 + \sum_{i=1}^{p}(-\bar{z}_i^l)|w_i|\right\} \tag{3.30}$$

3. Compute $z_i^{l+1} = |w_i^{l+1}| \quad \forall i = 1, ..., p.$
4. $l \leftarrow l + 1.$

until $\|(w^l, z^l) - (w^{l-1}, z^{l-1})\|_2 \leq \tau(\|(w^{l-1}, z^{l-1})\|_2 + 1)$ or $|\bar{F}_{\theta_k}(w^l, z^l) - \bar{F}_{\theta_k}(w^{l-1}, z^{l-1})| \leq \tau\left(|\bar{F}_{\theta_k}(w^{l-1}, z^{l-1})| + 1\right)$.

In the step 2 of DCA2, we see that the problem (3.30) has the form of a ℓ_1-regularization problem but with different weights on components of $|w_i|$. So DCA2 iteratively solves the weighted-ℓ_1 problem (3.30) with an update of the weights $-\bar{z}_i^l$ at each iteration l.

Theorem 3.1 *(i) DCA1 and DCA2 generate, respectively, the sequences $\{w^l\}_l$ in $\mathbb{R}^p$ and $\{(w^l, z^l)\}_l$ in Λ^k such that $\{F_{\theta_k}(w^l)\}_l$ and $\{\bar{F}_{\theta_k}(w^l, z^l)\}_l$ are decreasing.*

(ii) If the sequence $\{w^l\}_l$ (resp. $\{(w^l, z^l)\}_l$) is bounded, then every limit point w^* (resp. (w^*, z^*)) of the sequence $\{w^l\}_l$ (resp. $\{(w^l, z^l)\}_l$) is a critical point of the problem (3.19) (resp. (3.25)).

(iii) In the case of Capped-ℓ_1 ($\eta_\alpha = \eta_{\alpha,2}$), the sequence $\{w^l\}_l$ and $\{(w^l, z^l)\}_l$ respectively convergence to w^* and (w^*, z^*) after a finite number of iterations. Moreover, the points w^* and (w^*, z^*) are critical points of the problems (3.19) and (3.25), respectively. If, in addition,

$$w_i^* \notin \left\{\frac{1}{\alpha}, -\frac{1}{\alpha}\right\} \ \forall i = 1, ..., p, \tag{3.31}$$

(resp. $z_i^* \neq 1/\alpha \ \forall i = 1, ..., p$) then w^* (resp. (w^*, z^*)) is in fact a local minimizer of (3.19) (resp. (3.25)).

Proof : (i) and (ii) are direct consequences of convergence properties of general DC programs while the first part of (iii) is a convergence property of a DC polyhedral program.

For the second part of (iii), observing that the second DC component of (3.19) (resp. (3.25)) is a polyhedral function. If the condition (3.31) (resp. $z_i^* \neq 1/\alpha \ \forall i = 1, ..., p$) holds, then H_1 (resp. H_2) is differentiate at w^* (resp. (w^*, z^*)). Using the DCA's convergence property (v) in Theorem 1.2, we deduce that w^* (resp. (w^*, z^*)) is a local minimizer of (3.19) (resp. (3.25)) in the case of Capped-ℓ_1. □

We can now describe our main algorithms based on alternating methods and DCA for solving the approximate SOS problems.

3.4 Description of the main algorithms and their convergence properties

Finally, the alternating scheme based on DCA for solving the problem (3.5) can be described as follows.

ADCA1 Alternating scheme based on DCA for the problem (3.5)

---

for $k = 1$ to K, compute k-th discriminant vector w_k as follows:

Initialization: $w_k^0 \in \mathbb{R}^p$, $\theta_k^0 = s_k^0/\sqrt{(s_k^0)^T D s_k^0}$, where $s_k^0 = (I - Q_{k-1}Q_{k-1}^T D)D^{-1}\theta_k^*$ with $\theta_k^* \in \mathbb{R}^Q$, and $l = 0$.

repeat

1. For fixed θ_k^l, compute w_k^{l+1} by DCA1 using w_k^l as initial point.
2. For fixed w_k^{l+1}, compute $s_k^{l+1} = (I - Q_{k-1}Q_{k-1}^T D)D^{-1}Y^T X w_k^{l+1}$ and set $\theta_k^{l+1} = s_k^{l+1}/\sqrt{(s_k^{l+1})^T D s_k^{l+1}}$.
3. $l \leftarrow l + 1$.

until Stopping criterion.

end for

The stopping criterion of ADCA1 is given by

$$\|(w_k^l, \theta_k^l) - (w_k^{l-1}, \theta_k^{l-1})\|_2 \leq \tau(\|(w_k^{l-1}, \theta_k^{l-1})\|_2 + 1),$$

or

$$|F(w_k^l, \theta_k^l) - F(w_k^{l-1}, \theta_k^{l-1})| \leq \tau(|F(w_k^{l-1}, \theta_k^{l-1})| + 1),$$

where $F(w_k, \theta_k)$ is the objective function of the problem (3.5).

For each algorithm, we use two approximations of the ℓ_0-norm ($\eta_{\alpha,1}$ and $\eta_{\alpha,2}$). We denote by ADCA1-Exp (resp. ADCA1-Cap) ADCA1 using $\eta_\alpha = \eta_{\alpha,1}$ (resp. $\eta_\alpha = \eta_{\alpha,2}$).

Furthermore, the alternating scheme using DCA for solving the problem (3.6) is given by the following algorithm.

ADCA2 Alternating scheme based on DCA for the problem (3.6)

---

for $k = 1$ to K, compute k-th discriminant vector w_k as follows:

Initialization: $w_k^0 \in \mathbb{R}^p$, $\theta_k^0 = s_k^0/\sqrt{(s_k^0)^T D s_k^0}$, where $s_k^0 = (I - Q_{k-1}Q_{k-1}^T D)D^{-1}\theta_k^*$ with $\theta_k^* \in \mathbb{R}^Q$, and $l = 0$.

repeat

1. For fixed θ_k^l, compute (w_k^{l+1}, z_k^{l+1}) by DCA2 using w_k^l as initial point.
2. For fixed (w_k^{l+1}, z_k^{l+1}), compute $s_k^{l+1} = (I - Q_{k-1}Q_{k-1}^T D)D^{-1}Y^T X w_k^{l+1}$ and set $\theta_k^{l+1} = s_k^{l+1}/\sqrt{(s_k^{l+1})^T D s_k^{l+1}}$.
3. $l \leftarrow l + 1$.

until Stopping criterion.
end for

The stopping criterion of ADCA2 is given by

$$\|(w_k^l, z_k^l, \theta_k^l) - (w_k^{l-1}, z_k^{l-1}, \theta_k^{l-1})\|_2 \leq \tau(\|(w_k^{l-1}, z_k^{l-1}, \theta_k^{l-1})\|_2 + 1),$$

or

$$|\bar{F}(w_k^l, z_k^l, \theta_k^l) - \bar{F}(w_k^{l-1}, z_k^{l-1}, \theta_k^{l-1})| \leq \tau(|\bar{F}(w_k^{l-1}, z_k^{l-1}, \theta_k^{l-1})| + 1),$$

where $\bar{F}(w_k, z_k, \theta_k)$ is the objective function of the problem (3.6).

ADCA2 using $\eta_\alpha = \eta_{\alpha,1}$ (resp. $\eta_\alpha = \eta_{\alpha,2}$) is denoted by ADCA2-Exp (resp. ADCA2-Cap).

The convergence properties of ADCA1 and ADCA2 are given by Theorem 3.2 below.

Theorem 3.2 *(i) ADCA1 generates the sequences $\{(w_k^l, \theta_k^l)\}_l$ in $\mathbb{R}^p \times \Omega^k$, $k = 1, ..., K$ such that $\{F(w_k^l, \theta_k^l)\}_l$ is decreasing.*
(ii) If the sequence $\{(w_k^l, \theta_k^l)\}_l$ generated by ADCA1 is bounded, then every limit point (w_k^*, θ_k^*) of this sequence is a critical point of the problem (3.5).

Similarly, we have
(iii) ADCA2 generates the sequences $\{(w_k^l, z_k^l, \theta_k^l)\}_l$ in $\Lambda^k \times \Omega^k$, $k = 1, ..., K$ such that $\{\bar{F}(w_k^l, z_k^l, \theta_k^l)\}_l$ is decreasing.
(iv) If the sequence $\{(w_k^l, z_k^l, \theta_k^l)\}_l$ generated by ADCA2 is bounded, then every limit point $(w_k^*, z_k^*, \theta_k^*)$ of this sequence is a critical point of the problem (3.6).

Proof : The properties (i) and (iii) (resp. (ii) and (iv)) are proved analogously. Therefore we give here the proof for (i) and (ii) only.

For (i), we assume that $\{(w_k^l, \theta_k^l)\}_l$, $k = 1, ..., K$ are generated by ADCA1. We have $\{(w_k^l, \theta_k^l)\}_l$ in $\mathbb{R}^p \times \Omega^k$ and

$$F(w_k^{l+1}, \theta_k^{l+1}) - F(w_k^l, \theta_k^l) = F_{w_k^{l+1}}(\theta_k^{l+1}) - F_{w_k^{l+1}}(\theta_k^l) + F_{\theta_k^l}(w_k^{l+1}) - F_{\theta_k^l}(w_k^l).$$

By the Lemma 3.1, we have

$$F_{w_k^{l+1}}(\theta_k^{l+1}) - F_{w_k^{l+1}}(\theta_k^l) \leq 0. \tag{3.32}$$

For fixed θ_k^l, w_k^{l+1} is computed by DCA1 using w_k^l as initialization, then we reduce from (i) of Theorem 3.1 that

$$F_{\theta_k^l}(w_k^{l+1}) - F_{\theta_k^l}(w_k^l) \leq 0. \tag{3.33}$$

Thus, we have $F(w_k^{l+1}, \theta_k^{l+1}) - F(w_k^l, \theta_k^l) \leq 0$.

(ii) We assume that the sequence $\{(w_k^l, \theta_k^l)\}_l$ is bounded. Let (w_k^*, θ_k^*) be a limit point of $\{(w_k^l, \theta_k^l)\}_l$. Thus, there exists a subsequence $(w_k^{l_t}, \theta_k^{l_t}) \to (w_k^*, \theta_k^*)$ as $t \to +\infty$. We will prove that (w_k^*, θ_k^*) is a critical point of the problem (3.5), i.e.

$$\emptyset \neq \partial_{w_k} G_1(w_k^*, \theta_k^*) \cap \partial_{w_k} H_1(w_k^*), \tag{3.34}$$

$$\{\theta_k^*\} = \arg\min_{\theta_k \in \Omega^k} F_{w_k^*}(\theta_k). \tag{3.35}$$

From the step 2 in ADCA1, we have $\theta_k^{l_t} = s_k^{l_t}/\sqrt{(s_k^{l_t})^T D s_k^{l_t}}$, where $s_k^{l_t} = (I - Q_{k-1}Q_{k-1}^T D)D^{-1}Y^T X w_k^{l_t}$. Taking the limit as $t \to +\infty$, we get $\theta_k^* = s_k^*/\sqrt{(s_k^*)^T D s_k^*}$, with $s_k^* = (I - Q_{k-1}Q_{k-1}^T D)D^{-1}Y^T X w_k^*$. By Lemma 3.1, it follows that the condition (3.35) holds.

Since $\{\theta_k^{l_t-1}\}_t$ is a subsequence of $\{\theta_k^l\}_l$, $\{\theta_k^{l_t-1}\}_t$ is also bounded. Without loss of generality, we can suppose (by extracting a subsequence if necessary) that the sequence $\theta_k^{l_t-1} \to \theta_k^{**}$ as $t \to +\infty$. We know that $\{F(w_k^l, \theta_k^l)\}_l$ is decreasing and it is bounded below by 0. Besides, we have

$$F(w_k^l, \theta_k^l) \leq F(w_k^l, \theta_k^{l-1}) \leq F(w_k^{l-1}, \theta_k^{l-1}).$$

Thus, $\lim_{l\to+\infty} F(w_k^l, \theta_k^l) = \lim_{l\to+\infty} F(w_k^l, \theta_k^{l-1}) = \inf_l F(w_k^l, \theta_k^l)$. Using the fact that F is continuous, we get

$$F(w_k^*, \theta_k^*) = \lim_{t\to+\infty} F(w_k^{l_t}, \theta_k^{l_t}) = \lim_{t\to+\infty} F(w_k^{l_t}, \theta_k^{l_t-1}) = F(w_k^*, \theta_k^{**}). \tag{3.36}$$

According to Lemma 3.1, the problem $\min_{\theta_k \in \Omega^k} F_{w_k^*}(\theta_k)$ has a unique solution. Hence, we deduce from (3.36) that $\theta_k^{**} = \theta_k^*$. From (ii) of Theorem 3.1, we have

$$\emptyset \neq \partial_{w_k} G_1(w_k^{l_t}, \theta_k^{l_t-1}) \cap \partial_{w_k} H_1(w_k^{l_t}).$$

Therefore, there exists $y_k^{l_t}$ such that

$$y_k^{l_t} \in \partial_{w_k} G_1(w_k^{l_t}, \theta_k^{l_t-1}) \cap \partial_{w_k} H_1(w_k^{l_t}). \tag{3.37}$$

From the computation of $\partial_{w_k} H_1(w_k^{l_t})$ it follows that the sequence $\{y_k^{l_t}\}_t$ is bounded. Thus, without loss of generality, we can suppose that the sequence $y_k^{l_t} \to y^*$ as $t \to +\infty$. We have

$$y_k^{l_t} \in \partial_{w_k} G_1(w_k^{l_t}, \theta_k^{l_t-1}) \Leftrightarrow G_1(w_k^{l_t}, \theta_k^{l_t-1}) + G_1^*(y_k^{l_t}, \theta_k^{l_t-1}) = \langle w_k^{l_t}, y_k^{l_t-1} \rangle, \tag{3.38}$$

where $G_1^*(., \theta_k^{l_t-1})$ is the conjugate function of $G_1(., \theta_k^{l_t-1})$ and

$$y_k^{l_t} \in \partial_{w_k} H_1(w_k^{l_t}) \Leftrightarrow H_1(w_k^{l_t}) + H_1^*(y_k^{l_t}) = \langle w_k^{l_t}, y_k^{l_t} \rangle. \tag{3.39}$$

Taking $t \to +\infty$ and using Lemma 2 in Pham Dinh and Le Thi (1997) we obtain

$$G_1(w_k^*, \theta_k^*) + G_1^*(y_k^*, \theta_k^*) = \langle w_k^*, y_k^* \rangle,$$
$$H_1(w_k^*) + H_1^*(y_k^*) = \langle w_k^*, y_k^* \rangle,$$

and hence $y_k^* \in \partial_{w_k} G_1(w_k^*, \theta_k^*) \cap \partial_{w_k} H_1(w_k^*)$. The proof of (ii) in then complete. □

3.5 Numerical experiments

3.5.1 Comparative algorithms

We have four main algorithms based on alternating schemes and DCA, named ADCA1-Exp, ADCA1-Cap, ADCA2-Exp and ADCA2-Cap. To demonstrate the usefulness of the proposed ℓ_0-sparse optimal scoring methods in view of many classification methods in the literature we will compare our algorithms with the five state-of-the-art algorithms: the SVM based method proposed in Fan et al. (2008) (S_SVM) (note that SVM is the most popular classification method), the three LDA based approaches proposed in Witten and Tibshirani (2011) (PLDA), Clemmensen et al. (2011) (SDA), Guo et al. (2007) (RDA), and the sparse partial least squares discriminant analysis Chun and Keles (2010) (SPLS_DA). In thesemethods the ℓ_1 regularization is used to deal with sparsity. We are also interested in the comparison between the four versions of our algorithms to evaluate the efficiency of the two DC approximations of the ℓ_0-norm as well as the two different DC formulations.

We also compare the proposed algorithms in this chapter with the ℓ_0-sparse Fisher LDA (DCA1-Cap) proposed in previous chapter and ℓ_0-sparse multiclass support vector machine proposed in Le Thi and Nguyen (2013) (SMSVM-Cap).

3.5.1.1 Penalized linear discriminant analysis (PLDA)

PLDA penalized the objective function of the Fisher's discriminant problem (2.7) with the ℓ_1 penalty on the discriminant vector, namely

$$\max_{w_k \in \mathbb{R}^p} \{w_k^T \Sigma_b w_k - \lambda_k ||w_k||_1 : w_k^T \Sigma w_k = 1;\ w_k^T \Sigma w_l = 0, l = 1, ..., k-1\}, \tag{3.40}$$

where λ_k is a nonnegative tuning parameter. The problem (3.40) is nonconvex. Witten and Tibshirani (2011) used the minorization-maximization approach for finding a local of this problem. This algorithm is in fact a version of DCA. The R package **penalizedLDA** is available from CRAN [2].

3.5.1.2 Sparse discriminant analysis (SDA)

SDA using the ℓ_1-norm was proposed by Clemmensen et al. (2011), that is

$$\begin{array}{ll} \min\limits_{w_k, \theta_k} & \{\frac{1}{n}||Y\theta_k - Xw_k||_2^2 + \gamma w_k^T \Omega w_k + \lambda ||w_k||_1\} \\ \text{subject to} & \frac{1}{n}\theta_k^T Y^T Y \theta_k = 1;\ \theta_k^T Y^T Y \theta_l = 0, l = 1, ..., k-1, \end{array} \tag{3.41}$$

2. http://cran.r-project.org/

where Ω is a positive definite matrix, γ and λ are nonnegative tuning parameters. In Clemmensen et al. (2011), the authors used an alternating scheme for finding a local optimum of this problem. The R package **sparseLDA** is available from CRAN.

3.5.1.3 Shrunken centroids regularized discriminant analysis (RDA)

RDA is based on the same underlying model as LDA (see (Guo et al., 2007)) and it regularizes the within-class covariance matrix used by LDA

$$\tilde{\Sigma} = \alpha\Sigma + (1-\alpha)I_p, \tag{3.42}$$

where $0 \leq \alpha \leq 1$. In order to perform feature selection, one can perform soft-thresholding of the quantity $\tilde{\Sigma}^{-1}\mu_k$. That is, we compute

$$\text{sgn}(\tilde{\Sigma}^{-1}\mu_k)(|\tilde{\Sigma}^{-1}\mu_k| - \delta)_+, \tag{3.43}$$

where δ is a nonnegative tuning parameter. The R package **rda** is available from CRAN.

3.5.1.4 Sparse partial least squares discriminant analysis (SPLS_DA)

SPLS_DA used the lasso to promote sparsity of a surrogate direction vector c instead of the original latent direction vector α, while keep α and c close (see (Chun and Keles, 2010)). That is, the first SPLS_DA direction vector solves

$$\begin{aligned} \min_{\alpha, c \in \mathbb{R}^p} \quad & \left\{-\kappa\alpha^T M\alpha + (1-\kappa)(c-\alpha)^T M(c-\alpha) + \lambda||c||_1 + \gamma||c||^2\right\} \\ \text{subject to} \quad & \alpha^T\alpha = 1, \end{aligned} \tag{3.44}$$

where κ is a tuning parameter with $0 \leq \kappa \leq 1$, and λ, γ are nonnegative tuning parameters. Performing the SPLS_DA method obtains $c_1, ..., c_s$ sparse surrogate direction vectors. Then, we obtain a classification rule by performing standard LDA on the low dimensional space $(Xc_1, ..., Xc_s)$. The R package **spls** is available from CRAN.

3.5.1.5 Sparse support vector machines (S_SVM)

S_SVM use the ℓ_1-regularization for the multi-class support vector machine. This method is supported by the LIBLINEAR package (Fan et al., 2008). The R package **LiblineaR** is also available from CRAN.

3.5.1.6 Sparse multiclass support vector machine (SMSVM-Cap)

Given n training observations $(x_i, y_i), i = 1, ..., n$, the classification rule of muticlass support vector machine (MSVM) is to classify an observation x to a class y defined by

$$y = \arg\max\langle w_k, x\rangle + b_k. \tag{3.45}$$

For the feature selection purpose, Le Thi and Nguyen (2013) proposes to use the $\ell_0 - \ell_2$ regularization for the MSVM model (Weston and Watkins, 1999), that leads to the so called ℓ_2-ℓ_0-MSVM problem which is defined by

$$\min_{(w,b,\xi)\in\Omega} C\sum_{i=1}^{n}\sum_{k\neq y_i}\xi_{ik} + \beta\sum_{k=1}^{Q}\|w_k\|_2^2 + \sum_{k=1}^{Q}\|w_k\|_0, \tag{3.46}$$

where C, β are nonnegative tuning parameters, and

$$\Omega = \left\{ \begin{array}{l} (w, b, xi) \in \mathbb{R}^{Q\times d} \times \mathbb{R}^{Q} \times \mathbb{R}_{+}^{n\times Q} : \\ \langle w_{y_i} - w_k, x_i\rangle + b_{y_i} - b_k \geq 1 - \xi_{ik}, \forall 1 \leq i \leq n, 1 \leq k \neq y_i \leq Q \end{array} \right\}.$$

Using the appropriate DC approximation functions of the ℓ_0-norm, the resulting the problems are the DC programs which are solved by DCA (Le Thi and Nguyen, 2013).

3.5.2 Datasets

We evaluate the performance of comparative algorithms on three synthetic datasets and a collection of real world datasets. Three synthetic 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 $p = 500$. 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.

The second simulation setup S2 includes two classes of multivariate normal distributions $N(\mu_1, \Sigma)$ and $N(\mu_2, \Sigma)$, each of dimension $p = 500$. The components of μ_1 are assumed to be 0 and for μ_2, $\mu_{2j} = 0.6$ if $j \leq 200$ and 0 otherwise. The covariance matrix Σ is the block diagonal matrix with five blocks of dimension 100×100 whose element (j, j') is $0.6^{|j-j'|}$. For each class, 100 training samples, 100 tuning samples and 10000 test samples are generated.

For the last setup S3, 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 two real datasets from UCI Machine Learning Repository and NIPS 2003 Feature Selection Challenge (Internet Advertisement, Gisette), nine real microarray gene expression datasets, and one dataset for handwritten character recognition (MNIST). All the datasets are preprocessed by normalizing each dimension of the

Table 3.1: Real datasets used in experiments.

Datasets	#Features	#Samples	#Classes
Internet Advertisement (ADV)	1558	3279	2
Colon Tumor[1] (COL)	2000	62	2
SRBCT[2] (SRB)	2308	83	4
Pencillium (PEN)	3754	36	3
Gisette (GIS)	5000	7000	2
ALL/AML[3] (ALL)	7129	72	3
Lung Cancer[4] (LUN)	12533	181	2
Leukemia[5] (LEU)	12558	248	6
MLL-Leukemia[6] (MLL)	12582	72	3
Protaste (PRO)	12600	136	2
Ovarian Cancer[7] (OVA)	15154	253	2
MNIST[8]	784	60000/10000	10

data to zero mean and unit variance. The detailed information of these datasets is summarized in Table 3.1.

3.5.3 Experimental setups

The proposed algorithms were implemented in the Visual Studio 2012, and performed on a PC Intel i7 CPU3770, 3.40 GHz of 8GB RAM.

In our experiments, the tuning parameters are γ, λ and α in (3.3) and (3.4). We fixed $\alpha = 5$ as suggested in Bradley and Mangasarian (1998), and γ, λ are performed by 5-fold cross-validation procedure on training or tuning set from sets of candidates given by $\Gamma = \{0.1, ..., 0.9, 1\}$ and

$$\Lambda = \{0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.012, 0.1, 0.15, 0.4, 0.6, 0.7, 0.9\},$$

respectively. By this way, we avoid performing tuning parameter selection on a three-dimensional grid.

The stop tolerance of DCA and ADCA is $\tau = 10^{-5}$ while the stop tolerance of the coordinate descent method is $\epsilon = 10^{-4}$. The starting point (w_k^0, θ_k^0) of ADCA is computed by $w_k^0 = 0$ and $\theta_k^0 = s_k^0/\sqrt{(s_k^0)^T D s_k^0}$, where $s_k^0 = (I - Q_{k-1}Q_{k-1}^T D)D^{-1}\theta_k^*$ with each element

1. http://genomics-pubs.princeton.edu/oncology/
2. http://research.nhgri.nih.gov/microarray/Supplement/
3. http://www-genome.wi.mit.edu
4. http://datam.i2r.a-star.edu.sg/datasets/krbd/
5. http://datam.i2r.a-star.edu.sg/datasets/krbd/
6. http://research.dfci.harvard.edu/korsmeyer/Supp_pub/Supp_Armstrong_Main.html
7. http://datam.i2r.a-star.edu.sg/datasets/krbd/
8. http://yann.lecun.com/exdb/mnist/

of θ_k^* is picked randomly in $[-1, 1]$. w_k^l is a starting point of DCA at l-th iteration of ADCA. We select relevant features as follows: feature i is deleted if $|w_{ki}| < 10^{-6}$ for all $k = 1, ..., K$.

3.5.4 Experiments on synthetic data

In this experiment, we generate training, tuning, and test sets in the same manner as described in Sect. 3.5.2. The tuning sets are used to choose the parameters λ, γ and the number of discriminant vectors used K, 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 3.2. In this table, the average number (#FS) and percentage (%FS) of selected features (standard deviations), the average percentage of accuracy of classifiers (ACC) and its standard deviation over 10 trials, the number of discriminant vectors used K (#DV), as well as training time in second (CPUs) are reported.

We observe from Table 3.2, in terms of feature selection, the DCA based algorithms are comparable and they give better results than PLDA, SDA, RDA, SPLS_DA and S_SVM. ADCA1-Cap, ADCA2-Exp and ADCA2-Cap give the best results on the S2, S2 and S3 datasets, respectively.

The DCA based algorithms not only provide a good performance in terms of feature selection, but also give a high accuracy of classifiers. ADCA1-Cap and ADCA2-Cap give the best accuracy of classifiers on 2/3 synthetic datasets.

The training time of all the algorithms is quite small: less than 2 seconds (except for the algorithm SDA).

3.5.5 Experiments on real datasets

For the experiments on the first eleven real datasets, we use the cross-validation scheme to validate the performance of various classifiers. Each real dataset is split into a training set containing 2/3 of the samples and a test set containing 1/3 of the samples. This process is repeated 10 times, each with a random choice of training set and test set. The parameter γ, λ and the number of discriminant vectors K which are used are chosen via 5-fold cross-validation.

The computational results given by ADCA1-Exp, ADCA1-Cap, ADCA2-Exp, ADCA2-Cap, PLDA, SDA, RDA and SPLS_DA are reported in Tables 3.3-3.4. We are interested in the efficiency (the sparsity and the accuracy of classifiers), the number of discriminant vectors used K, as well as the rapidity of these algorithms. The discriminant vectors can be used to visualize the datasets such as in Figure 3.1.

Table 3.2: Comparative results of ADCA1-Exp, ADCA1-Cap, ADCA2-Exp, ADCA2-Cap, PLDA, SDA, RDA, SPLS_DA and S_SVM on the synthetic data. Bold fonts indicate the best results in each row.

		ADCA1-Exp	ADCA1-Cap	ADCA2-Exp	ADCA2-Cap	PLDA	SDA	RDA	SPLS_DA	S_SVM
S1	ACC	**100** (0)	**100** (0)	**100** (0)	**100**(0)	82.94 (9.77)	**100** (0)	99.98 (0.04)	99.99 (0.02)	**100** (0)
	#FS	105.6 (3.28)	107.30 (1.55)	**97.1** (2.68)	108.4 (4.17)	170.2 (19.93)	114.09 (13.1)	104.9 (0.3)	140.6 (62.72)	154 (4.75)
	%FS	21.12 (0.67)	21.46 (0.31)	**19.42** (0.56)	21.68 (0.83)	34.04 (3.98)	22.81 (2.62)	20.98 (0.06)	28.12 (12.54)	30.8 (0.95)
	#DV	2	2	2	2	2	2	-	-	-
	CPUs	1.15 (0.26)	1.32 (0.17)	0.79 (0.16)	1.02 (0.15)	0.42 (0.06)	4.93 (3.71)	0.17 (0.008)	**0.11** (0.01)	0.23 (0.51)
S2	ACC	92.62 (1.15)	94.82 (0.58)	93.57 (0.71)	95.19 (0.53)	**96.62** (0.45)	94.41 (0.64)	92.26 (1.58)	95.36 (1.68)	92.4 (0.82)
	#FS	102.1 (4.76)	**89.4** (4.35)	89.8 (3.05)	100.4 (3.29)	159 (9.34)	108.1 (4.57)	134.2 (13.64)	98 (21.12)	98.5 (6.92)
	%FS	20.1 (0.95)	**17.88** (0.87)	17.96 (0.61)	20.08 (0.65)	31.9 (1.86)	21.62 (0.91)	26.84 (2.72)	19.6 (4.22)	19.7 (1.38)
	#DV	1	1	1	1	1	1	-	-	-
	CPUs	0.1 (0.03)	0.05 (0.01)	0.11 (0.01)	0.05 (0.009)	0.19 (0.12)	0.36 (0.04)	**0.03** (0.004)	**0.03** (0.01)	0.11 (0.02)
S3	ACC	96.26 (0.65)	**97.09** (0.56)	96.19 (0.81)	**97.09** (0.65)	96.58 (0.34)	96.85 (0.59)	97 (0.6)	96.83 (0.14)	50.65 (1.68)
	#FS	110.3 (4.9)	116.6 (5.4)	96.8 (6.4)	**92.6** (4.29)	293.7 (7.53)	113.6 (5.44)	240.7 (8.1)	123.8 (5.77)	170.5 (8.08)
	%FS	22.06 (0.98)	23.32 (1.08)	19.36 (1.28)	**18.52** (0.85)	58.74 (1.5)	22.72 (1.08)	48.14 (1.62)	24.76 (1.15)	34.1 (1.61)
	#DV	1	1	1	1	1	1	-	-	-
	CPUs	0.91 (0.24)	0.37 (0.06)	0.67 (0.18)	0.31 (0.04)	0.49 (0.17)	11.64 (3.08)	0.03 (0.006)	**0.02** (0.008)	0.12 (0.2.39)

Table 3.3: Comparative results of ADCA1-Exp, ADCA1-Cap, ADCA2-Exp, ADCA2-Cap, PLDA, SDA, RDA, SPLS_DA and S_SVM 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.

	ADCA1-Exp	ADCA1-Cap	ADCA2-Exp	ADCA2-Cap	PLDA	SDA	RDA	SPLS_DA	S_SVM
ADV	**34.7** (3.16)	253.3 (22.85)	144.5 (7.68)	259.8 (25.6)	516.9 (12.16)	321.9 (30.9)	426 (10.58)	1542 (7.93)	347 (44.55)
	2.22 (0.20)	16.11 (1.47)	9.27 (0.49)	16.67 (1.64)	33.17 (0.78)	20.66 (1.97)	27.34 (0.67)	99.03 (0.5)	22.27 (2.85)
COL	**5.80** (1.47)	23.9 (9.4)	6.5 (2.5)	29.5 (3.17)	434 (159.84)	31.5 (3.07)	1936.3 (11.39)	35 (5.23)	26.8 (4.66)
	0.29 (0.07)	1.19 (0.47)	0.32 (0.12)	1.47 (0.15)	21.7 (7.99)	1.57 (0.15)	96.81 (0.56)	1.75 (0.26)	1.34 (0.23)
SRB	54.08(7.25)	70.6 (6)	**32.3** (2.86)	73.7 (8.36)	1324.9 (147.15)	77.9 (3.91)	281 (5.16)	563.6 (179.14)	99.25 (1.56)
	2.37(0.31)	3.05 (0.26)	**1.39** (0.12)	3.19 (0.36)	57.4 (6.37)	3.37 (0.16)	12.2 (0.22)	24.41 (7.76)	3.7 (0.25)
PEN	**2**(0)	22.0 (4.42)	4.2 (2.56)	23.8 (5.13)	2383 (113.44)	4 (0)	3539.3 (2.23)	2679.1 (297.13)	26.5 (3.47)
	0.05 (0)	0.59 (0.12)	0.11 (0.06)	0.63 (0.13)	63.47 (3.02)	0.1 (0)	94.28 (0.06)	71.36 (7.91)	0.7 (0.09)
GIS	1015.7 (24.56)	1054.6 (25.49)	**960.5** (25.26)	1262.1 (21.97)	1413.7 (10.19)	1562.3 (24.68)	4933.5 (8.2)	3884.5 (23.76)	1124.4 (21.44)
	20.31 (0.49)	21.09 (0.5)	**19.21** (0.5)	25.24 (0.43)	28.27 (0.21)	31.24 (0.49)	98.67 (0.16)	77.69 (0.47)	22.48 (0.42)
ALL	**9.6** (1.57)	63.1 (11.4)	14.9 (2.36)	60.4 (5.06)	2701 (183.54)	71.4 (4.45)	7128.9 (0.3)	90.6 (15.23)	81.2 (7.85)
	0.13 (0.02)	0.88 (0.16)	0.2 (0.03)	0.84 (0.07)	37.88 (2.57)	1 (0.06)	99.99 (0.004)	1.27 (0.21)	1.13 (0.11)
LUN	**3.7** (0.43)	27.8 (2.44)	3.8 (0.6)	37.7 (4.73)	2614.2 (354.48)	47.2 (5.38)	12341.1 (19.66)	223.6 (14.45)	46.6 (5.23)
	0.02 (0.003)	0.22 (0.01)	0.3 (0.004)	0.3 (0.03)	20.85 (2.82)	0.37 (0.04)	98.25 (0.15)	1.78 (0.11)	0.37 (0.04)
LEU	**21.8** (2.61)	37.9 (5.65)	22.4 (1.68)	52.6 (9.11)	4421.9 (29.04)	44 (5.47)	10500.2 (6.66)	1438.8 (7.39)	313.1 (15.78)
	0.17 (0.02)	0.3 (0.05)	0.17 (0.01)	0.41 (0.07)	35.21 (0.23)	0.35 (0.04)	83.61 (0.05)	11.45 (5.89)	2.49 (0.12)
MLL	**96.3** (9.62)	132.6 (15.09)	103.1 (8.98)	150.5 (9.64)	6296.4 (147.73)	103.2 (3.69)	12581.3 (0.78)	456.7 (86.43)	142.9 (7.56)
	0.76 (0.07)	1.05 (0.12)	0.81 (0.07)	1.19 (0.07)	50.04 (1.17)	0.82 (0.03)	99.99 (0.01)	3.62 (0.68)	1.13 (0.06)
PRO	69.8 (12.41)	**45.5** (9.68)	51.6 (5.12)	55.9 (3.8)	75.4 (3.38)	68.7 (3.74)	12600 (0)	11174.9 (9.95)	76.4 (7.83)
	0.55 (0.09)	**0.36** (0.07)	0.4 (0.04)	0.44 (0.03)	0.59 (0.02)	0.54 (0.02)	100 (0)	9.32 (0.55)	0.6 (0.06)
OVA	**4** (0.47)	27.7 (2.35)	4.1 (0.3)	28.2 (2.44)	4160.8 (80.6)	8.1 (1.13)	93.8 (8.58)	60.4 (14.8)	56.8 (9.48)
	0.02 (0.003)	0.18 (0.01)	0.02 (0.001)	0.18(0.01)	27.45 (0.53)	0.05 (0.01)	0.62 (0.06)	0.39 (0.09)	0.37 (0.06)

Table 3.4: ADCA1-Exp, ADCA1-Cap, ADCA2-Exp, ADCA2-Cap, PLDA, SDA, RDA, SPLS_DA and S_SVM 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 discriminant vectors used K (the data is projected onto a K-dimensional space) (second row), and the average of training time in second and its standard deviation (third row). Bold fonts indicate the best results in each row.

	ADCA1-Exp	ADCA1-Cap	ADCA2-Exp	ADCA2-Cap	PLDA	SDA	RDA	SPLS_DA	S_SVM
ADV	96.94 (0.32)	97.27 (0.30)	**97.28** (0.31)	97.27 (0.28)	94.23 (0.31)	97.23 (0.27)	96.45 (0.36)	97.14 (0.17)	97.25 (0.92)
	1	1	1	1	1	1	-	-	-
	2.89 (0.64)	2.41 (0.46)	2.31 (0.88)	2(0.3)	**0.004**(0.007)	47.97 (3.21)	62.55 (0.96)	31.74 (3.13)	2.03 (0.14)
COL	84.32 (4.49)	**86.32** (5.94)	83.35 (3.86)	84.87 (5.81)	78.78 (6.54)	82.43 (7.89)	79.39 (5.8)	85.3 (4.89)	78.73 (4.62)
	1	1	1	1	1	1	-	-	-
	0.95 (0.5)	0.23 (0.23)	1.36 (0.76)	0.33 (0.24)	**0.01** (0.001)	0.34 (0.03)	0.23 (0.01)	**0.01** (0.01)	0.06 (0.01)
SRB	98.17 (2.60)	**99.64** (1.12)	98.19 (3.72)	99.27 (1.54)	97.44 (3.66)	98.54 (1.78)	96.73 (4.69)	97.44 (3.99)	99.25 (1.56)
	3	3	3	3	3	3	-	-	-
	16.56 (3.22)	5.13 (1.01)	16.07 (3.74)	5.85 (1.58)	**0.07** (0.01)	19.23 (6.81)	0.08 (0.01)	7.07 (2.08)	0.19 (0.01)
PEN	**100** (0)	**100** (0)	**100** (0)	**100** (0)	**100** (0)	**100**(0)	92.5 (10.83)	98.33 (3.33)	**100** (0)
	2	2	2	2	2	2	-	-	-
	4.25 (1.95)	7.08 (1.88)	5.23 (1.96)	6.57 (4.91)	**0.03** (0.01)	0.46 (0.36)	0.12 (0.01)	23.68 (3.62)	0.28 (0.03)
GIS	**97.64** (0.24)	97.59 (0.23)	97.39 (0.26)	97.51 (0.22)	86.88 (0.63)	97.58 (0.25)	84.52 (0.65)	96.39 (0.25)	97.6 (0.29)
	1	1	1	1	1	1	-	-	-
	30.11 (16.02)	48.51 (5.97)	24.03 (9.17)	32.56 (1.22)	**0.04** (0.08)	3665.69 (51.12)	829.91 (63.81)	256.59 (41.76)	14.62 (2.57)
ALL	95.43 (2.30)	**97.06** (2.03)	94.58 (3.25)	95.83 (2.63)	92.13 (6.68)	95.83 (2.63)	95.44 (3.9)	96.26 (4.35)	94.88 (4.43)
	2	2	2	2	2	2	-	-	-
	39.55 (8.92)	48.8 (9.87)	58.25 (13.22)	52.66 (16.44)	**0.05** (0.03)	52.2 (21.1)	0.42 (0.01)	31.24 (13.21)	0.47 (0.08)
LUN	98.67 (1.05)	98.84 (1.12)	98.17 (1.88)	98.17 (1.74)	**99.16** (0.73)	98.66 (1.45)	98 (1.45)	97.84 (1.29)	98.67 (1.52)
	1	1	1	1	1	1	-	-	-
	41.88 (6.59)	33.86 (3.15)	53.52 (25.82)	44.89 (6.15)	**0.06** (0.01)	6.25 (0.77)	1.07 (0.05)	0.21 (0.02)	0.90 (0.04)
LEU	94.69 (2.5)	95.52 (2.49)	94.08 (2.75)	95.05 (1.46)	96.86 (1.33)	95.88 (1.71)	**98.42** (1.08)	92.25 (1.65)	96.49 (1.83)
	5	5	5	5	5	5	-	-	-
	284.41 (52.78)	106.05 (30.65)	205.98 (34.32)	117.65 (15.71)	**0.21** (0.01)	53.17 (22.66)	1.66 (0.03)	168.37 (39.78)	2.23 (0.08)
MLL	97.08 (2.34)	**97.92** (2.19)	95.41 (4.73)	94.58 (4.58)	88.67 (7.41)	95.41 (5.08)	82.06 (17.27)	95.81 (3.72)	97.11 (4.4)
	2	2	2	2	2	2	-	-	-
	159.85 (33.32)	163.18 (51.26)	170.74 (42.5	159.44 (59.59)	0.15 (0.04)	157.96 (74.33)	**0.49**(0.01)	99.09 (41.01)	0.87 (0.07)
PRO	81.52 (5.97)	81.51 (2.79)	78.21 (6.06)	80.83 (34.68)	78.17 (6.17)	79.76 (5.73)	**87.91** (2.95)	82.62 (3.49)	80.5 (6.06)
	1	1	1	1	1	1	-	-	-
	57.21 (7.51)	30.58 (10.94)	63.77 (26.41)	27.11 (10.89)	1.73 (0.11)	14.3 (1.87)	0.77 (0.04)	**0.31** (0.02)	0.62 (0.03)
OVA	**100** (0)	**100** (0)	**100** (0)	**100** (0)	89.57 (4.27)	**100** (0)	99.64 (0.54)	99.05 (0.88)	**100** (0)
	1	1	1	1	1	1	-	-	-
	64.32 (25.95)	31.7 (7.27)	80.51 (30.12)	33.39 (7.93)	4.71 (0.1)	2.08 (0.24)	2.03 (0.06)	**0.38** (0.02)	1.47 (0.05)

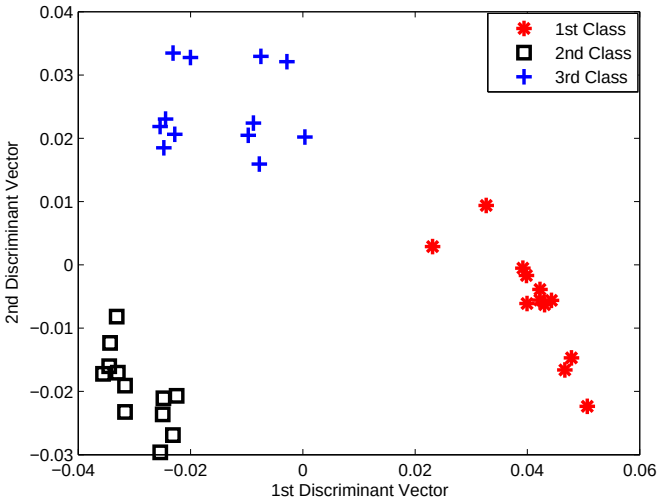

Figure 3.1: The penicillium data is projected onto the first two sparse discriminant vectors. The samples in each class are shown by using a distinct symbol.

Comments on computational results:
Sparsity. In on all the datasets, the classifiers obtained by the DCA based algorithms are sparser than those obtained by PLDA, SDA, RDA and SPLS_DA. The ADCA1-Exp approach is the best on 8 out of 11 datasets and ADCA2-Exp is the best on 2 out of 11 datasets. We also see that ADCA1-Cap achieves the best performance on the PRO dataset. Especially, on some datasets such as the COL, PEN, ALL, LUN and OVA datasets, ADCA1-Exp only selects a very small number of features (5.8, 2, 9.6, 3.7 and 4 features, respectively). ADCA1-Exp, ADCA1-Cap, ADCA2-Exp and ADCA2-Cap respectively select from $0,02\%$ to 20.31%, 0.18% to 21.09%, $0,02\%$ to 19.21% and 0.18% to 25.24% of features while PLDA, SDA, RDA, SPLS_DA and S_SVM respectively select from 0.59% to 63.47%, 0.05% to 31.24%, 0.62% to 100%, 0.39% to 99.03% and 0.37% to 22.48% of features.

Accuracy of classifiers. The accuracy of classifiers of the DCA based algorithms attain better than PLDA, SDA, RDA, SPLS_DA and S_SVM on 8/11 datasets. ADCA1-Cap is the best on 6/11 datasets. ADCA1-Exp and ADCA2-Exp are the best on 3/11 datasets. PLDA and RDA are slightly better than the DCA based algorithms on three datasets (LUN, LEU and PRO). This can be explained by the fact PLDA and RDA select much more features than the DCA based algorithms (the ratio is, respectively, 1042, 491 and 181 times on the LUN, LEU and PRO datasets).

Training time. Training time of the DCA based algorithms is quite small and acceptable: less than 81 seconds (except for datasets LEU and MLL).

For the experiments on the MNIST dataset. his dataset is quite different, as it has a very large number of observations (70000) while the number of features is not large (784). The parameter γ, λ and the number of discriminant vectors K which are used are chosen via 5-fold cross-validation on the training set. The test set is used to measure the accuracy of

Table 3.5: Comparative results of MNIST dataset in terms of the number (percentage) of selected features #FS (%FS), the percentage of accuracy of classifiers (ACC) on the test set, and training time in second. Bold fonts indicate the best results in each column.

Method	#FS (%FS)	ACC	Training time
ADCA1-Exp	492 (62.75)	89.8	765.18
ADCA1-Cap	529 (67.47)	**89.82**	937.98
ADCA2-Exp	**491** (62.62)	89.79	869.8
ADCA2-Cap	530 (67.6)	89.69	733.35
PLDA	610 (77.8)	81.59	**100.84**
SDA	641 (81.76)	87.43	317143.43
RDA	-	-	-
SPLS_DA	715 (91.19)	86.34	717.05
S_SVM	655 (83.54)	89.07	464.06

various classifiers trained on the training set. The computational results are reported on Table 3.5. Notably in this dataset, RDA is not able to perform since the amount of RAM is insufficient. From Table 3.5, we observe that the DCA based algorithms outperform PLDA, SDA, SPLS_DA and S_SVM in term of selected features and ADCA2-Exp is the best. In term of accuracy, the DCA based algorithms are also better than the other approaches. For the training time, PLDA is the fastest. Contrary to other datasets, here DCA based algorithm require much more training time, since the number of samples is very large.

Overall, the DCA based algorithms realize better a trade-off between accuracy and sparsity than the other algorithms. They suppress considerably the number of features (up to 99.98%) while the correctness of classification is quite good (from 78.17% to 100%).

3.5.6 Comparison with ℓ_0-sparse Fisher LDA and ℓ_0-sparse MSVM

Before finishing this chapter, we compare ADCA1-Cap with the sparse Fisher linear discriminant analysis (DCA1-Cap) proposed in the previous chapter and the sparse multiclass support vector machine (SMSVM-Cap) using the $\ell_2 + \ell_0$ regularization (Le Thi and Nguyen, 2013). The all three methods use the Capped-ℓ_1 approximation function of the ℓ_0-norm.

The comparative results of ADCA1-Cap, DCA1-Cap and SMSVM-Cap are reported in Table 3.6. We observe from this table that in terms of classification accuracy, ADCA1-Cap is better than DCA1-Cap while DCA1-Cap outperforms SMSVM-Cap. More precisely, ADCA1-Cap is the best on 10/14 datasets and DCA1-Cap is the best on 7/14 datasets in terms of classification accuracy. In terms of the sparsity, ADCA1-Cap and SMSVM-Cap give the best and second best results, respectively. Specifically, ADCA1-Cap, SMSVM-Cap and DCA1-Cap are respectively the best on 8/14, 4/14 and 2/14 datasets with respect to the sparsity. Concerning the running time, we notice that DCA1-Cap is the fast, following by ADCA1-Cap, while SMSVM-Cap runs much lower than DCA1-Cap

Table 3.6: Comparative results of ADCA1-Cap, DCA1-Cap and SMSVM-Cap. Bold fonts indicate the best results.

	Accuracy of classifiers			Number/percentage of selected features			CPU time in second		
	ADCA1-Cap	DCA1-Cap	SMSVM-Cap	ADCA1-Cap	DCA1-Cap	SMSVM-Cap	ADCA1-Cap	DCA1-Cap	SMSVM-Cap
S1	**100** (0)	**100** (0)	99.99 (0.02)	**107.3** (1.55) **21.46** (0.31)	110 (3) 22 (0.6)	115.4 (2.6) 23 (0.5)	1.32 (0.17)	**0.012** (0.001)	66.6 (2.3)
S2	**94.82** (0.58)	94.5 (0.8)	93.25 (0.5)	**89.4** (4.35) **17.88** (0.87)	100.9 (6.5) 20.1 (1.3)	113.1 (3.9) 22.6 (0.7)	0.05 (0.01)	**0.002** (0.001)	10.1 (0.3)
S3	**97.09** (0.56)	96.66 (1.2)	50.77 (2)	116.6 (5.4) 23.32 (1.08)	**114** (6.6) **22.8** (1.3)	135.3 (6.3) 27 (1.2)	0.37 (0.06)	**0.005** (0.003)	41 (1)
ADV	**97.27** (0.3)	94.25 (0.29)	96.14 (0.3)	253.3 (22.85) 16.11 (1.47)	467.71 (17.6) 30.02 (1.13)	**46.5** (5.7) **2.9** (0.3)	2.41 (0.46)	**0.003** (0.006)	10.4 (0.1)
COL	**86.32** (5.94)	81.57 (5.08)	77.05 (6.1)	23.9 (9.4) 1.19 (0.47)	**2** (0.8) **0.1** (0.04)	19 (2.5) 0.9 (0.1)	0.23 (0.23)	**0.001** (0.003)	11.2 (1.9)
SRB	**99.64** (1.12)	99.27 (1.46)	98.92 (1.6)	70.6 (6) 3.05 (0.26)	552.3 (674.62) 23.93 (29.23)	**27.8** (3) **1.2** (0.1)	5.13 (1.01)	**0.048** (0.037)	378.6 (28.8)
PEN	**100** (0)	**100** (0)	96 (5.8)	**22.0** (4.42) **0.59** (0.12)	2312.08 (79.58) 61.59 (2.12)	31.5 (1.7) 0.84 (0.04)	7.08 (1.88)	**0.056** (0.023)	89.1 (7.4)
GIS	**97.59** (0.23)	87.45 (0.63)	97.04 (0.2)	1054.6 (25.49) 21.09 (0.5)	2528 (259) 50.56 (5.18)	**627.1** (24.6) **12.5** (0.4)	48.51 (5.97)	**0.008** (0.012)	4263 (2094.6)
ALL	**97.06** (2.03)	96.24 (3.64)	92.27 (4.7)	**63.1** (11.4) 0.88 (0.16)	2437.9 (268.14) 34.19 (3.76)	74.2 (7.8) 1.04 (0.1)	48.8 (9.87)	**7.47** (0.05)	572.9 (283.2)
LUN	98.84 (1.12)	**99.34** (0.81)	98.45 (1.8)	**27.8** (2.44) 0.22 (0.01)	1542.81 (1265.83) 12.31 (10.1)	37.2 (4.7) 0.29 (0.03)	33.86 (3.15)	**0.036** (0.016)	837.28 (201.73)
LEU	95.52 (2.49)	**96.99** (1.32)	93.2 (3.1)	**37.9** (5.65) **0.3** (0.05)	3549.38 (1748.89) 28.21 (13.9)	173.8 (28.54) 1.38 (0.22)	106.05 (30.65)	**0.27** (0.089)	5482.48 (284.67)
MLL	97.92 (2.19)	**98.48** (2.17)	92.85 (3.17)	**132.6** (15.09) **1.05** (0.12)	6398 (388.61) 50.85 (3.08)	236.1 (8.26) 1.87 (0.06)	163.18 (51.28)	**7.68** (0.13)	3719.3 (1725.52)
PRO	81.51 (2.79)	**89.92** (2.1)	89.46 (4.3)	45.5(9.68) 0.36 (0.07)	5898.06 (2963.52) 46.81 (23.52)	**26.4** 0.2 (0.02)	30.58 (10.94)	**0.82** (0.03)	486.9 (45.6)
OVA	**100** (0)	**100** (0)	**100** (0)	**27.7** (2.35) **0.18** (0.01)	4870.49 (2712.56) 32.14 (17.9)	30.1 (3.1) 0.2 (0.02)	31.7 (7.27)	**1.36** (0.004)	3356.7 (146.6)

and ADCA1-Cap. DCA1-Cap runs extremely quickly (less than 8s). This is explained by using the diagonal estimate matrix of the within-class covariance matrix in the model, hence DCA1-Cap is explicitly computed at each iteration. We also note that SMSVM-Cap is very expensive when p is quite large, because its convex quadratic sub-problem has $2Qp + Q + (Q-1)n$ variables and $2(Q-1)n + Qp$ constraints.

3.6 Conclusion

We have proposed efficient approaches for solving the Sparse Optimal Scoring (SOS) problem using the ℓ_0 regularization. Among several sparse inducing functions of the ℓ_0-norm we used two appropriate approximations functions, and reformulated the SOS problem as continuous nonconvex optimization problems. We proposed two DC formulations of the approximate SOS problems and then investigated the alternating schemes based on DCA for them. Overall, four DCA based algorithms have been developed in order to exploit the nice effect of DC decompositions / DC formulations on one part, and the efficiency of the approximate functions on another part. The research developed in this chapter permitted us to highlight the following observations / recommendations:

• About the two DC formulations and their resulting DCA based algorithms: The second formulation seems to be more complicated as it needs an additional variable z. However, the use of z does not affect the complexity of DCA2: the two convex problems in DCA1 and DCA2 have the same form and the same dimension. Therefore, intuitively we can say that DCA1 and DCA2 have the same complexity. Regarding the behavior of algorithms, with the same approximate function, DCA2 seems to be more interesting in the sense that DCA2 is a re-weighted ℓ_1 type algorithm while DCA1 is simply a perturbed ℓ_1 algorithm (the coefficient of the ℓ_1-term is fixed during DCA scheme). In fact, by updating the coefficients of the ℓ_1-term in the convex problem at each iteration, DCA2 likely furnishes a better solution, consequently ADCA2 likely gives a better classification accuracy than ADCA1. This observation and the numerical results on the synthetic data suggest us to promote the use of DCA2 when we know that the data following a multivariate normal / Gaussian distribution. Meanwhile, averagely speaking, DCA1 and DCA2 are comparable on classification and CPU time. As for sparsity, it depends significantly on the approximate function.

• About the two approximate functions and their resulting DCA based algorithms: Intuitively, as has been discussed in Le Thi et al. (2015), Capped-ℓ_1 is more interesting than the exponential approximation (Exp) since the resulting DCA has a finite convergence and it gives in almost always cases a local minimum (whereas, theoretically, a solution obtained by DCA using Exp is only a critical point). It is easy to check the local optimality condition and in our experiments the DCA using Capped-ℓ_1 gives a local minimum in all test problems. This superiority of Capped-ℓ_1 versus Exp is confirmed by the numerical results in our experiments: generally speaking, in the same problem formulation, ADCA with Capped-ℓ_1 gives a better classification accuracy than ADCA using Exp. However, like numerical results in numerous previous works (see e.g. (Le Thi et al., 2015, 2014a,

2008; Ong and Le Thi, 2013a)), our numerical results show that Exp is more efficient than Capped-ℓ_1 when promoting sparsity (except for only one data set S3). Hence the users are recommended to use Capped-ℓ_1 (resp. Exp) when the classification accuracy (resp. the sparsity of classifier) is the most important criterion in the considered classification problem.

• Overall, the four versions of DCA based algorithms are comparable and the numerical results showed that ADCA1-Exp seems to be the most promising algorithm that realizes a good trade-off between accuracy and sparsity.

The efficiency of the four proposed methods have been compared with five standard algorithms which use the ℓ_1 regularization. The computational results show the robustness and the effectiveness of the DCA based algorithms and their superiority with respect to these standard approaches. It turns out that

• The ℓ_0 sparse optimal scoring methods are more efficient than other algorithms in classification of high dimensional data, especially when the number of features are much larger than the number of observations. The use of the proposed methods is strongly recommended for this type of data. It is worth noting that the SSVM is not suitable for such a data. Indeed, as has been seen in our numerical experiments, SVM using convex regularization like S_SVM is not efficient to feature selection, and therefore nonconvex approximations of the ℓ_0 regularization is necessary to deal with sparsity. However, if the sparse MSVM is also treated by the Capped-ℓ_1 (SMSVM-Cap) and the exponential concave function as we did on the optimal scoring methods, then the convex sub-problems in DCA are quadratic which require second order methods (see (Le Thi and Nguyen, 2013)). Such methods are very expensive when p is quite large, because the convex quadratic sub-problem has $2Qp+Q+(Q-1)n$ variables and $2(Q-1)n+Qp$ constraints (note however that the sub-problems in our SOS methods have $Kp(K<Q)$ variables).

• For any type of data, ℓ_0 regularization produces much better sparsity than ℓ_1 regularization. Hence we suggest to use the ℓ_0 sparse optimal scoring methods when the sparsity is significantly desired.

For future works, we plan to study more extensive applications of the SOS problem. In particular, we extend our works to more complex settings, such as the case where the observations from each class are drawn from a mixture of Gaussian distributions resulting in nonlinear separations between classes.

Chapter 4

Sparse Covariance Matrix Estimation

Abstract: *This chapter proposes a novel approach using the ℓ_0-norm regularization for the sparse covariance matrix estimation (SCME) problem. The objective function of SCME problem is composed of a nonconvex part and the ℓ_0 term which is discontinuous, and difficult to tackle as well. Appropriate DC (Difference of Convex functions) approximations of ℓ_0-norm are used that result to approximation SCME problems which are still nonconvex. DC programming and DCA (DC Algorithm), powerful tools in nonconvex programming framework, are investigated. Two DC formulations are proposed and then corresponding DCA schemes are developed. Two applications of the SCME problem are considered, that are classification via sparse quadratic discriminant analysis and portfolio optimization. A careful empirical experiment are performed through both simulated datasets and real datasets to study the performance of the proposed algorithms. Numerical results showed their efficiency and their superiority compared with seven state-of-the-art methods.*

4.1 Introduction

The estimation of covariance matrix is a common statistical problem that emerges from many scientific applications, and it quickly becomes an active and fast growing field of research. Much statistical analysis of such high dimensional data requires estimating a covariance matrix or its inverse. Several applications in numerous domains such as portfolio management and risk assessment (Ledoit and Wolf, 2003, 2004; Jagannathan and

1. The material of this chapter is based on the following works:
[1] Duy Nhat Phan, Hoai An Le Thi and Tao Pham Dinh. A DC Programming Approach for Sparse Estimation of a Covariance Matrix. Modelling Computation an Optimization in Information Systems and Management Sciences, Advances in Intelligent Systems and Computing, Volume 359, pp. 131-142, Springer (2015).
[2] Duy Nhat Phan and Hoai An Le Thi and Tao Pham Dinh. Sparse Covariance Matrix Estimation by DCA based Algorithms. Submitted.

Ma, 2003; Kourtis et al., 2012; Fan et al., 2013; Xue et al., 2012; Lai et al., 2011; Deng and Tsui, 2013), high dimensional classification (Guo et al., 2007; Witten and Tibshirani, 2011; Tibshirani et al., 2003), analysis of independence and conditional independence relationships between components in graphical models, statistical inference like controlling false discoveries in multiple testing (Leek and Storey, 2008; Efron, 2010), finding quantitative trait loci based on longitudinal data (Yap et al., 2009; Xiong et al., 2011), testing the capital asset pricing model (Sentana, 2009), ... have reported success stories of using covariance matrix estimation. For instance, principal component analysis (PCA) applies the eigen-decomposition of the covariance matrix for dimension reduction. In classification, linear discriminant analysis (LDA), quadratic discriminant analysis and other procedures exploit the inverse of a covariance matrix to compute the classification rule. In finance, portfolio optimization often uses the covariance matrix for minimizing the portfolio risk. More notably, graphical models are especially of interest in the analysis of gene expression data, since it is believed that genes operate in pathways, or networks. Graphical models based on gene expression data can provide a powerful tool for visualizing the relationships of genes and generating biological hypotheses (Toh and Horimoto, 2002; Dobra et al., 2004; Schafer and Strimmer, 2005a,b).

Let $Y = (Y_1, ..., Y_p)^T$ be a p-dimensional random vector with the covariance matrix $\Sigma = [\Sigma_{ij}]_{1 \le i,j \le p}$, where Σ_{ij} is the covariance between Y_i and Y_j. Suppose that we observe a sample including n observational data points $X_1, ..., X_n$ from a multivariate normal distribution $\mathcal{N}(0, \Sigma)$. The general purpose is to estimate Σ from this sample. This amounts to minimizing the negative log-likelihood function Mardia et al. (1979), Chap. 4 defined by

$$\ell(\Sigma) = \frac{n}{2} \left[\log \det \Sigma + \operatorname{tr}(\Sigma^{-1} S) + p \log 2\pi \right], \tag{4.1}$$

where $S = 1/n \sum_{i=1}^{n} X_i X_i^T$ is the sample covariance matrix under the assumption that the data is normalized to zero mean.

The problem is that with the increasing abundance of high-dimensional datasets, the sample covariance matrix S becomes an extremely noisy estimator of the covariance matrix, and besides, the number of parameters used to estimate grows quadratically with the number of variables. Intuitively, the most suitable approach to cope with this problem is finding an estimate of the covariance matrix which is as sparse as possible, since the sparsity leads to the effective reduction in the number of parameters. In addition, the sparsity is visualized by the so-called covariance graph (Chaudhuri et al., 2007). In the covariance graph, each node presents a random variable in a random vector and these nodes are connected by bidirectional edges if the covariances between the corresponding variables are nonzero. Note that the two random variables Y_i and Y_j are marginally independent if and only if the covariance between Y_i and Y_j is zero. Hence the zeros in a covariance matrix correspond to marginal independencies between variables, and sparse estimation of the covariance matrix is equivalent to estimating a covariance graph having a small number of edges. Thus the sparsity of the covariance matrix or its inverse is useful to improve the estimation accuracy and/or to explore the structure of the covariance graphical model.

In recent years, in connection with the Big data phenomenon, the sparse covariance matrix estimation (SCME) problem attracts a lot of attention of researchers and constitutes a challenge for the machine learning community. Existing methods for the SCME problems follow two directions. The first direction is to estimate the sparse inverse covariance matrix. A popular method in this direction consists in adding the lasso penalty (ℓ_1-norm regularization) on the entries of the inverse covariance matrix to the normal likelihood (see e.g. (Meinshausen and Buhlmann, 2006; Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2008; Rothman et al., 2008; Danaher et al., 2014; Cai et al., 2011; Zhang and Zou, 2014)). The second direction is to estimate directly the sparse covariance matrix. In this chapter, we follow the latter one.

A natural way to deal with sparsity in machine learning is using the ℓ_0-norm in the regularization term that leads to the following mathematical formulation of the SCME problem:

$$\min_{\Sigma \succ 0} \left\{ \log\det \Sigma + \mathrm{tr}(\Sigma^{-1} S) + \lambda ||\Sigma||_0 \right\}, \tag{4.2}$$

where λ is a nonnegative tuning parameter, the notation $\Sigma \succ 0$ means that Σ is symmetric positive definite, and $||\Sigma||_0$ denotes the ℓ_0-norm of Σ, i.e. the number of nonzero elements of matrix Σ. It is clear that the SCME problem is much more difficult than the classical estimation of covariance matrix problem. In view of optimization, the SCME problem includes a double difficulties: both the negative log-likelihood function and the ℓ_0-norm are nonconvex.

Note that the solution to (4.2) is positive definite. This property is crucial for any covariance matrix estimator from both methodological and practical aspects. Positive definite covariance matrices are required in all statistical procedures that use the normal distribution (for example, the principal component analysis, the parametric bootstrap method and the linear or quadratic discriminant analysis). Even some important statistical methods that do not use the normal distribution still need positive definite covariance matrix estimators such as some portfolio optimization methods based on the celebrated Markowitz model.

If S is nonsingular, then the problem (4.2) is equivalent to the following problem:

$$\min_{\Sigma \succeq \delta I_p} \left\{ \log\det \Sigma + \mathrm{tr}(\Sigma^{-1} S) + \lambda ||\Sigma||_0 \right\}, \tag{4.3}$$

for some $\delta > 0$ (see Bien and Tibshirani (2011)). Here, I_p denotes the $p \times p$ identity matrix, and the notation $\Sigma \succeq \delta I_p$ means that $\Sigma - \delta I_p$ is symmetric positive semidefinite. Note that if S is not full rank, we can replace S with $S + \epsilon I_p$ for some $\epsilon > 0$. In this setting, the observed data reside in a lower dimensional subspace of $\mathbb{R}^p$, and the addition of S and ϵI_p means the dataset with points that are not completely included in the span of the observed data is enhanced.

Optimization methods involving the ℓ_0-norm can be divided into three categories according to the way treating the ℓ_0 -norm: convex approximation, nonconvex approximation, and nonconvex exact reformulation. We refer to Le Thi et al. (2015) for an excellent

review on exact/approximation approaches to deal with the ℓ_0-norm. When the objective function (besides the ℓ_0-term) is convex, convex approximation techniques result to a convex optimization problem which is so far easy to solve. Unfortunately, due to the non-convexity of the negative log-likelihood function, the SCME problem (4.2) remains nonconvex with any approximation convex or nonconvex, of the ℓ_0-norm. How to deal with the ℓ_0-norm and how to treat the nonconvexity of the negative log-likelihood loss function are two crucial questions to be studied. Several works have been developed to the SCME problem, but designing an efficient method for it is still a challenge in this research field.

The approaches for solving the SCME problem can be divided into two groups: most of them are included in the first group that we name *convex approach*. Here one seeks to deter the non-convexity by replacing the negative log-likelihood function with a surrogate convex loss function and using the ℓ_1-norm or ℓ_2-norm instead of the ℓ_0-norm to deal with sparsity (see e.g. Deng and Tsui (2013); Liu et al. (2014); Rothman et al. (2009); Rothman (2012); Xue et al. (2012)). The resulting problems are then convex and so solvable, but it is unsurprising that the quality of solutions can be not good. In the second group including Bien and Tibshirani (2011); Lam and Fan (2009), the negative log-likelihood function is keeped and the ℓ_1-regularization is used to deal with sparsity. As mentioned above, due to the non-convexity of the negative log-likelihood function, the resulting problem is still nonconvex. In Bien and Tibshirani (2011) the authors applied the minorization-maximization (MM) approach for solving the resulting problem. Previous works on sparse optimization showed that the use of the ℓ_1-norm as a convex approximation of the ℓ_0-norm is not a good way in general (see Le Thi et al. (2015) and references therein). Instead, the approach approximating the ℓ_0-norm by a nonconvex continuous function (actually a DC function) is more suitable from theoretical perspective (Le Thi et al., 2015). The advantage of the ℓ_0-regularization versus the ℓ_1-regularization in learning with sparsity has been proved by several machine learning algorithms, however the difficulty caused by the ℓ_0-norm prevents researchers use the ℓ_0-regularization.

Our contributions. In this chapter, taking into account the advantages of some DC approximations of the ℓ_0-norm developed in Le Thi et al. (2015), we seek the most natural way to tackle the sparsity - the ℓ_0-regularization, and use these DC approximations to design two new models for the SCME problem. More precisely, we maintain the negative log-likelihood function in the problem (4.3) and replace the ℓ_0-regularization by these DC approximations. The resulting approximate problems are far more difficult than the existing models, but they are DC programs. With our best knowledge, this is the first time in the literature the ℓ_0-regularization is considered and its nonconvex approximations are used for the SCME problem.
Furthermore, we tackle the resulting approximate SCME problem by DC programming and DCA, powerful tools in nonconvex programming framework. Our motivation is based on the fact that DCA is a fast and scalable approach which has been successfully applied to many large-scale (smooth or non-smooth) non-convex programs in various domains of applied sciences, in particular in data analysis and machine learning (see e.g. Collobert et al. (2006); Krause and Singer (2004); Le Thi and Pham Dinh (2005); Le Thi et al.

(2012, 2014b, 2007, 2008, 2014a); Pham Dinh and Le Thi (1997, 1998, 2014); Le Hoai et al. (2013); Liu et al. (2005); Le Thi and Nguyen (2014)) and the list of reference on http://lita.sciences.univ-metz.fr/~lethi/DCA.html). We note in passing that the MM based approach proposed in Bien and Tibshirani (2011) for the SCME problem is also a version of DCA. Constituting the backbone of smooth/nonsmooth nonconvex programming and global optimization, DC programming and DCA address general DC programs of the form

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

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. The general DCA scheme is a philosophy but not an algorithm. There is not only one DCA but one family of DCAs for a considered problem. The main feature of DCA is that it is constructed from DC components but not the DC function F itself which has infinitely many DC decompositions, and there are as many DCA as there are DC decompositions. Such decompositions play a critical role in determining the speed of convergence, stability, robustness, and globality of sought solutions. Hence, what is a "good" DC decomposition is a crucial question when developing DCA for a DC program. The design of an efficient DCA for a concrete problem should be based on its special structure.

In this work, we propose two DC formulations of the approximate SCME problem based on two DC decompositions of its objective function. The first results from a natural DC decomposition while the second is introduced to exploit nice effects of DC decompositions. It turns out that the complexity of two corresponding DCA schemes are quite different, because that convex subproblems in the second DCA scheme can be solved by a very inexpensive algorithm. The ratio of gain between the two DCAs in terms of CPU times in our numerical experiments is up to 44 times. Among various existing sparse inducing functions, we are choosing, for implementing our algorithms, the piecewise linear approximation (Capped-ℓ_1) (Peleg and Meir, 2008) and the piecewise exponential approximation (Bradley and Mangasarian, 1998). This choice is motivated by the fact that the Capped-ℓ_1 has been proved theoretically in Le Thi et al. (2015) to be the tightness approximation while the piecewise exponential function has been showed to be efficient via the numerical results in numerous works (Bradley and Mangasarian, 1998; Le Thi et al., 2008, 2015, 2014c; Ong and Le Thi, 2013b). Applying DCA on two DC formulations with two approximations, we have then four DCA based algorithms for the approximate SCME problem. Special convergence analysis results of our algorithms are provided. We consider two important applications of the SCME problem in our experiments. The first is the quadratic discriminant analysis using sparse covariance matrices estimated by the proposed algorithms. The second is a portfolio optimization problem. Numerical experiments are carefully achieved on several test problems on both simulated datasets and real datasets with 11 algorithms including 7 state-of-the-art methods and the 4 proposed DCA schemes.

The rest of the chapter is organized as follows. The DCA based methods for solving the SCME problem is presented in Section 4.2. The numerical experiments are reported in

Section 4.3 and, finally, Section 4.4 concludes the chapter.

4.2 DCA for solving the sparse covariance matrix estimation SCME problem

4.2.1 The approximation SCME problem

The discontinuity of the ℓ_0-norm is overcome by using a DC approximation function. Define the step function $s : \mathbb{R} \to \mathbb{R}$ by $s(t) = 1$ for $t \neq 0$ and $s(t) = 0$ otherwise. Then $\|x\|_0 = \sum_{i=1}^{n} s(x_i)$. The idea of approximation methods is to replace the discontinuous step function by a continuous approximation η_α, where $\alpha > 0$ is a parameter controlling the tightness of approximation. The approximation of the ℓ_0-norm is then defined by

$$||\Sigma||_0 \approx \sum_{i,j} \eta_\alpha(\Sigma_{ij}). \tag{4.4}$$

This leads to the approximation SCME problem of (4.3) which takes the form

$$\min_{\Sigma \in \Omega} \left\{ F(\Sigma) = \log\det\Sigma + \operatorname{tr}(\Sigma^{-1}S) + \lambda \sum_{i,j} \eta_\alpha(\Sigma_{ij}) \right\}, \tag{4.5}$$

where $\Omega = \{\Sigma \in \mathbb{S}^p_{++} : \Sigma \succeq \delta I_p\}$.

We consider in this work two approximation functions $\eta_{\alpha,1}$ and $\eta_{\alpha,2}$ given by

$$\eta_{\alpha,1}(t) = 1 - \exp(-\alpha|t|) \quad \forall t \in \mathbb{R},$$

(the piecewise exponential concave approximation (Bradley and Mangasarian, 1998)) and

$$\eta_{\alpha,2}(t) = \min\{1, \alpha|t|\} \quad \forall t \in \mathbb{R},$$

(the Capped-ℓ_1 (Peleg and Meir, 2008)). In the sequel, for the sake of convenience, we use the common notation η_α to design both $\eta_{\alpha,1}$ and $\eta_{\alpha,2}$. It has been proved in Le Thi et al. (2015) that η_α is a DC function verifying the Assumption 1 of Le Thi et al. (2015), and with a suitable value of α the Capped-ℓ_1 approximation SCME problem (say, (4.5) when $\eta_\alpha = \eta_{\alpha,2}$) is equivalent to the original SCME problem (4.3).

We now investigate DCA for solving the nonconvex problem (4.5).

4.2.2 The first DCA scheme for solving the approximation SCME problem (4.5)

The approximation η_α can be presented as a DC function:

$$\eta_\alpha(t) = g(t) - h(t), \text{with} \quad g(t) = \alpha|t|, \tag{4.6}$$

and

$$h(t) = -1 + \alpha|t| + \exp(-\alpha|t|) \text{ if } \eta_\alpha = \eta_{\alpha,1}, \quad h(t) = -1 + \max\{1, \alpha|t|\} \text{ if } \eta_\alpha = \eta_{\alpha,2}. \quad (4.7)$$

In addition, we see that $\log\det\Sigma$ is concave in Σ (Boyd and Vanderberghe, 1979) while $\text{tr}(\Sigma^{-1}S)$ is convex. Indeed, we have

$$\text{tr}(\Sigma^{-1}S) = \sum_{i=1}^{n} X_i^T \Sigma^{-1} X_i,$$

and the function $X_i^T\Sigma^{-1}X_i$ is convex in Σ (Boyd and Vanderberghe, 1979). It follows that $\text{tr}(\Sigma^{-1}S)$ is convex. Consequently, the following DC decomposition of $F(\Sigma)$ seems to be natural

$$F(\Sigma) = G_1(\Sigma) - H_1(\Sigma), \quad (4.8)$$

where

$$G_1(\Sigma) = \text{tr}(\Sigma^{-1}S) + \lambda \sum_{i,j} g(\Sigma_{ij}) + \chi_\Omega(\Sigma),$$

and

$$H_1(\Sigma) = -\log\det\Sigma + \lambda \sum_{i,j} h(\Sigma_{ij}),$$

are clearly convex functions. Now, the optimization problem (4.5) can be rewritten as:

$$\min_{\Sigma\in\mathbb{R}^{p\times p}} \{F(\Sigma) = G_1(\Sigma) - H_1(\Sigma)\}. \quad (4.9)$$

According to the generic DCA scheme, at each iteration l, we have to compute $V^l \in \partial H_1(\Sigma^l)$ and then solve the convex program of the form (P_l), namely

$$\min_{\Sigma\in\mathbb{R}^{p\times p}} \{F_1(\Sigma) := G_1(\Sigma) - \langle V^l, \Sigma\rangle\}. \quad (4.10)$$

The computation of $V^l \in \partial H_1(\Sigma^l)$ depends on η_α. More precisely, for $\eta_\alpha = \eta_{\alpha,1}$, V^l is computed by

$$V_{ij}^l = -\left[(\Sigma^l)^{-1}\right]_{ij} + \text{sgn}(\Sigma_{ij}^l)\lambda\alpha\left(1 - \exp(-\alpha|\Sigma_{ij}^l|)\right), \quad (4.11)$$

where $\text{sgn}(\Sigma_{ij}^l)$ is the sign of Σ_{ij}^l. For $\eta_\alpha = \eta_{\alpha,2}$, V^l is calculated as

$$V_{ij}^l = \begin{cases} -\left[(\Sigma^l)^{-1}\right]_{ij} + \text{sgn}(\Sigma_{ij}^l)\lambda\alpha & \text{if} \quad \alpha|\Sigma_{ij}^l| \geq 1, \\ -\left[(\Sigma^l)^{-1}\right]_{ij} & \text{otherwise.} \end{cases} \quad (4.12)$$

For solving the convex problem (4.10), we have to use an iterative method in convex programming. For instance, we use the generalized gradient descent (GGD) algorithm (Beck and Teboulle, 2009) that iteratively solves the following problem at each iteration k:

$$\min_{\Sigma\succeq\delta I_p} \left\{ l_k(\Sigma) := \frac{1}{2\nu}||\Sigma - \Sigma^{l,k} + \nu[-(\Sigma^{l,k})^{-1}S(\Sigma^{l,k})^{-1} - V^l]||_F^2 + \lambda\alpha||\Sigma||_1 \right\}, \quad (4.13)$$

where $t > 0$ is a suitable stepsize. And then the problem (4.13) can be solved by the alternating direction method of multipliers (ADMM) (Boyd et al., 2011). The augmented Lagrangian function of this problem is

$$L_1(X,Y,Z) = \frac{1}{2\nu}||X - \Sigma^{l,k} + \nu[-(\Sigma^{l,k})^{-1}S(\Sigma^{l,k})^{-1} - V^l]||_F^2 + \lambda\alpha||Y||_1 + \\ + \langle Z, X - Y\rangle + \frac{\rho}{2}||X - Y||_F^2,$$

and ADMM solves the following problems at each iteration i:

$$X^{i+1} = \arg\min_{X \succeq \delta I_p} L_1(X, Y^i, Z^i),$$
$$Y^{i+1} = \arg\min_{Y \in \mathbb{R}^{p\times p}} L_1(X^{i+1}, Y, Z^i),$$
$$Z^{i+1} = X^i + \rho(X^{i+1} - Y^{i+1}).$$

Let $\mathcal{S}$ be the elementwise soft-thresholding operator defined by $\mathcal{S}(A,B)_{ij} = \text{sgn}(A_{ij})(|A_{ij}| - B_{ij})_+$. Then, finally, DCA for solving (4.9) can be described as follows.

DCA1

Initialization: $\Sigma^0 \succeq \delta I_p$, $l = 0$, $\tau > 0$, and compute δ.
repeat
1. Compute $V^l \in \partial H_1(\Sigma^l)$ according to (4.11) (resp. 4.12)) when $\eta_\alpha = \eta_{\alpha,1}$ (resp. $\eta_\alpha = \eta_{\alpha,2}$).
2. **Initialization (GGD):** $\Sigma^{l,0} = \Sigma^l, k = 0$, and $\nu, \epsilon_2 > 0$.
repeat
- Compute $\Delta^k = \Sigma^{l,k} + \nu[(\Sigma^{l,k})^{-1}S(\Sigma^{l,k})^{-1} + V^l]$.
- **Initialization (ADMM):** $Y^0 = \mathcal{S}\left(\Delta^k, \lambda\alpha\nu\right), Z^0 = 0, i = 0$, and $\epsilon_1, \rho > 0$.
repeat
- Compute $X^{i+1} = UD_\delta U^T$ where $D_\delta = \text{diag}(\max(D_{ii}, \delta))$ and $(\Delta^k + \nu\rho Y^i - \nu Z^i)/(1 + \nu\rho) = UDU^T$.
- Compute $Y^{i+1} = \mathcal{S}\left(X^{i+1} + Z^i/\rho, \lambda\alpha/\rho\right)$.
- Compute $Z^{i+1} = Z^i + \rho(X^{i+1} - Y^{i+1})$.
- $i \leftarrow i + 1$.
until $|l_k(X^i) - l_k(X^{i-1})| \le \epsilon_1$.
- $\Sigma^{l,k+1} = X^i$.
- $k \leftarrow k + 1$.
until $\|\Sigma^{l,k} - \Sigma^{l,k-1}\|_1 \le \epsilon_2$.
3. $\Sigma^{l+1} = \Sigma^{l,k}$.
4. $l \leftarrow l + 1$.
until $||\Sigma^l - \Sigma^{l-1}||_F \le \tau\left(||\Sigma^{l-1}||_F + 1\right)$ or $|F(\Sigma^l) - F(\Sigma^{l-1})| \le \tau\left(|F(\Sigma^{l-1})| + 1\right)$.

The complexity of one iteration of DCA1 is determined as follows. The computation of the subgradient of H_1 needs $O(p^3)$ operations. The combined GGD-ADMM for solving

convex subproblem requires $O\left(N_{iter}^{GGD} \times N_{iter}^{ADMM1} \times p^3\right)$ operations, where N_{iter}^{GGD} and N_{iter}^{ADMM1} are the number of iterations of the GGD and ADMM algorithms, respectively. Thus, the complexity of DCA1 is

$$O\left(N_{iter}^{DCA1} \times N_{iter}^{GGD} \times N_{iter}^{ADMM1} \times p^3\right), \tag{4.14}$$

where N_{iter}^{DCA1} denotes the number of iterations of DCA1.

We remark that DCA1 might be quite expensive because we have to use two iterative methods for solving the convex subproblem (4.10). This motivates us to consider another DC formulation of the problem (4.5).

4.2.3 The second DCA scheme for solving the approximation SCME problem (4.5)

Observing that the convex problem (4.10) is difficult due to the presence of the term $\text{tr}(\Sigma^{-1}S)$ in G_1, we seek another DC decomposition by moving $\text{tr}(\Sigma^{-1}S)$ to the second DC component. We then propose the following second DC formulation of the problem (4.5):

$$\min_{\Sigma \in \mathbb{R}^{p\times p}} \{F(\Sigma) = G_2(\Sigma) - H_2(\Sigma)\}, \tag{4.15}$$

where

$$G_2(\Sigma) = \frac{\mu}{2}||\Sigma||_F^2 + \lambda \sum_{i,j} g(\Sigma_{ij}) + \chi_\Omega(\Sigma), \tag{4.16}$$

and

$$H_2(\Sigma) = \frac{\mu}{2}||\Sigma||_F^2 - \text{tr}(\Sigma^{-1}S) - \log\det\Sigma + \lambda \sum_{i,j} h(\Sigma_{ij}), \tag{4.17}$$

are convex functions when μ is large enough. For estimating μ, we state the following lemma.

Lemma 4.1 *If $\mu \geq 2||S||_2\delta^{-3}$, then $H_2(\Sigma)$ is convex.*

Proof : Since the function $-\log\det\Sigma + \lambda\sum_{i,j} h(\Sigma_{ij})$ is convex and the sum of two convex functions is also convex, it is sufficient to show that $\frac{\mu}{2}||\Sigma||_F^2 - \text{tr}(\Sigma^{-1}S)$ becomes convex, i.e. μ is greater than the spectral radius of the Hessian matrix of $\Lambda(\Sigma) = \text{tr}(\Sigma^{-1}S)$. We have

$$\rho(\nabla^2\Lambda(\Sigma)) \leq ||\nabla^2\Lambda(\Sigma)||_2, \tag{4.18}$$

where $\rho(\nabla^2\Lambda(\Sigma))$ and $||\nabla^2\Lambda(\Sigma)||_2$ are the spectral radius and the spectral norm of the Hessian matrix of $\Lambda(\Sigma)$, respectively. The differential $d\Lambda(\Sigma)$ of $\Lambda(\Sigma)$ is defined by

$$d\Lambda(\Sigma) = \text{tr}\left[(d\Sigma^{-1})S\right] = \text{tr}\left[-\Sigma^{-1}(d\Sigma)\Sigma^{-1}S\right] = \text{tr}\left(-\Sigma^{-1}S\Sigma^{-1}d\Sigma\right).$$

Hence, we get the gradient of $\Lambda(\Sigma)$ as follows

$$\nabla\Lambda(\Sigma) = \frac{d}{d\Sigma}\Lambda(\Sigma) = -\Sigma^{-1}S\Sigma^{-1}. \tag{4.19}$$

We recall the product rule for matrix derivatives

$$\frac{d}{d\Sigma}f(\Sigma)g(\Sigma) = \left(g(\Sigma)^T \otimes I_p\right)\frac{d}{d\Sigma}f(\Sigma) + \left(I_p \otimes f(\Sigma)\right)\frac{d}{d\Sigma}g(\Sigma),$$

where $f, g : \mathbb{R}^{p\times p} \to \mathbb{R}^{p\times p}$. Using this product rule, we have

$$\frac{d}{d\Sigma}\nabla\Lambda(\Sigma) = -\left[(S\Sigma^{-1})^T \otimes I_p\right]\frac{d}{d\Sigma}\Sigma^{-1} - \left(I_p \otimes \Sigma^{-1}\right)\frac{d}{d\Sigma}S\Sigma^{-1}.$$

Moreover, by the product rule, we also obtain $\frac{d}{d\Sigma}\Sigma^{-1} = -\Sigma^{-1} \otimes \Sigma^{-1}$ and

$$\frac{d}{d\Sigma}S\Sigma^{-1} = \left[(\Sigma^{-1})^T \otimes I_p\right]\frac{d}{d\Sigma}S + \left(I_p \otimes S\right)\frac{d}{d\Sigma}\Sigma^{-1} = \left(I_p \otimes S\right)\left(-\Sigma^{-1} \otimes \Sigma^{-1}\right).$$

Therefore, the Hessian of $\Lambda(\Sigma)$ is

$$\nabla^2\Lambda(\Sigma) = \left(\Sigma^{-1}S \otimes I_p\right)\left(\Sigma^{-1} \otimes \Sigma^{-1}\right) + \left(I_p \otimes \Sigma^{-1}\right)\left(I_p \otimes S\right)\left(\Sigma^{-1} \otimes \Sigma^{-1}\right).$$

Using the property $(A \otimes B)(C \otimes D) = AC \otimes BD$, we finally get

$$\nabla^2\Lambda(\Sigma) = \Sigma^{-1}S\Sigma^{-1} \otimes \Sigma^{-1} + \Sigma^{-1} \otimes \Sigma^{-1}S\Sigma^{-1}. \tag{4.20}$$

We can deduce from (4.20) that $||\nabla^2\Lambda(\Sigma)||_2 \leq 2||S||_2\delta^{-3}$, and then by (4.18) we have $\rho(\nabla^2\Lambda(\Sigma)) \leq 2||S||_2\delta^{-3}$. The lemma is then proved. □

Remark 4.1 *From the Lemma 4.1, we can choose* $\mu = 2||S||_2\delta^{-3}$.

Applying DCA on (4.15), we have to compute $V^l \in \partial H_2(\Sigma^l)$ and then solve the convex program of the form (P_l), say

$$\min_{\Sigma\succeq\delta I_p}\{F_2(\Sigma) := \frac{\mu}{2}||\Sigma||_F^2 + \lambda\alpha\|\Sigma\|_1 - \langle V^l, \Sigma\rangle\}. \tag{4.21}$$

The computation of V^l is given by, for $\eta_\alpha = \eta_{\alpha,1}$:

$$V_{ij}^l = \mu\Sigma_{ij}^l + \left[(\Sigma^l)^{-1}S(\Sigma^l)^{-1}\right]_{ij} - \left[(\Sigma^l)^{-1}\right]_{ij} + \mathrm{sgn}(\Sigma_{ij}^l)\lambda\alpha\left(1 - \exp(-\alpha|\Sigma_{ij}^l|)\right), \tag{4.22}$$

and for $\eta_\alpha = \eta_{\alpha,2}$:

$$V_{ij}^l = \begin{cases} \mu\Sigma_{ij}^l + \left[(\Sigma^l)^{-1}S(\Sigma^l)^{-1}\right]_{ij} - \left[(\Sigma^l)^{-1}\right]_{ij} + \mathrm{sgn}(\Sigma_{ij}^l)\lambda\alpha & \text{if } \alpha|\Sigma_{ij}^l| \geq 1, \\ \mu\Sigma_{ij}^l + \left[(\Sigma^l)^{-1}S(\Sigma^l)^{-1}\right]_{ij} - \left[(\Sigma^l)^{-1}\right]_{ij} & \text{otherwise.} \end{cases}. \tag{4.23}$$

For solving the convex subproblem (4.21), we use the ADMM algorithm. The augmented Lagrangian function of (4.21) is

$$L_2(\Sigma, X, Y) = \frac{\mu}{2}||\Sigma||_F^2 - \langle V^l, \Sigma\rangle + \lambda\alpha||X||_1 + \langle Y, \Sigma - X\rangle + \frac{\rho}{2}||\Sigma - X||_F^2.$$

More specifically, ADMM solves the following problems at each iteration k:

$$\Sigma^{l,k+1} = \arg\min_{\Sigma \succeq \delta I_p} L_2(\Sigma, X^k, Y^k) \tag{4.24}$$

$$X^{k+1} = \arg\min_{X \in \mathbb{R}^{p\times p}} L_2(\Sigma^{l,k+1}, X, Y^k) \tag{4.25}$$

$$Y^{k+1} = Y^k + \rho(\Sigma^{l,k+1} - X^{k+1}). \tag{4.26}$$

Hence, DCA for solving (4.15) can be described as follows.

DCA2

Initialization: $\Sigma^0 \succeq \delta I_p$, $\tau > 0$, $l = 0$, and compute δ, μ.
repeat
1. Compute $V^l \in \partial H_2(\Sigma^l)$ according to (4.22) (resp. 4.23)) when $\eta_\alpha = \eta_{\alpha,1}$ (resp. $\eta_\alpha = \eta_{\alpha,2}$).
2. Initialization (ADMM): $X^0 = S\left(V^l/\mu, \lambda\alpha/\mu\right), Y^0 = 0, k = 0$, and $\rho, \epsilon_1 > 0$.
repeat
1. Compute $\Sigma^{l,k+1} = UD_\delta U^T$ where $D_\delta = \text{diag}(\max(D_{ii}, \delta))$ and $(V^l - Y^k + \rho X^k)/(\mu + \rho) = UDU^T$.
2. Compute $X^{k+1} = S\left(\Sigma^{l,k+1} + Y^k/\rho, \lambda\alpha/\rho\right)$.
3. Compute $Y^{k+1} = Y^k + \rho(\Sigma^{l,k+1} - X^{k+1})$
4. $k \leftarrow k + 1$.
until $\|\Sigma^{l,k} - \Sigma^{l,k-1}\|_F \leq \epsilon_1$.
3. $\Sigma^{l+1} = \Sigma^{l,k}$.
4. $l \leftarrow l + 1$.
until $||\Sigma^l - \Sigma^{l-1}||_F \leq \tau\left(||\Sigma^{l-1}||_F + 1\right)$ or $|F(\Sigma^l) - F(\Sigma^{l-1})| \leq \tau\left(|F(\Sigma^{l-1})| + 1\right)$.

We observe that the convex subproblem (4.21) is easier to solve than the convex subproblem (4.10) in DCA1. It requires only one iterative algorithm (ADMM) which is explicit at each iteration. The complexity of DCA2 is

$$O\left(N_{iter}^{DCA2} \times N_{iter}^{ADMM2} \times p^3\right), \tag{4.27}$$

where N_{iter}^{DCA2} and N_{iter}^{ADMM2} denote the number of iterations of DCA2 and ADMM, respectively.

4.2.4 Convergence analysis

Now we will prove the convergence properties of DCA1 and DCA2. For a DC program, the convergence of DCA is guaranteed when the optimal value is finite and the sequence

generated by DCA is bounded.

Lemma 4.2 *The optimal value of the problem* (4.5) *is finite.*

Proof : Since Σ is positive definite and S is positive semidefinite, then $\mathrm{tr}(\Sigma^{-1}S)$ is always nonnegative. On the other hand, $\lambda\sum_{i,j}\eta_\alpha(\Sigma_{ij}) > 0$ and $\log\det\Sigma \geq p\log\delta$ for all $\Sigma \succeq \delta I_p$. Hence $F(\Sigma) > p\log\delta$ for all $\Sigma \succeq \delta I_p$. The lemma has been proved. □

Lemma 4.3 *Let $A, B \in \mathbb{S}^p_{++}$ be the positive definite matrices. We have*
(a) $\lambda_{\min}(A+B) \geq \lambda_{\min}(A) + \lambda_{\min}(B)$.
(b) $\lambda_{\min}(AB) \geq \lambda_{\min}(A)\lambda_{\min}(B)$.
(c) $\lambda_{\min}(A \otimes B) \geq \lambda_{\min}(A)\lambda_{\min}(B)$.

Proof : (a) The part (a) is a consequence of Theorem III.2.1 in Bhatia (1997).

(b) For any positive definite matrix A, we have

$$\|A\| = \lambda_{\max}(A),$$

where $\|A\|$ denotes the operator norm of A. Hence, we get

$$\lambda_{\max}((AB)^{-1}) = \|B^{-1}A^{-1}\| \leq \|B^{-1}\|\|A^{-1}\| = \lambda_{\max}(B^{-1})\lambda_{\max}(A^{-1}).$$

Since the eigenvalues of the inverse matrix are the inverse of the eigenvalues, we obtain

$$\frac{1}{\lambda_{\min}(AB)} \leq \frac{1}{\lambda_{\min}(A)\lambda_{\min}(B)} \Rightarrow \lambda_{\min}(AB) \geq \lambda_{\min}(A)\lambda_{\min}(B).$$

(c) First of all, we will show that $\lambda_{\min}(A \otimes I_p) = \lambda_{\min}(A)$ and $\lambda_{\min}(I_p \otimes B) = \lambda_{\min}(B)$. Let $A = U\mathrm{diag}(a_1, ..., a_p)U^T$ and $B = V\mathrm{diag}(b_1, ..., b_p)V^T$ be the eigendecomposition of A and B, respectively. We have

$$A \otimes I_p = (U\mathrm{diag}(a_1, ..., a_p)U^T) \otimes I_p = (U \otimes I_p)(\mathrm{diag}(a_1, ..., a_p) \otimes I_p)(U^T \otimes I_p).$$

Note that $U^T \otimes I_p = (U \otimes I_p)^T$ and $\mathrm{diag}(a_1, ..., a_p) \otimes I_p$ is a $p^2 \times p^2$ diagonal matrix that its diagonal entries are $a_1, ..., a_1, ..., a_p$. Then, $a_1, ..., a_1, ..., a_p$ are also the eigenvalues of $A \otimes I_p$. It follows that $\lambda_{\min}(A \otimes I_p) = \lambda_{\min}(A)$. Similarly, we have

$$I_p \otimes B = (I_p \otimes V)(I_p \otimes \mathrm{diag}(b_1, ..., b_p))(I_p \otimes V^T).$$

Since $I_p \otimes V^T = (I_p \otimes V)^T$ and $I_p \otimes \mathrm{diag}(b_1, ..., b_p)$ is a $p^2 \times p^2$ diagonal matrix with the diagonal entries $b_1, ..., b_p, ..., b_p$, we get $\lambda_{\min}(I_p \otimes B) = \lambda_{\min}(B)$.

On the other hand, we have $A \otimes B = (A \otimes I_p)(I_p \otimes B)$. Then, from the part (b) of the lemma, we obtain

$$\lambda_{\min}(A \otimes B) \geq \lambda_{\min}(A \otimes I_p)\lambda_{\min}(I_p \otimes B) = \lambda_{\min}(A)\lambda_{\min}(B).$$

This completes the proof of Lemma 4.3. □

The convergence properties of DCA1 and DCA2 are given in the following theorem.

Theorem 4.1 *Let $\{\Sigma^l\}$ be the sequence generated by DCA1 (resp. DCA2), we have*
(a) $\{F(\Sigma^l)\}$ is decreasing.
(b) $\{\Sigma^l\}$ is bounded.
(c) $\sum_{l=0}^{+\infty} \|\Sigma^l - \Sigma^{l+1}\|_F^2 < +\infty$, and hence $\|\Sigma^l - \Sigma^{l+1}\|_F \to 0$ as $l \to +\infty$.
(d) The sequence $\{\Sigma^l\}_l$ has at least one limit point and every limit point of this sequence is a critical point of the problem (4.9) (resp. the problem (4.15)).

Proof : (a) is direct consequence of convergence properties of general DC programs.

(b) First, we will prove that the level set $L = \{\Sigma \in \Omega : F(\Sigma) \leq F(\Sigma^0)\}$ is bounded, $\forall \Sigma^0 \in \Omega$. Assume that this level set L is not bounded. Then, there is a sequence $\{\bar{\Sigma}^k\} \subset L$ such that $\|\bar{\Sigma}^k\|_F \to +\infty$ as $k \to +\infty$. Let $\lambda_1(\bar{\Sigma}^k), ..., \lambda_p(\bar{\Sigma}^k)$ be the eigenvalues of $\bar{\Sigma}^k$ with $\lambda_{\max}(\bar{\Sigma}^k) := \lambda_1(\bar{\Sigma}^k) \geq ... \geq \lambda_p(\bar{\Sigma}^k) := \lambda_{\min} \geq \delta$. We have

$$\|\bar{\Sigma}^k\|_F \leq \sqrt{p}\lambda_{\max}(\bar{\Sigma}^k) = \sqrt{p}\lambda_1(\bar{\Sigma}^k).$$

Besides, we have $\|\bar{\Sigma}^k\|_F \to +\infty$ as $k \to +\infty$. Thus $\lambda_1(\bar{\Sigma}^k) \to +\infty$ as $k \to +\infty$. Since $\mathrm{tr}((\bar{\Sigma}^k)^{-1}S)$ is always nonnegative and $\lambda \sum_{i,j} \eta_\alpha(\bar{\Sigma}^k) > 0$, we have

$$F(\bar{\Sigma}^k) > \log\det(\bar{\Sigma}^k) = \sum_{i=1}^{p} \log \lambda_i(\bar{\Sigma}^k) \geq \log \lambda_1(\bar{\Sigma}^k) + (p-1)\log\delta. \tag{4.28}$$

It follows that $F(\bar{\Sigma}^k) \to +\infty$ as $k \to +\infty$. But, the fact that $\bar{\Sigma}^k \in L$ implies $F(\bar{\Sigma}^k) \leq F(\Sigma^0)$ for all k. Thus, we have a contradiction, i.e. the level set $L = \{\Sigma \in \Omega : F(\Sigma) \leq F(\Sigma^0)\}$ is bounded, $\forall \Sigma^0 \in \Omega$.

Since the sequence $\{F(\Sigma^l)\}$ is monotonically decreasing, we have $\{\Sigma^l\} \subseteq \{\Sigma \in \Omega : F(\Sigma) \leq F(\Sigma^0)\}$ for some $\Sigma^0 \in \Omega$. This and the boundness of the level set L imply (b).

(c) We will show that the first DC components (G_1 and G_2) are strongly convex. By the definition of $G_2 := \frac{\mu}{2}||\Sigma||_F^2 + \lambda \sum_{i,j} g(\Sigma_{ij}) + \chi_\Omega(\Sigma)$, it is obviously that G_2 is strongly convex. As for G_1, it is sufficient to show that $\mathrm{tr}(\Sigma^{-1}S)$ is strongly convex. From the proof of Lemma 4.1, we have

$$\nabla^2\Lambda(\Sigma) = \Sigma^{-1}S\Sigma^{-1} \otimes \Sigma^{-1} + \Sigma^{-1} \otimes \Sigma^{-1}S\Sigma^{-1}. \tag{4.29}$$

Applying Lemma 4.3, we get

$$\begin{aligned}\lambda_{\min}(\nabla^2\Lambda(\Sigma)) &\geq \lambda_{\min}(\Sigma^{-1}S\Sigma^{-1}\otimes\Sigma^{-1}) + \lambda_{\min}(\Sigma^{-1}\otimes\Sigma^{-1}S\Sigma^{-1}) \\ &\geq 2\lambda_{\min}(\Sigma^{-1}S\Sigma^{-1})\lambda_{\min}(\Sigma^{-1}) \\ &\geq 2\lambda^3_{\min}(\Sigma^{-1})\lambda_{\min}(S) = \frac{2\lambda_{\min}(S)}{\lambda^3_{\max}(\Sigma)}.\end{aligned}$$

As mentioned in Section 4.1, we assumed that S is positive definite. If S is not full rank, we replace S by $S+\epsilon I_p$ for some $\epsilon>0$. Thus $\lambda_{\min}(\nabla^2\Lambda(\Sigma)) \geq \frac{2\lambda_{\min}(S)}{\lambda^3_{\max}(\Sigma)} > 0$. So Λ is strongly convex.

Let $\{\Sigma^l\}$ be the sequence generated by DCA1. If $\{\Sigma^l\}$ is generated by DCA2, then the part (c) of the theorem will be proved analogously. Recall that Σ^{l+1} is an optimal solution of the problem

$$\min_{\Sigma\in\mathbb{R}^{p\times p}} \left\{G_1(\Sigma) - \langle V^l, \Sigma\rangle\right\},$$

where $V^l \in \partial H_1(\Sigma^l)$. Then the first-order optimality condition holds at Σ^{l+1}, i.e, $0 \in \partial G_1(\Sigma^{l+1}) - V^l$ and then

$$V^l \in \partial G_1(\Sigma^{l+1}). \tag{4.30}$$

Hence, we have

$$G_1(\Sigma^l) \geq G_1(\Sigma^{l+1}) + \langle V^l, \Sigma^l - \Sigma^{l+1}\rangle + \frac{\rho(G_1)}{2}\|\Sigma^l - \Sigma^{l+1}\|_F^2, \tag{4.31}$$

where $\rho(G_1)$ is the modulus of the strong convexity of G_1. Since $V^l \in \partial H_1(\Sigma^l)$, we also have

$$H_1(\Sigma^{l+1}) \geq H_1(\Sigma^l) + \langle V^l, \Sigma^{l+1} - \Sigma^l\rangle. \tag{4.32}$$

Combining (4.31) and (4.32), we have

$$\begin{aligned}&G_1(\Sigma^l) \geq G_1(\Sigma^{l+1}) + H_1(\Sigma^l) - H_1(\Sigma^{l+1}) + \frac{\rho(G_1)}{2}\|\Sigma^l - \Sigma^{l+1}\|_F^2 \\ \Rightarrow &G_1(\Sigma^l) - H_1(\Sigma^l) \geq G_1(\Sigma^{l+1}) - H_1(\Sigma^{l+1}) + \frac{\rho(G_1)}{2}\|\Sigma^l - \Sigma^{l+1}\|_F^2 \\ \Rightarrow &F(\Sigma^l) - F(\Sigma^{l+1}) \geq \frac{\rho(G_1)}{2}\|\Sigma^l - \Sigma^{l+1}\|_F^2.\end{aligned}$$

Moreover, since G_1 is strongly convex, $\rho(G_1) > 0$. Hence, we obtain

$$\|\Sigma^l - \Sigma^{l+1}\|_F^2 \leq \frac{2}{\rho(G_1)}\left(F(\Sigma^l) - F(\Sigma^{l+1})\right). \tag{4.33}$$

Let N be a positive integer. Summing (4.33) from $l=0$ to N, we get

$$\sum_{l=0}^{N}\|\Sigma^l - \Sigma^{l+1}\|_F^2 \leq \frac{2}{\rho(G_1)}\left(F(\Sigma^0) - F(\Sigma^{N+1})\right). \tag{4.34}$$

On the other hand, from the proof of Lemma 4.2 we see that $F(\Sigma^{N+1}) \geq p\log(\delta)$. Combining this and (4.34) we get

$$\sum_{l=0}^{N} \|\Sigma^l - \Sigma^{l+1}\|_F^2 \leq \frac{2}{\rho(G_1)} \left(F(\Sigma^0) - p\log(\delta)\right).$$

Taking the limit as $N \to +\infty$, we obtain

$$\sum_{l=0}^{+\infty} \|\Sigma^l - \Sigma^{l+1}\|_F^2 < +\infty, \tag{4.35}$$

and hence $\lim_{l\to+\infty} \|\Sigma^l - \Sigma^{l+1}\|_F = 0$.

(d) is deduced from (b), Lemma 4.2 and the DCA's convergence property (iii) of Theorem 1.2 in Chapter 1. □

4.3 Numerical experiments

4.3.1 Comparative algorithms

For each algorithm, we use two DC approximations of the ℓ_0-norm. Hence we have four DCA based algorithms: DCA1-CaP, DCA1-PiE, DCA2-CaP and DCA2-PiE, where CaP and PiE denote the algorithm using the Capped-ℓ_1 ($\eta_\alpha = \eta_{\alpha,2}$) and the piecewise exponential approximation function ($\eta_\alpha = \eta_{\alpha,1}$), respectively. To study the performance of the proposed algorithms, we compare our algorithms with seven standard methods that cover all types of algorithms mentioned in Section 1.

- Methods follow the first direction - estimate the sparse inverse covariance matrix: CLIME (Cai et al., 2011) and SPME (Zhang and Zou, 2014).
- Methods follow the second direction (estimate directly the sparse covariance matrix), in the first group (the convex approach), i.e. they use surrogate convex loss functions of the negative log likelihood function and the ℓ_1-norm (resp. ℓ_2-norm): PDSCE (Rothman, 2012) (resp. PCME (Deng and Tsui, 2013)).
- Methods follow the second direction, in the second group - replace the ℓ_0-norm by the ℓ_1-norm: SPCOV1 and SPCOV2 (Bien and Tibshirani, 2011).

Moreover, for the two real applications (classification and portfolio selection) we consider in addition a method using the sample covariance matrix: the Quadratic Discriminant Algorithm (QDA). We also compare the four DCA based algorithms to study the performance of the two DC approximations of the ℓ_0-norm as well as the two different DC decompositions.

4.3.1.1 Constrained ℓ_1 mInimization approach to sparse precision Matrix Estimation (CLIME)

CLIME (Cai et al., 2011) estimated an inverse covariance matrix $\Omega = \Sigma^{-1}$ by solving the following problem

$$\min_{\Omega \in \mathbb{R}^{p \times p}} \{\|\Omega\|_1 : \|S\Omega - I\|_\infty \leq \lambda\}, \tag{4.36}$$

where λ is a tuning parameter. The convex program (4.36) can be further decomposed into p vector minimization problems which is solved by the primal dual interior method. The **clime** package for CLIME is also available from CRAN.

4.3.1.2 Sparse Precision Matrix Estimation via lasso penalized D-trace loss (SPME)

SPME (Zhang and Zou, 2014) used a surrogate loss function instead of the negative log likelihood function with the ℓ_1 penalty on the precision matrix $\Theta = \Sigma^{-1}$, namely

$$\min_{\Theta \succeq \epsilon I} \left\{ \frac{1}{2}\langle \Theta^2, S\rangle - \text{tr}(\Theta) + \lambda \|\Theta\|_{1,\text{off}} \right\}, \tag{4.37}$$

where λ is a tuning parameter and $\|\Theta\|_{1,\text{off}} = \sum_{i \neq j} |\Theta_{ij}|$. Zhang and Zou (2014) used the alternating direction method of multipliers for solving the problem (4.37). The source code of this method is available on the author' homepage (https://math.cos.ucf.edu/tengz/).

4.3.1.3 Positive Definite Sparse Covariance Estimators (PDSCE)

Rothman (2012) proposed the sparse covariance matrix estimator by solving the following problem

$$\min_{\Sigma \succ 0} \left\{ \frac{1}{2}\|\Sigma - S\|_F^2 - \tau \log\det(\Sigma) + \lambda \|\Sigma\|_1 \right\}, \tag{4.38}$$

where λ is a tuning parameter and $\tau > 0$ is fixed at a small value. The author developed a blockwise coordinate descent algorithm to compute the solution to (4.38). The PDSCE algorithm is included in **PDSCE** package of R software.

4.3.1.4 Penalized Covariance Matrix Estimation using a matrix logarithm transformation (PCME)

The PCME method (Deng and Tsui, 2013) used an estimate of covariance matrix $\hat{\Sigma} = \sum_{k=0}^{\infty} \frac{\hat{A}^k}{k!} \equiv \exp(\hat{A})$, where $\hat{A}$ solves the following problem

$$\min_{A \in \mathbb{S}^p} \left\{ \text{tr}(A) + \text{tr}[\exp(-A)S] + \lambda \text{tr}(A^2) \right\}, \tag{4.39}$$

where λ is a tuning parameter. The MATLAB code for PCME is available on http://www.tandfonline.com/doi/suppl/10.1080/10618600.2012.715556.

4.3.1.5 Sparse estimation of a COVariance Matrix (SPCOV)

Bien and Tibshirani (2011) considered the negative log-likelihood function with ℓ_1 penalty on the entries of the covariance matrix, namely

$$\min_{\Sigma \succ 0} \left\{ \log\det\Sigma + \mathrm{tr}(\Sigma^{-1}S) + \lambda||W \circ \Sigma||_1 \right\}, \tag{4.40}$$

where W is an arbitrary matrix with nonnegative elements. The problem (4.40) is non-convex. Bien and Tibshirani (2011) used the MM approach for solving this problem (that is in fact a version of DCA). Denote by SPCOV1 (reps. SPCOV2) the MM approach for solving the problem (4.40) with $W_{ij} = 0$ if $i = j$ and 1 otherwise (resp. with $W_{ij} = 0$ if $i = j$ and $W_{ij} = \frac{1}{|S_{ij}|}$ otherwise) (Bien and Tibshirani, 2011). The R package **spcov** for SPCOV1 and SPCOV2 is available from CRAN (http://cran.r-project.org/).

4.3.2 Experimental setups

The proposed algorithms are implemented in R software and all algorithms are performed on a PC Intel i7 CPU3770, 3.40 GHz of 8GB RAM.

In experiments, we set the stop tolerance $\tau = 10^{-4}$ for DCA based algorithms. The parameters of GGD and ADMM are chosen as proposed in Bien and Tibshirani (2011). The starting point Σ^0 of DCA is the sample covariance matrix S. The values of parameter λ and approximation parameter of the Capped-ℓ_1 are chosen through a 5-fold cross-validation procedure on training set. The approximation parameter of the piecewise exponential approximation function is chosen $\alpha = 5$ as suggested in Bradley and Mangasarian (1998).

The cross-validation procedure is described as follows (Bien and Tibshirani, 2011). For $\mathcal{A} \subseteq \{1, ..., n\}$, let $S_{\mathcal{A}} = |\mathcal{A}|^{-1} \sum_{i \in \mathcal{A}} X_i X_i^T$, and $\mathcal{A}_i^c$ denotes the component of $\mathcal{A}$. We divide $\{1, ..., n\}$ into 5 subsets, $\mathcal{A}_1, ..., \mathcal{A}_5$, and then compute

$$f(\lambda) = \frac{1}{5} \sum_{i=1}^{5} \ell \left\{ \hat{\Sigma}_\lambda(S_{\mathcal{A}_i^c}); S_{\mathcal{A}_i} \right\}, \tag{4.41}$$

where $\hat{\Sigma}_\lambda(S_{\mathcal{A}_i^c})$ is an estimate of the covariance matrix Σ with the parameter λ and $S_{\mathcal{A}_i^c}$, and $\ell \left\{ \hat{\Sigma}_\lambda(S_{\mathcal{A}_i^c}); S_{\mathcal{A}_i} \right\} = -\log\det\hat{\Sigma}_\lambda(S_{\mathcal{A}_i^c}) - \mathrm{tr}\left(\left[\hat{\Sigma}_\lambda(S_{\mathcal{A}_i^c}) \right]^{-1} S_{\mathcal{A}_i} \right)$. Finally, we choose $\hat{\lambda} = \arg\max_\lambda f(\lambda)$.

4.3.3 Numerical results on synthetic datasets

We evaluate the performance of DCA1-CaP, DCA2-CaP, DCA1-PiE and DCA2-PiE on four synthetic datasets. We generate $X = [X_1, ..., X_n]$ from a multivariate normal distribution $N_p(0, \Sigma)$, where Σ is a sparse symmetric positive definite matrix. We consider three types of covariance graphs and a moving average model as follows Bien and Tibshirani (2011):

Cliques model: $\Sigma = \text{diag}(\Sigma_1, ..., \Sigma_5)$, where $\Sigma_1, ..., \Sigma_5$ are dense matrices.

Hubs model: $\Sigma = \text{diag}(\Sigma_1, ..., \Sigma_5)$ again, however each submatrix Σ_k is zero except elements in the last row and the last column. This corresponds to a graph with five connected components each of which has all nodes connected to one particular node.

Random model: in this model, we take $\Sigma_{ij} = \Sigma_{ji}$ to be nonzero with the probability 0.02, independently of other elements.

First-order moving average model: we generate $\Sigma_{i,i-1} = \Sigma_{i-1,i}$ to be nonzero for $i = 2, ..., p$.

In the first three cases, the nonzero entries of matrix Σ are randomly drawn in the set $\{+1, -1\}$. In the moving average model, all nonzero values are set to be 0.4. In this experiment, for each covariance model, we generate 10 training sets with size $n = 200, p = 100$.

To evaluate the performance of each method, we consider three loss functions which are the root-mean-square error (RMSE), the entropy loss (EN), and the Kullback-Leibler loss (KL), respectively.

$$\begin{aligned} \text{RMSE} &= ||\hat{\Sigma} - \Sigma||_F/p, \\ \text{EN} &= -\log\det(\hat{\Sigma}\Sigma^{-1}) + \text{tr}(\hat{\Sigma}\Sigma^{-1}) - p, \\ \text{KL} &= -\log\det(\hat{\Sigma}^{-1}\Sigma) + \text{tr}(\hat{\Sigma}^{-1}\Sigma) - p, \end{aligned}$$

where $\hat{\Sigma}$ is a sparse estimate of the covariance matrix Σ.

The experimental results on synthetic datasets are given in Table 4.1. In this Table, the average of root-mean-square error (RMSE), entropy loss (EN), Kullback-Leibler loss (KL), number of nonzero elements (NZ), CPU time in seconds, and their standard deviations over 10 samples are reported.

We observe from Table 4.1 that in the cliques model, DCA2-CaP gives the lowest root-mean-square error while DCA2-PiE gives the best results in terms of the entropy loss and Kullback-Leibler loss. We further note that in terms of the sparsity, the number of the nonzero elements, the DCA based algorithms achieve much better performances than the other six approaches.

For the hubs and random models, the best and the second best performing methods with respect to the losses and the sparsity are DCA1-PiE and DCA2-PiE, respectively.

Table 4.1: Comparative results in terms of the average of root-mean-square error (RMSE), entropy loss (EN), Kullback-Leibler loss (KL), number of nonzero elements, CPU time in second (and their standard deviations) over 10 runs. Bold fonts indicate the best result in each row.

		DCA1-CaP	DCA2-CaP	DCA1-PiE	DCA2-PiE	SPCOV1	SPCOV2	CLIME	PCME	SPME	PDSCE
Cliques	RMSE	0.399	**0.381**	0.456	0.413	0.398	0.384	0.387	0.474	0.544	0.418
		(0.003)	(0.004)	(0.01)	(0.008)	(0.0.004)	(0.005)	(0.004)	(0.003)	(0.005)	(0.004)
	EN	15.37	14.15	14.99	**13.54**	15.77	16.83	83.54	27.85	31.81	80.42
		(0.95)	(0.61)	(1.07)	(0.7)	(0.57)	(0.53)	(3.32)	(0.36)	(0.62)	(2.68)
	KL	23.23	20.56	15.77	**14.29**	31.43	34.39	21.33	71.23	47.12	21.02
		(4.11)	(1.22)	(1.28)	(0.59)	(2.23)	(2.04)	(0.32)	(1.94)	(1.33)	(0.21)
	NZ	2623	2419.6	**1011.4**	1018.2	3775.4	3565.8	8998.4	9847	2624.4	2755.6
		(510.08)	(209.34)	(30.21)	(18.59)	(156.13)	(94.57)	(18.37)	(15.72)	(821.28)	(72.65)
	CPUs	338.3	70.77	130.97	114.06	180.04	112.99	687.95	223.87	33.51	**2.72**
		(103.55)	(14.08)	(14.91)	(61.96)	(17.84)	(8.9)	(21.12)	(1.36)	(1.3)	(0.06)
Hubs	RMSE	0.085	0.077	**0.062**	0.072	0.073	0.073	0.194	0.237	0.183	0.109
		(0.004)	(0.004)	(0.008)	(0.006)	(0.0.003)	(0.004)	(0.084)	(0.002)	(0.007)	(0.003)
	EN	3.32	3.09	**1.53**	1.98	3.54	2.63	264.8	29.03	61.46	141.61
		(0.25)	(0.34)	(0.23)	(0.38)	(0.27)	(0.21)	(56.5)	(0.76)	(1.53)	(2.35)
	KL	4.44	3.83	**1.72**	2.2	5.79	3.6	18.94	64.92	22.41	13.99
		(0.38)	(0.35)	(0.27)	(0.34)	(0.73)	(0.29)	(3.85)	(2.45)	(0.82)	(0.19)
	NZ	552.6	530.2	**301.2**	328.4	879	582.2	1174	9493.8	4640	597.6
		(25.02)	(21.36)	(5.38)	(12.51)	(33.06)	(24.5)	(124.21)	(30.68)	(1195.54)	(30.56)
	CPUs	96.1	53. 43	74.63	49.52	99.73	86.31	676.59	279.42	29.68	**3.24**
		(7.29)	(11.81)	(7.77)	(23)	(6.42)	(4.51)	(26.89)	(2.16)	(0.83)	(0.23)
Random	RMSE	0.096	0.086	**0.051**	0.052	0.086	0.066	0.089	0.177	0.125	0.083
		(0.001)	(0.002)	(0.003)	(0.003)	(0.0.002)	(0.002)	(0.002)	(0.001)	(0.003)	(0.002)
	EN	5.42	3.91	**1.58**	1.61	3.9	2.47	32	23,57	16.21	27.69
		(0.43)	(0.15)	(0.21)	(0.19)	(0.16)	(0.15)	(4.06)	(0.35)	(0.63)	(2.04)
	KL	5.68	4.53	**1.7**	1.74	5.07	3.02	7.41	52.61	12.17	6.67
		(0.42)	(0.22)	(0.24)	(0.21)	(0.49)	(0.24)	(0.16)	(1.17)	(0.49)	(0.15)
	NZ	527.2	604.2	**287.8**	289	791.2	518.4	8972.6	9349.6	5932.8	614.22
		(26.73)	(31.6)	(4.68)	(5.31)	(46.64)	(20.09)	(449.65)	(37.82)	(943.41)	(37.28)
	CPUs	124.39	82.6	156.26	30.08	127.65	87.3	598.09	292.6	29.48	**3.25**
		(14.64)	(20.96)	(35.89)	(17.77)	(9)	(4.36)	(7.82)	(2.78)	(0.36)	(0.21)
Moving	RMSE	**0.009**	0.01	0.015	0.012	0.038	0.015	0.025	0.05	0.044	0.021
		(0.0007)	(0.0004)	(0.0008)	(0.0008)	(0.0007)	(0.0009)	(0.0007)	(0.004)	(0.0009)	(0.001)
	EN	**1.02**	1.14	2.05	1.51	7.95	2.29	41.23	30.64	171.97	82.13
		(0.11)	(0.09)	(0.15)	(0.15)	(0.38)	(0.17)	(1.65)	(11.84)	(4.23)	(3.5)
	KL	**1.09**	1.18	2.48	1.66	11.71	2.84	15.21	43.44	37.23	20.65
		(0.19)	(0.15)	(0.22)	(0.21)	(0.82)	(0.25)	(0.3)	(8.2)	(0.89)	(0.63)
	NZ	**298.4**	**298.4**	451.4	361.6	1380	591.8	9986.4	8440.66	4311	640
		(1.2)	(1.2)	(26.3)	(14.41)	(53.4)	(25.16)	(6.24)	(213.4)	(2258.65)	(30.24)
	CPUs	48.91	**6.67**	86.4	27.81	110.86	78.98	679.42	258.07	29.45	8.85
		(5.28)	(1.39)	(6.13)	(4.1)	(6.31)	(6.59)	(6.41)	(27.93)	(0.34)	(0.41)

Conversely, when using the moving model, DCA1-CaP outperforms the others with the loss and the sparsity measures. However, it provides slightly smaller improvement than DCA2-CaP. Here, notably the results obtained with DCA2-CaP and DCA2-PiE are still superior to that of SPCOV1, SPCOV2, CLIME, PCME, SPME, and PDSCE.

Regarding the training time, DCA2-CaP and DCA2-PiE are remarkably faster than the other algorithms in the nonconvex approach keeping the negative log-likelihood function, say DCA1-CaP, DCA1-PiE, SPCOV1 and SPCOV2, as well as the two methods estimating the inverse covariance matrix (CLIME and PCME). This can be explained by the fact that DCA2-CaP and DCA2-PiE lead to the sequences of convex problems which are easily solved by an explicit ADMM algorithm. As for the convex based approaches SPME and PDSCE, not suprisingly, they are faster than the DCA based algorithms but achieve much worse performances on all losses as well as sparsity. The ratio of gain of DCA varies from 1.1 to 168.6 in terms of losses and from 2 to 20.6 in terms of sparsity.

4.3.4 Numerical results on real datasets

We illustrate the use of the sparse covariance matrix estimation problem via two real applications: a classification problem and a portfolio optimization problem of stock data. These applications require an estimate of the covariance matrix.

4.3.4.1 Sparse quadratic discriminant analysis

Let X be an $n \times p$ training data matrix with observations x_i $(i = 1, ..., n)$ on the rows and features on the columns. We assume that the n_i observations within the kth class C_k are normal distributed $\mathcal{N}(\mu_k, \Sigma_k)$. We denote the prior probability of the kth class by π_k. The quadratic discriminant function is

$$\delta_k(x) = -\frac{1}{2}\log\det\Sigma_k - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k. \tag{4.42}$$

Then the predicted class for a new observation x is $\arg\max_k \delta_k(x)$. The decision boundary between each pair of classes k and l is described by a quadratic equation $\{x : \delta_k(x) = \delta_l(x)\}$.

In practice we do not know π_k, μ_k, Σ_k, and will need to estimate them using the training data: $\pi_k = n_k/n, \mu_k = 1/n_k \sum_{x_i \in C_k} x_i$, and a natural way to estimate the Σ_k is via maximum likelihood. Let $S^k = 1/n_k \sum_{x_i \in C_k}(x_i - \mu_k)(x_i - \mu_k)^T$ is the sample covariance matrix for the class k, the negative log likelihood for the data takes the form (up to a constant)

$$\frac{1}{2}\sum_{k=1}^{Q} n_k \left(\log\det\Sigma_k + \mathrm{tr}(\Sigma_k^{-1}S^k)\right). \tag{4.43}$$

Table 4.2: Two datasets from UCI repository used in experiments.

Data	No. of features	No. of samples	No. of classes
Ionosphere	34	351	2
Waveform 2	40	5000	3

We propose to estimate $\Sigma_1, ..., \Sigma_Q$ by minimizing the penalized negative log likelihood

$$\min_{\Sigma_k \succ 0, k=1,...,Q} \sum_{k=1}^{Q} \left(\log \det \Sigma_k + \text{tr}(\Sigma_k^{-1} S^k) + \lambda_k ||\Sigma_k||_0 \right), \tag{4.44}$$

where $\lambda_1, ..., \lambda_Q$ are nonnegative tuning parameters. We refer to this classification method as sparse quadratic discriminant analysis.

Solving the problem (4.44) respect to $\Sigma_1, ..., \Sigma_Q$ can be separated into Q independent subproblems of the same form. This leads to a potentially massive reduction in computational complexity. Recently, Sun and Zhao (2015) has also proposed the sparse quadratic discriminant analysis using a lasso penalty on the entries of the inverse covariance matrices. This work can be viewed as applications of the methods proposed in Rothman (2012).

In our work, we directly estimate the covariance matrices $\Sigma_1, ..., \Sigma_Q$ by using DCA1-CaP, DCA2-CaP, DCA1-PiE and DCA2-PiE. To study the performance of the proposed algorithms, we use SPCOV1, SPCOV2, CLIME, PCME, SPME and PDSCE to estimate $\Sigma_1, ..., \Sigma_Q$. We also compare with the quadratic discriminant analysis (QDA) replacing Σ_k in (4.42) by the sample covariance matrix S^k. Note that if S^k is singular, then we replace it by $S^k + \epsilon I_p$, where ϵ is chosen through a 5-fold cross-validation on a set of candidates $\mathcal{E} = \{10^{-4}, ..., 10^{-8}\}$.

For the experiment, we evaluate the proposed algorithms on three datasets: US Postal Service (USPS) dataset and two datasets from UCI Machine Learning Repository (Ionosphere and Waveform 2).

The US Postal Service task is still one of the most widely used reference dataset for handwritten digit recognition. Here we focus on a difficult sub-task, that of distinguishing handwritten 3s and 8s. All images are of size 16×16 pixels. There are 658 threes and 542 eights in the training set, and 166 test samples for each. In our experiment we name this data as 3s and 8s dataset. It can be downloaded at http://statweb.stanford.edu/~tibs/ElemStatLearn/data.html. A random selection is showed in Figure 4.1. We filter the data by replacing each non-overlapping 2×2 pixel block by its average. This reduces the dimension of the feature space from 256 to 64.

The detailed information of Johns Hopkins University Ionosphere and Waveform 2 datasets is summarized in Table 4.2. We use the cross-validation scheme to validate the performance of various approaches on these two datasets. The dataset is split into a training set containing 2/3 of the samples and a test set containing 1/3 of the samples. This process is repeated 10 times, each with a random choice of training set and test set. The parameter $\lambda_1, ..., \lambda_Q$ are chosen via 5-fold cross-validation.

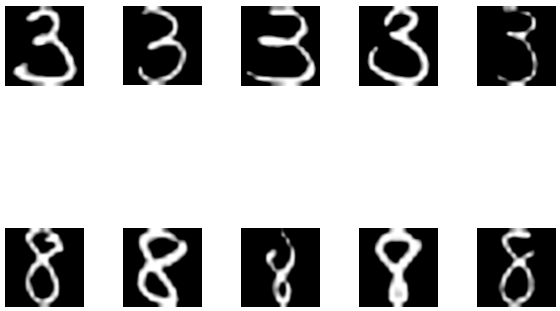

Figure 4.1: Examples of digitized handwritten 3s and 8s. Each image is a 8 bit, 16×16 grayscale version of the original binary image.

Table 4.3: Digit classification results of 3s and 8s. Bold fonts indicate the best result in each column.

	Testing error (%)	Training error (%)	Training time (s)
DCA1-CaP	4.81	2.08	129.53
DCA2-CaP	**2.71**	**1.91**	34.97
DCA1-PiE	4.51	3.08	113.13
DCA2-PiE	3.01	2.16	39.5
SPCOV1	4.81	2	110.43
SPCOV2	4.51	3.16	87.39
QDA	6.32	2.16	-
CLIME	4.01	2.83	218.48
PCME	4.31	2.25	64.18
SPME	5.83	4.25	22.3
PDSCE	5.32	3.33	**7.45**

Table 4.4: Comparative results of Ionosphere and Waveform 2 datasets in terms of the average of percentage of testing errors, training errors, training time in second and their standard deviations over 10 training/test set splits. The bold font indicates the best result in each column.

		Testing error (%)	Training error (%)	Training time (s)
Ionosphere	DCA1-CaP	6.58 ± 2.16	7.82 ± 1.5	18.95 ± 7.57
	DCA2-CaP	**4.52 ± 1.66**	6.23 ± 0.94	**1.17 ± 0.15**
	DCA1-PiE	6.75 ± 1.68	8.47 ± 1.49	15.08 ± 4.41
	DCA2-PiE	**4.52 ± 1.62**	6.58 ± 1.22	1.22 ± 0.23
	SPCOV1	6.41 ± 2.03	5.89 ± 0.87	21.44 ± 16.21
	SPCOV2	-	-	-
	QDA	11.62 ± 3.38	**3.33 ± 0.49**	-
	CLIME	10.96 ± 2.04	9.64 ± 1.19	33.96 ± 0.31
	PCME	11.25 ± 1.37	10.68 ± 0.43	19.6 ± 2.38
	SPME	7.17 ± 2.15	5.51 ± 0.68	8.62 ± 0.39
	PDSCE	10.38 ± 2.82	11.34 ± 1.91	3.97 ± 0.13
Waveform 2	DCA1-CaP	14.33 ± 0.61	11.26 ± 0.68	12.3 ± 0.63
	DCA2-CaP	14.31 ± 0.36	10.87 ± 0.54	1.38 ± 0.87
	DCA1-PiE	**14.27 ± 0.59**	11.4 ± 0.69	13.5 ± 0.97
	DCA2-PiE	14.37 ± 0.52	11.73 ± 0.67	**1.37 ± 0.57**
	SPCOV1	14.91 ± 0.71	10.26 ± 0.24	9.75 ± 0.13
	SPCOV2	14.84 ± 0.67	10.28 ± 0.19	11.31 ± 0.35
	QDA	16.27 ± 0.57	**9.09 ± 0.16**	-
	CLIME	15.42 ± 0.84	10.45 ± 0.43	77.38 ± 0.36
	PCME	15.16 ± 0.73	9.79 ± 0.12	16.29 ± 0.78
	SPME	15.57 ± 0.65	14.68 ± 0.48	14.32 ± 0.73
	PDSCE	15.01 ± 0.93	10.89 ± 1.22	3.71 ± 0.11

The computational results are reported in Tables 4.3-4.4. In Table 4.3, the classification results and the training time in second for 3s and 8s dataset are given, and we notice that the testing and training errors obtained with DCA2-CaP are the lowest.

In Table 4 we reported the average percentage of the testing and training errors, the training time in seconds as well as the standard deviations of Ionosphere and Waveform 2 datasets over 10 training/test set splits. We observe that, on the Ionosphere dataset, DCA2-CaP and DCA2-PiE outperform DCA1-CaP, DCA1-PiE, SPCOV1, QDA, CLIME, PCME, SPME and PDSCE in terms of the testing error. Notably in this dataset, SPCOV2 is not able to perform since there exists some zero elements in the sample covariance matrices. In the Waveform 2, the DCA based algorithms give slightly better testing error than the other approaches, and DCA1-PiE has the smallest testing error. However, QDA gives the best training error on both Ionosphere and Waveform 2 datasets. As for the training time, DCA2-CaP is the fastest on the Ionosphere dataset while DCA2-PiE is the best on the Waveform 2 dataset, and we further note that both of them are significantly faster than the other approaches on these two datasets. On the 3s and 8s dataset, DCA2-CaP and DCA2-PiE run slower than PDSCE but faster than the seven remaining approaches.

4.3.4.2 Portfolio optimization

In this section, we apply the sparse covariance matrix estimation problem on an application of portfolio optimization. The celebrated Markowitz portfolio selection problem (Markowitz, 1952) in finance is to construct the optimal mean-variance efficient portfolios by minimizing the following quadratic optimization problem:

$$\min_{w\in\mathbb{R}^p} \left\{ w^T\Sigma w : w^T\mu = \mu_*; w^T e = 1; w \geq 0 \right\}, \tag{4.45}$$

where e denotes the p-dimensional vector of ones and μ_* is the expected rate of return that is required on the portfolio. When short selling is allowed, the constraint $w \geq 0$ in the problem (4.45) can be removed.

In the recent literature, many works aim to find an estimate of the covariance matrix to improve its portfolio strategy (see e.g. Ledoit and Wolf (2003, 2004); Jagannathan and Ma (2003); Kourtis et al. (2012); Fan et al. (2013); Xue et al. (2012); Lai et al. (2011); Deng and Tsui (2013)). We follow Jagannathan and Ma (2003); Kourtis et al. (2012); Fan et al. (2013); Xue et al. (2012); Deng and Tsui (2013) to focus on the global minimum variance portfolio is the minimum risk portfolio with weights that sum to unity, namely

$$\min_{w\in\mathbb{R}^p} \left\{ w^T\Sigma w : w^T e = 1 \right\}. \tag{4.46}$$

We expect that an accurate covariance matrix estimate will lead to a better portfolio strategy. We use eleven different approaches to estimate Σ and to obtain w: our approaches (DCA1-CaP, DCA2-CaP, DCA1-PiE, DCA2-PiE), SPCOV1, SPCOV2, CLIME, PCME, SPME, PDSCE and the sample covariance matrix S.

We consider a stock dataset used in Deng and Tsui (2013) and it is also available from Yahoo!Finance (`http://finance.yahoo.com`). This dataset is the weekly returns of 30 components of the Dow Jones Industrial Index. The dataset of adjusted close prices of the weekly returns were extracted in the past three and a half years from January 8, 2007 to June 28, 2010. The dataset is divided in the same way as in Deng and Tsui (2013). The first 50 observations of the weekly returns data is the training set, the next 50 observations is the tuning set, and the remaining data is the test set. The performance of a portfolio w is measured by the realized return

$$R(w) = \sum_{x\in X_{ts}} w^T x, \tag{4.47}$$

the realized risk

$$\sigma(w) = \sqrt{w^T S_{ts} w}, \tag{4.48}$$

and the Sharpe ratio

$$S(w) = \frac{R(w)}{\sigma(w)}, \tag{4.49}$$

where X_{ts} is the test set and S_{ts} is the sample covariance matrix of X_{ts}.

Table 4.5: The comparison of the realized return, realized risk and Sharpe ratio. Bold fonts indicate the best result in each column.

	$R(w)$	$\sigma(w)$	Sharpe ratio	Training time (s)
DCA1-CaP	0.2398	0.0303	7.9038	14.52
DCA2-CaP	0.2429	**0.028**	8.6613	0.33
DCA1-PiE	0.2441	0.0309	7.8788	4.51
DCA2-PiE	**0.27**	0.0303	**8.886**	**0.22**
SPCOV1	0.2398	0.0303	7.9038	14.57
SPCOV2	0.2567	0.0314	8.1566	4.66
S	0.0593	0.0348	1.7045	-
CLIME	0.1121	0.0237	4.73	13.77
PCME	0.2172	0.0307	7.07	0.65
SPME	0.2272	0.0302	7.51	1.09
PDSCE	0.1258	0.0343	3.67	0.61

In Table 5, the realized return, the realized risk and the Sharpe ratio of the eleven comparative methods DCA1-CaP, DCA2-CaP, DCA1-PiE, DCA2-PiE, SPCOV1, SPCOV2, CLIME, PCME, SPME, PDSCE and S are presented. These results indicate that DCA2-PiE is the best one. Although the realized risk produced by this approach is a little bit greater than DCA2-CaP, its realized return achieves the highest value. Hence, it follows that the largest Sharpe ratio is achieved by DCA2-PiE. We also notice that DCA2-PiE and DCA2-CaP are respectively the fastest and the second fastest in training time.

The next experiment, we evaluate the performance of portfolios in different periods. This experiment is the same as Deng and Tsui (2013) which is reviewed as follows: the first 50 observations of the weekly returns is the training set, the next 50 observations is the tuning set, and the third 50 observations of the weekly returns is the test set. By changing the starting week during the period from January 8, 2007 to August 20, 2007, we have the 33 different consecutive test periods. The realized return, the realized risk and the Sharpe ratio are calculated for each test period using the optimal portfolio w based on the corresponding training set. The computational results are shown in Figures 4.2-4.4.

The results in Figures 4.2 and 4.3 show that the portfolios created by DCA2-CaP and DCA2-PiE have higher realized returns than the other approaches and these two methods also provide comparable realized risks with the others in almost all periods. Moreover, the results in Figure 4.4 show that the Sharpe ratios of the portfolios produced by DCA2-CaP and DCA2-PiE are larger than SPCOV1, SPCOV2, CLIME, PCME, SPME, PDSCE and S. Thus, it is convincing to conclude that DCA2-CaP and DCA2-PiE result in better portfolio strategies.

4.4 Conclusion

In this chapter, we have investigated DC approximation approaches and DCA for solving the sparse covariance matrix estimation problem using the ℓ_0-norm. We propose two DC formulations for the approximation SCME problem and develop four DCA based

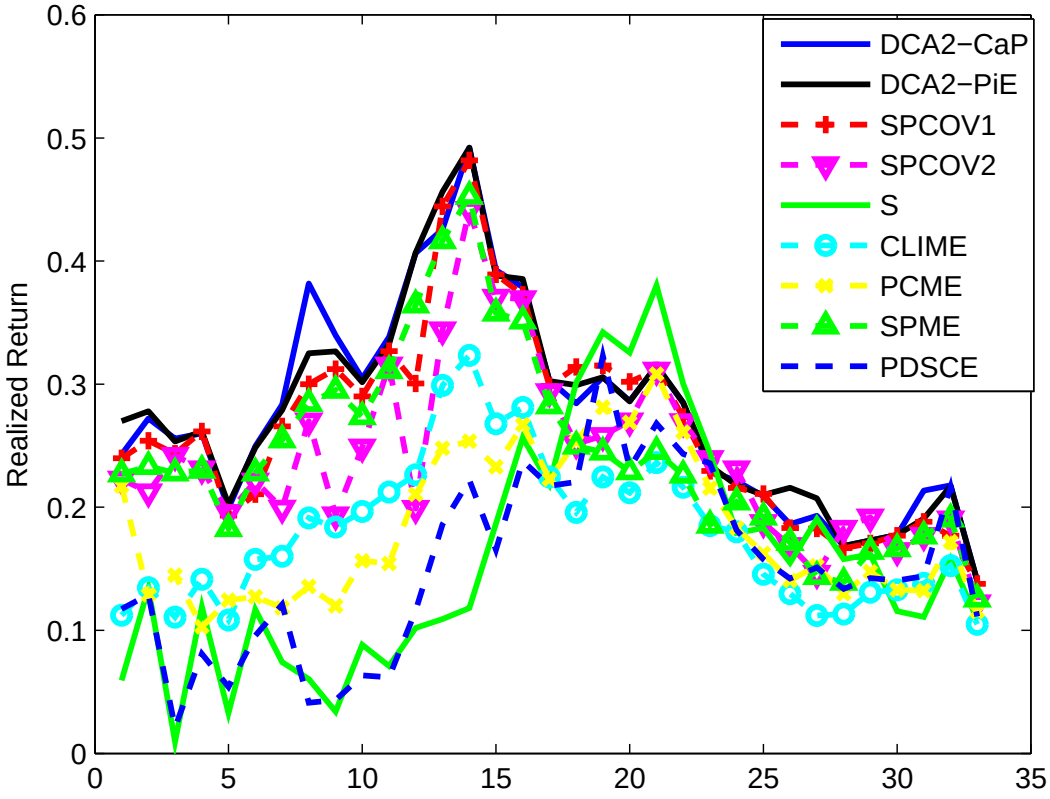

Figure 4.2: The comparison of the realized return in different test periods.

algorithms, by using two appropriate DC approximation functions of the ℓ_0-norm. The robustness and the effectiveness of our DCA based algorithms have been demonstrated through the computational results on both simulated and real datasets. The nice effect of DC decomposition has been exploited: the second DC decomposition seems to be very suitable since it leads to an efficient, fast and scalable DCA scheme. In the experimental results, DCA2-CaP and DCA2-PiE have obtained the best performance in terms of most of comparison criteria, and have taken the shortest time for training. The utility of our approaches have been illustrated via two important applications: classification by a quadratic discriminant function and portfolio optimization. Their superiority on seven state-of-the-art algorithms are proved via various numerical experiments.

We are convinced that the approaches developed in this chapter bring to researchers and practitioners new and efficient methods to treat an important and difficult problem that can be used to many applications in various domains.

As a part of future work, we plan to study more extensive applications of the sparse covariance matrix estimation problem.

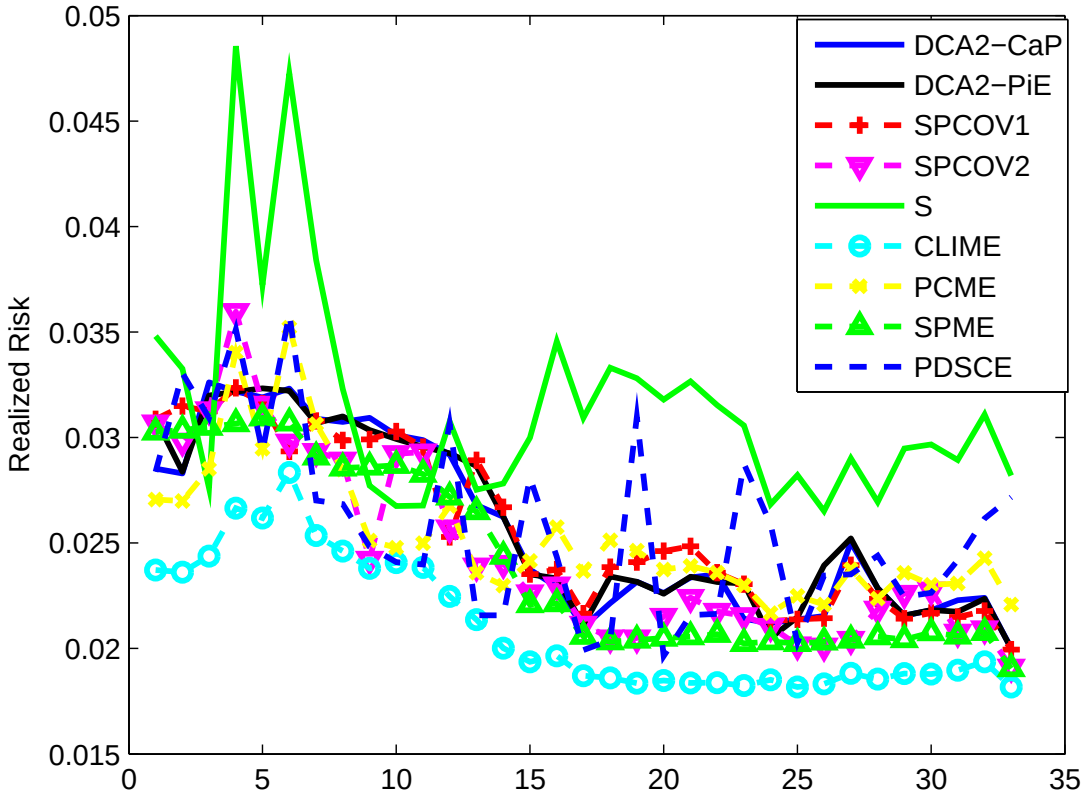

Figure 4.3: The comparison of the realized risk in different test periods.

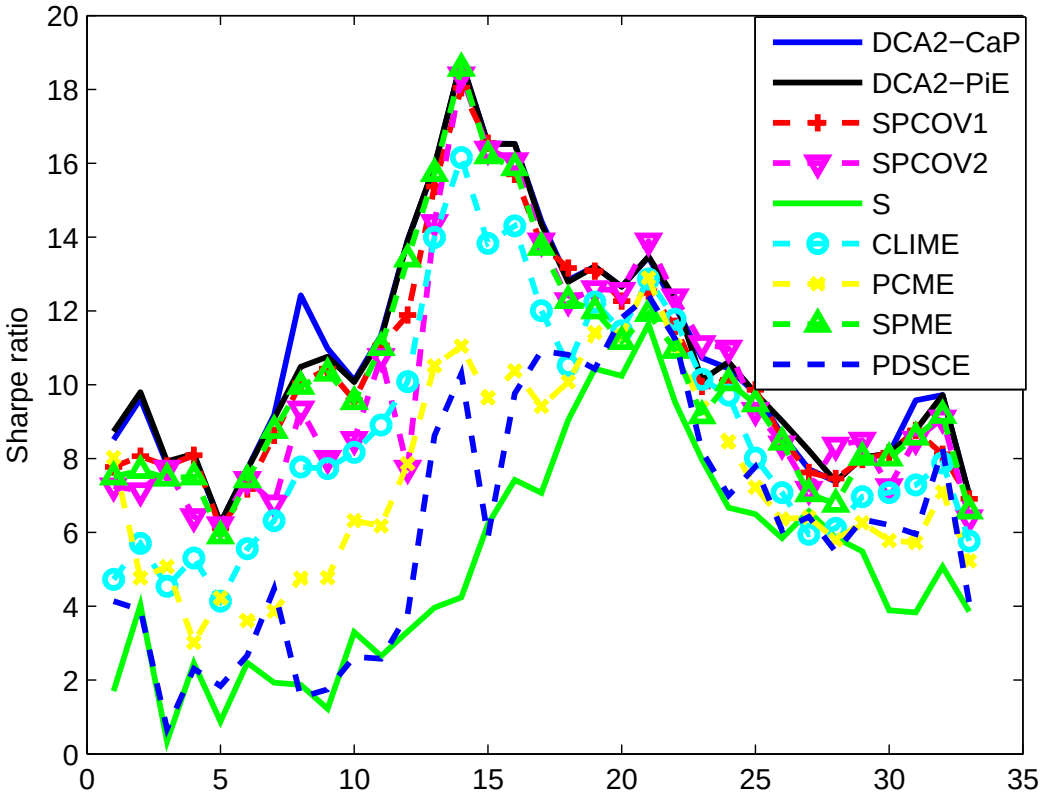

Figure 4.4: The comparison of the Sharpe ratio in different test periods.

Part II

Group Variable Selection and Classification

Chapter 5

Group Variable Selection: Applications to Optimal Scoring and Estimation of Multiple Covariance Matrices

Abstract: *The need to select groups of variables arises in many statistical modeling problems and applications. In this chapter, we introduce a new regularization using the $\ell_{p,0}$-norm for enforcing group sparsity. Using a DC (Difference of Convex functions) approximation of the $\ell_{p,0}$-norm, we show that the approximate problem is equivalent to the original problem with suitable parameters. Considering two equivalent formulations of the approximate problem we develop DC programming and DCA (DC Algorithm) for solving them. When $p = 1$ (resp. $p = 2$), our algorithms include ℓ_1-perturbed algorithm (resp. $\ell_{2,1}$-perturbed algorithm) and reweighted-ℓ_1 algorithm (resp. reweighted-$\ell_{2,1}$ algorithm). It turns out that, among $\ell_{p,0}$ regularizations, the $\ell_{1,0}$ is the most interesting regularization with several advantages in both theoretical and computational aspects. As applications, we implement the proposed algorithms for group variable selection in optimal scoring problem and estimation of multiple covariance matrices. In the first application, 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. In the second application multiple covariance matrices sharing some common structures such as the locations or weights of non-zero elements, we combine the ℓ_0-norm and the $\ell_{p,0}$-norm for enforcing sparsity on each covariance matrix and across multiple covariance matrices, respectively. The experimental results on both simulated and real datasets demonstrate the efficiency of the proposed algorithms.*

1. The material of this chapter is based on the following work:
[1] Hoai An Le Thi and Duy Nhat Phan. Efficient Nonconvex Group Variable Selection and Application to Group Sparse Optimal Scoring. Submitted.

5.1 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 chapter, 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 $\ell_{p,0}$-regularized problem takes the form:

$$\min \left\{ f(x, y) + \lambda \|x\|_{p,0} : (x, y) \in K \subset \mathbb{R}^d \times \mathbb{R}^m \right\}, \tag{5.1}$$

where λ is a nonnegative tuning parameter.

Let us mention some important applications of group variable selection corresponding to the model (5.1).

Group variable selection in linear regression: Given n observations (x_i, y_i), $i = 1, ..., n$, where y_i is the response variable, $x_i = (x_{i1}', ..., x_{iJ}')'$ 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 (5.1):

$$\min \left\{ \frac{1}{2n} \|y - \sum_{j=1}^{J} X_j \beta^j\|_2^2 + \lambda \sum_{j=1}^{J} s(\|\beta^j\|_p) \right\}. \tag{5.2}$$

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 (5.1):

$$\min \left\{ \sum_{j=1}^{T} \sum_{i=1}^{n_j} \mathcal{L}(y_i^j, w_j^T x_i^j + b_j) + \lambda |||W||_{p,0} \right\}, \tag{5.3}$$

where $\mathcal{L}$ denotes the loss function, $W = [w_1, ..., w_T] \in \mathbb{R}^{d \times T}$ 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 is 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 SVMs is a special case of the problem (5.3) where $\mathcal{L}$ is the hinge loss function given by

$$\mathcal{L}(y_i^j, w_j^T x_i^j + b_j) = \sum_{l=1, l \neq j}^{T} \max\left(1 - (w_j^T x_i^j - w_l^T x_i^j + b_j - b_l), 0\right).$$

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.

$$\min_{A \in \mathbb{R}^{p \times k}} \left\{ \|X - XAA^T\|_F^2 : A^T A = I_k \right\}, \tag{5.4}$$

where the columns of A which minimize (5.4) 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.

$$\min_{A,B \in \mathbb{R}^{d \times k}} \left\{ \|X - XBA^T\|_F^2 + \lambda \|B\|_{p,0} : A^T A = I_k \right\}, \tag{5.5}$$

where the columns $\beta_1, ..., \beta_k$ of B correspond to the required 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 a L-dimensional space ($L \leq C - 1$), in which the between-class variance is maximized while the within-class variance is minimized, i.e.,

$$\max_{W \in \mathbb{R}^{d \times L}} \left\{ \text{tr}\left((W^T \Sigma_w W)^{-1} (W^T \Sigma_b W) \right) \right\}, \tag{5.6}$$

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 regularized problem takes the form of (5.1):

$$\min_{W \in \mathbb{R}^{d \times L}} \left\{ -\text{tr}\left((W^T \Sigma_w W)^{-1} (W^T \Sigma_b W) \right) + \lambda \|W\|_{p,0} \right\}. \tag{5.7}$$

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 and 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 group sparse optimization problem of the form (5.1):

$$\min_{W} \left\{ \|AW - B\|_F^2 + \lambda \|W\|_{p,0} \right\}. \tag{5.8}$$

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 and Lin, 2006), is closely connected to the Lasso (ℓ_1-norm) approximation of the ℓ_0-norm. Works in this direction include $\ell_{\infty,1}$-norm (Liu et al., 2009a; Quattoni et al., 2009; Zhang et al., 2010) and the $\ell_{2,1}$-norm which was widely used for group variable selection in multi-task learning (Argyriou et al., 2008; Bi et al., 2008; Liu et al., 2009b; Obozinski et al., 2006, 2010; Zhang et al., 2010; Nie et al., 2010; Lan et al., 2015), multiclass SVMs (Blondel 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, they may be not selection consistent and tend to select non important groups in the model. The second approach deals with nonconvex approximation and has been developed for the $\ell_{2,0}$-norm. More precisely, DC (difference of convex functions) approximation approaches based on the smoothly clipped absolute deviation penalty (SCAD) (Fan and 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 et al., 2012; Wang et al., 2007; Wei and Zhu, 2012)). These works have proved that these DC approximation approaches for the $\ell_{2,0}$-norm are more efficient than the methods using the $\ell_{2,1}$-norm.

In this chapter, we investigate DC approximation approaches for the general case, i.e. the $\ell_{p,0}$-norm with $p \geq 1$. We consider the problem (5.1), 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 chapter makes the following contributions.

Firstly, basing on the piecewise linear function (called Capped-ℓ_1) introduced in Peleg and Meir (2008), we approximate the $\ell_{p,0}$-norm by a DC function. We prove that, with suitable parameters, the nonconvex approximate problem is equivalent to the original problem (5.1). This result gives an important mathematical foundation for our approximation method.

Secondly, we develop solution methods based on DC programming and DCA (DC Algorithms), a powerful technique in nonconvex optimization (Le Thi and Pham Dinh, 2005; Pham Dinh and Le Thi, 1997), for solving the approximate problem. Considering two equivalent formulations of the approximate problem we propose two DCA. These two DCA schemes 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.

Among $\ell_{p,0}$-regularizations, we show that the $\ell_{1,0}$-regularization is the most interesting with several useful properties from both theoretical and computational aspects. 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 in many applications. This interesting feature makes our proposed approach very efficient in terms of computational complexity.

Finally, as applications, we consider the problem of group variable selection in optimal scoring and estimation of multiple covariance matrices. We also perform a careful empirical experiment to study the performance of the proposed approaches.

The rest of the chapter is organized as follows. In Section 5.2, we present the approximate problems and show that these problems are equivalent to the original problem. We illustrate how to apply DCA to solve the approximate problems in Section 5.3. The application of the proposed algorithms for group variable selection in optimal scoring is described in Section 5.4 while the application of the proposed algorithms for group variable selection in estimation of multiple covariance matrices is described in Section 5.5. Section 5.6 concludes the chapter.

Throughout the chapter, 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 $||A||_F = \sqrt{\sum_{i,j} A_{ij}^2}$.

5.2 DC approximate problems and the link with the original problem

We consider the approximate problem of (5.1) which takes the form

$$\min \left\{ F_\alpha^p(x,y) := f(x,y) + \lambda \sum_{j=1}^{J} \eta_\alpha(\|x^j\|_p) : (x,y) \in K \right\}, \tag{5.9}$$

where $\eta_\alpha(t) = \min\{1, \alpha|t|\}$ is the Capped-ℓ_1 function (Peleg and Meir, 2008), 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

$$\eta_\alpha(\|x^j\|_\infty) \leq \eta_\alpha(\|x^j\|_p) \leq \eta_\alpha(\|x^j\|_1) \leq s(\|x^j\|_p). \tag{5.10}$$

This shows that, with the same parameter α, $\eta_\alpha(\|.\|_1)$ is the closest to the step function s. We also consider another equivalent form of the problem (5.9) as follows:

$$\min \left\{ F_\alpha(x,y,z) := f(x,y) + \lambda \sum_{j=1}^{J} \eta_\alpha(z_j) : (x,y,z) \in K_p \right\}, \tag{5.11}$$

where $K_p = \{(x,y,z) : (x,y) \in K, \|x^j\|_p \leq z_j, j = 1, ..., J\}$. Indeed, the problems (5.9) and (5.11) are equivalent in the following sense.

Proposition 5.1 *A point $(x^*, y^*) \in K$ is a global (resp. local) solution of the problem (5.9) if and only if $(x^*, y^*, |x_J^*|_p)$ is a global (resp. local) solution to the problem (5.11). Moreover, if (x^*, y^*, z^*) is a global solution to (5.11) then (x^*, y^*) is a global solution to (5.9).*

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

$$|\|x^j\|_p - \|(x^*)^j\|_p| \leq \|x^j - (x^*)^j\|_p \leq d_j^{1/p-1/2} \|x^j - (x^*)^j\|_2. \tag{5.12}$$

Combining $d_j^{1/p-1/2} \leq a$ and (5.12), we obtain

$$|\|x^j\|_p - \|(x^*)^j\|_p| \leq a\|x^j - (x^*)^j\|_2 \ \forall j = 1, ..., J. \tag{5.13}$$

If $p \geq 2$, we have

$$|\|x^j\|_p - \|(x^*)^j\|_p| \leq \|x^j - (x^*)^j\|_p \leq \|x^j - (x^*)^j\|_2. \tag{5.14}$$

From (5.13)-(5.14) 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 the complete. □

In this section, we introduce the following assumption:

Assumption 1 *K is a compact polyhedral convex set. By Proposition 5.1, without the generality, we can assume that K_p is a compact convex set.*

Proposition 5.2 *Under the assumption (1), with $p = +\infty$. Then, there exists α_0 such that the approximate problem (5.9) is equivalent to the original problem (5.1) 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 (5.11) is an approximate problem of the following problem including the ℓ_0 penalty on the vector z.

$$\min \left\{ f(x, y) + \lambda \sum_{j=1}^{J} s(z_j) : (x, y, z) \in K_\infty \right\}. \tag{5.15}$$

Following the conclusion after Proposition 3 in Le Thi et al. (2015), there exists α_0 such that the problem (5.11) and (5.15) 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 problem (5.15) is equivalent to the problem (5.1). By Proposition 5.1, we have the equivalence between the problem (5.11) and the problem (5.9). Hence, the approximate problem (5.9) is equivalent to the original problem (5.1). Thus, Proposition 5.2 is proved. □

For $p = +\infty$, we have proved that the approximate problem (5.9) is equivalent to the problem (5.1) for all $\alpha > \alpha_0$. We now have the general result for all $p \geq 1$.

Proposition 5.3 *Under the assumption (1), there exists α_0 such that the approximate problem (5.9) is equivalent to the original problem (5.1) for all $\alpha > \alpha_0$ and $p \geq 1$.*

Proof : From the inequality (5.10), we have

$$\eta_\alpha(\|x^j\|_\infty) \leq \eta_\alpha(\|x^j\|_p) \leq s(\|x^j\|_p). \tag{5.16}$$

Hence, we obtain

$$F_\alpha^\infty(x,y) \leq F_\alpha^p(x,y) \leq f(x,y) + \lambda \sum_{j=1}^{J} s(\|x^j\|_\infty) \ \forall (x,y) \in K. \tag{5.17}$$

For $p = +\infty$, by Proposition 5.2, the problems (5.9) and (5.1) 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 with (5.17), 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 (5.9) is equivalent to the original problem (5.1). This completes the proof of the Proposition 5.3. □

We are now going to develop DCA based algorithms for solving the approximate problems (5.9)-(5.11).

5.3 Solution methods via DC programming and DCA

5.3.1 DCA for solving the first approximate problem

Firstly, we consider the approximate problem (5.9) and introduce a DCA scheme that includes algorithms of ℓ_1-perturbed/$\ell_{2,1}$-perturbed types. The function $\eta_\alpha(t)$ can be expressed as a DC function:

$$\eta_\alpha(t) = \alpha|t| - r(t), \tag{5.18}$$

where $r(t) = -1 + \max\{1, \alpha|t|\}$. Hence, the objective function of the problem (5.9) can be rewritten as a DC function:

$$F_\alpha^p(x, y) = G_1(x, y) - H_1(x, y), \tag{5.19}$$

where

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

and g, h are DC components of f, i.e., $f = g - h$. Hence a DC formulation of the problem (5.9) takes the form

$$\min_{(x,y)}\{G_1(x, y) - H_1(x, y)\}. \tag{5.20}$$

Following the generic DCA scheme, DCA for solving the problem (5.20) can be described as follows.

DCA1

Initialization: Choose $(x^0, y^0) \in K$, $l \leftarrow 0$ and let τ be a tolerance sufficient small.
repeat

1. Compute $(\bar{x}^l, \bar{y}^l) \in \partial h(x^l, y^l)$ and $(v^l)^j \in \lambda\partial r(\|(x^l)^j\|_p)$, $j = 1, ..., J$.
2. Compute (x^{l+1}, y^{l+1}) by solving the problem:

$$\min_{(x,y)\in K} \left\{g(x, y) - \langle \bar{x}^l, x\rangle - \langle \bar{y}^l, y\rangle + \lambda\alpha\|x\|_{p,1} - \langle v^l, x\rangle\right\}. \tag{5.21}$$

3. $l \leftarrow l + 1$.

until $\|(x^l, y^l) - (x^{l-1}, y^{l-1})\|_2 \leq \tau(\|(x^{l-1}, y^{l-1})\|_2 + 1)$ or $|F_\alpha^p(x^l, y^l) - F_\alpha^p(x^{l-1}, y^{l-1})| \leq \tau(|F_\alpha^p(x^{l-1}, y^{l-1})| + 1)$

Remark 5.1 *We see that the problem (5.21) has the form of an $\ell_{p,1}$-perturbed problem. Thus, for $p = 2$, (5.21) is an $\ell_{2,1}$-perturbed problem. For $p = 1$, we have $\|x\|_{1,1} \equiv \|x\|_1$ and the problem (5.21) can be rewritten as follows.*

$$\min_{(x,y)\in K} \left\{g(x, y) - \langle \bar{x}^l, x\rangle - \langle \bar{y}^l, y\rangle + \lambda\alpha\|x\|_1 - \langle v^l, x\rangle\right\}. \tag{5.22}$$

This problem has the form of an ℓ_1-perturbed problem which can be found in many previous works (see e.g. (Le Thi et al., 2008, 2014a, 2015; Ong and Le Thi, 2013b)). Thanks to the ℓ_1-norm, if $g(x, y)$ is separable in its variables, so is the problem (5.22). This leads to a potential massive reduction in computational complexity. 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 can be separated in columns. Hence, the problem (5.22) can be separated into independent sub-problems in these applications. Note, however, that the problem (5.21) is not separable if $p \neq 1$. Thus we can say that the $\ell_{1,0}$ is the most interesting regularization for DCA.

5.3.2 DCA for solving the second approximate problem

In the following, we introduce a DCA scheme for solving the problem (5.11) 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 (5.11) is a DC program of the form:

$$\min_{(x,y,z)} \{G_2(x,y,z) - H_2(x,y,z)\}, \tag{5.23}$$

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 (5.23) 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

$$(x^{l+1}, y^{l+1}, z^{l+1}) \in \arg\min_{(x,y,z)\in K_p} \left\{ g(x,y) - \langle \bar{x}^l, x\rangle - \langle \bar{y}^l, y\rangle - \langle v^l, z\rangle \right\}. \tag{5.24}$$

Since $v_j^l \leq 0$ $\forall j = 1, ..., J$, the (5.24) is equivalent to

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

Hence, DCA for solving the problem (5.11) can be described as follows.

DCA2

Initialization: Choose $(x^0, y^0) \in K$, $l \leftarrow 0$ and let τ be a tolerance sufficient small.
repeat

1. Compute $(\bar{x}^l, \bar{y}^l) \in \partial h(x^l, y^l)$ and $v_j^l \in \lambda \partial(-\eta_\alpha)(\|(x^l)^j\|_p)$ $\forall j = 1, ..., J$.

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

$$\min_{(x,y)\in K} \left\{ g(x,y) - \langle \bar{x}^l, x\rangle - \langle \bar{y}^l, y\rangle + \sum_{j=1}^{J} (-v_j^l)\|x^j\|_p \right\}. \tag{5.25}$$

3. $l \leftarrow l+1$.

until $\|(x^l, y^l, |x_J^l|_p) - (x^{l-1}, y^{l-1}, |x_J^{l-1}|_p)\|_2 \leq \tau(\|(x^{l-1}, y^{l-1}, |x_J^{l-1}|_p)\|_2 + 1)$ or $|F_\alpha(x^l, y^l, |x_J^l|_p) - F_\alpha(x^{l-1}, y^{l-1}, |x_J^{l-1}|_p)| \leq \tau(|F_\alpha(x^{l-1}, y^{l-1}, |x_J^{l-1}|_p)| + 1)$

Remark 5.2 *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*

$$\min_{(x,y)\in K} \left\{ f(x,y) + \sum_{j=1}^{J} (-v_j^l)\|x^j\|_p \right\}. \tag{5.26}$$

We see that (5.26) 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 (5.26) with weights $(-v_j^l)$ being updated at each iteration l.

For $p = 1$, (5.26) becomes a group-weighted-ℓ_1 problem. Moreover, if $f(x, y)$ is separable, problem (5.26) can be separated into independent sub-problems. This is the case in many applications.

For $p = 2$, (5.26) 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 and 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.

5.4 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 (5.6), 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 $X = [x_1; ...; x_n] \in \mathbb{R}^{n\times d}$. 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 $W = [w_1, ..., w_L] \in \mathbb{R}^{d\times L}$, that maps the data in the d-dimensional space to an 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

$$\begin{aligned} \min_{\Theta\in\mathbb{R}^{C\times L}, W\in\mathbb{R}^{d\times L}} \quad & \left\{\frac{1}{2n}||Y\Theta - XW||_F^2\right\} \\ \text{subject to} \quad & \frac{1}{n}\Theta^T Y^T Y\Theta = I_L, \end{aligned} \tag{5.27}$$

where I_L is the $L \times L$ identity matrix. Hastie et al. (1994) proposed an algorithm for solving the problem (5.27) 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

$$\min_{W\in\mathbb{R}^{d\times L}} \left\{\frac{1}{2n}||Y\Theta_0 - XW||_F^2\right\}. \tag{5.28}$$

(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.

In the optimal scoring problem, a feature j-th is selected if at least a component in the row j-th of W is nonzero and vice versa. Therefore, it is reasonable to consider rows of W as groups. Denote by w^j ($j = 1, \ldots, d$) the row j-th of W. In the step 2, using $\ell_{p,0}$-regularization for the problem (5.28) leads us to consider the group sparse optimal scoring problem.

$$\min_{W\in\mathbb{R}^{d\times L}} \left\{\frac{1}{2n}||Y\Theta_0 - XW||_F^2 + \lambda\|W\|_{p,0}\right\}, \tag{5.29}$$

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.

Observe that the problem (5.29) is a special case of (5.1) 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:

$$\min_{W\in\mathbb{R}^{d\times L}} \left\{ F^p(W) := \frac{1}{2n}||Y\Theta_0 - XW||_F^2 + \lambda \sum_{j=1}^{d} \eta_\alpha(\|w^j\|_p) \right\}. \tag{5.30}$$

By Proposition 5.1, this problem is equivalent to

$$\min_{(W,z)\in K_p} \left\{ F(W,z) := \frac{1}{2n}||Y\Theta_0 - XW||_F^2 + \lambda \sum_{j=1}^{d} \eta_\alpha(z_j) \right\}, \tag{5.31}$$

where $K_p = \{(W,z) : \|w^j\|_p \le z_j, j = 1,\dots,d\}$.

In the appendix A.1, we will prove that the optimal solution set of the problem (5.29) is bounded. Hence, without loss of generality, we can only consider the problem (5.29) on a box $K = [-M, M]^{d\times L}$ for sufficient large M. By Proposition 5.3, there exists α_0 such that the approximate problem (5.30) is equivalent to the original problem (5.29) for all $\alpha > \alpha_0$.

Here f is a convex quadratic function, and DC components of f are taken as $g = f$ and $h = 0$. According to DCA1 and DCA2 with $p = 1, 2$, we have four DCA based algorithms, $\ell_{2,0}$(DCA1), $\ell_{2,0}$(DCA2), $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2), described as follows.

$\ell_{1,0}$(DCA1) (DCA1 with $p = 1$ for solving (5.30))

Initialization: Choose $W^0 \in K$, $l \leftarrow 0$ and let τ be a tolerance sufficient small.
repeat

1. Set $(v^l)^j \in \lambda\partial r(\|(w^l)^j\|_1)$, $j = 1, ..., d$ as

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

2. Compute W^{l+1} by solving the problem:

$$\min_{W\in K} \left\{ \frac{1}{2n}||Y\Theta_0 - XW||_F^2 + \lambda\alpha\|W\|_{1,1} - \sum_{j=1}^{d} \langle (v^l)^j, w^j \rangle \right\}. \tag{5.32}$$

3. $l \leftarrow l + 1$.

until $\|W^l - W^{l-1}\|_F \le \tau(\|W^{l-1}\|_F + 1)$ or $|F^1(W^l) - F^1(W^{l-1})| \le \tau(|F^1(W^{l-1})| + 1)$

The problem (5.32) can be separated into L independent sub-problems of the same form:

$$\min_{w_k \in [-M,M]^d} \left\{ \frac{1}{2n} ||(Y\Theta_0)_k - Xw_k||_2^2 + \lambda\alpha \|w_k\|_1 - \langle v_k^l, w_k \rangle \right\}, \tag{5.33}$$

where $(Y\Theta_0)_k$ denotes the k-th column of $Y\Theta_0$ and $v_k^l = ((v^l)_k^1, ..., (v^l)_k^d)$. Hence, for solving the problem (5.32), we can solve L sub-problems in parallel. This shows the efficiency of the $\ell_{1,0}$-regularization. For solving problem (5.33), we use the coordinate descent method (Friedman et al., 2007).

$\ell_{1,0}$**(DCA2)** (DCA2 with $p = 1$ for solving (5.31))

Initialization: Choose $W^0 \in K$, $l \leftarrow 0$ and let τ be a tolerance sufficient small.
repeat

1. Set $v_j^l = -\lambda\alpha$ if $\|(w^l)^j\|_1 \leq 1/\alpha$ and 0 otherwise $\forall j = 1, ..., d$.
2. Compute W^{l+1} by solving the problem:

$$\min_{W \in K} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \sum_{j=1}^{d} (-v_j^l) \|w^j\|_1 \right\}. \tag{5.34}$$

3. $l \leftarrow l + 1$.

until $\|(W^l, |W_d^l|_1) - (W^{l-1}, |W_d^{l-1}|_1)\| \leq \tau(\|(W^{l-1}, |W_d^{l-1}|_1)\| + 1)$ of $|F(W^l, |W_d^l|_1) - F(W^{l-1}, |W_d^{l-1}|_1)| \leq \tau(|F(W^{l-1}, |W_d^{l-1}|_1)| + 1)$

The problem (5.34) can also be separated into L independent sub-problems of the same form,

$$\min_{w_k \in [-M,M]^d} \left\{ \frac{1}{2n} ||(Y\Theta_0)_k - Xw_k||_2^2 + \sum_{j=1}^{d} (-v_j^l) |w_k^j| \rangle \right\} \tag{5.35}$$

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 5.1 *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 (5.30) (resp. (5.31)). Furthermore, if $\|(w^*)^j\|_1 \neq \frac{1}{\alpha}$ $\forall j = 1, ..., d$, then W^* and $(W^*, |W_d^*|_1)$ are local solutions to (5.30) and (5.31), 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 (5.20) (resp. (5.23)) is polyhedral convex. Therefore, both (5.20) and (5.23) 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. $(W^*, |W_d^*|_1)$). Then W^* and $(W^*, |W_d^*|_1)$ are local solutions to (5.30) and (5.31), respectively. □

$\ell_{2,0}$(DCA1) (DCA1 with $p = 2$ for solving (5.30))

Initialization: Choose $W^0 \in K$, $l \leftarrow 0$ and let τ be a tolerance sufficient small.
repeat

1. Set $(v^l)^j \in \lambda \partial r(\|(w^l)^j\|_2)$, $j = 1, ..., d$ as

$$(v^l)^j = \begin{cases} \frac{\lambda\alpha}{\|(w^l)^j\|_2}(w^l)^j & \text{if} \quad \|(w^l)^j\|_2 > 1/\alpha \\ 0 & \text{otherwise.} \end{cases}$$

2. Compute W^{l+1} by solving the problem:

$$\min_{W \in K} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \lambda\alpha \sum_{j=1}^{d} \|w^j\|_2 - \sum_{j=1}^{d} \langle (v^l)^j, w^j \rangle \right\}. \tag{5.36}$$

3. $l \leftarrow l + 1$.

until $\|W^l - W^{l-1}\|_F \leq \tau(\|W^{l-1}\|_F + 1)$ or $|F^2(W^l) - F^2(W^{l-1})| \leq \tau(|F^2(W^{l-1})| + 1)$

Since $f(W)$ is not separable with respect to rows, the convex problem (5.36) cannot be separated into independent sub-problems as the previous cases. Here we apply a block coordinate descent algorithm for solving (5.36). We choose a row j to minimize, and consider the other rows as fixed. The resulting problem is

$$\min_{w^j \in [-M,M]^L} \left\{ \frac{1}{2n} \sum_{i=1}^{n} ||r_{(-j,i)} - X_{ij} w^j||_2^2 + \lambda\alpha \|w^j\|_2 - \langle (v^l)^j, w^j \rangle \right\}, \tag{5.37}$$

where $r_{(-j,i)} = (Y\Theta_0)_i - \sum_{m \neq j} X_{im} \hat{w}^{(m)}$. Combining the subgradient conditions with basic algebra, we get the solution $\hat{w}^j$ of the problem (5.37) without constraint as follows.

$$\hat{w}^j = \begin{cases} 0 \quad \text{if} \quad \| \sum_{i=1}^{n} r_{(-j,i)} X_{ij} + n(v^l)^j \|_2 \leq n\lambda\alpha, \\ \frac{\| \sum_{i=1}^{n} r_{(-j,i)} X_{ij} + n(v^l)^j \|_2 - n\lambda\alpha}{\| \sum_{i=1}^{n} r_{(-j,i)} X_{ij} + n(v^l)^j \|_2} \left(\sum_{i=1}^{n} r_{(-j,i)} X_{ij} + n(v^l)^j \right) \quad \text{otherwise.} \end{cases}$$

$\ell_{2,0}$**(DCA2)** (DCA2 with $p = 2$ for solving (5.31))

Initialization: Choose $W^0 \in K$, $l \leftarrow 0$.
repeat

1. Set $v_j^l = -\lambda\alpha$ if $\|(w^l)^j\|_2 \leq 1/\alpha$ and 0 otherwise $\forall j = 1, ..., d$.
2. Compute W^{l+1} by solving the problem:

$$\min_{W \in K} \left\{ \frac{1}{2n} ||Y\Theta_0 - XW||_F^2 + \sum_{j=1}^{d} (-v_j^l) \|w^j\|_2 \right\}. \tag{5.38}$$

3. $l \leftarrow l + 1$.

until $\|(W^l, |W_d^l|_2) - (W^{l-1}, |W_d^{l-1}|_2)\| \leq \tau(\|(W^{l-1}, |W_d^{l-1}|_2)\| + 1)$ of $|F(W^l, |W_d^l|_2) - F(W^{l-1}, |W_d^{l-1}|_2)| \leq \tau(|F(W^{l-1}, |W_d^{l-1}|_2)| + 1)$

The problem (5.38) is also solved by a block coordinate descent algorithm. The update with respect to the j-th row has the form

$$\hat{w}^j = \begin{cases} 0 \quad \text{if} \quad \| \sum_{i=1}^n r_{(-j,i)} X_{ij} \|_2 \leq n\lambda\alpha(-v_j^l), \\ \frac{\| \sum_{i=1}^n r_{(-j,i)} X_{ij} \|_2 - n\lambda\alpha(-v_j^l)}{\| \sum_{i=1}^n r_{(-j,i)} X_{ij} \|_2} \left(\sum_{i=1}^n r_{(-j,i)} X_{ij} \right) \quad \text{otherwise.} \end{cases}$$

Theorem 5.2 *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 (5.30).*

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 (5.31). Furthermore, if $\|(w^*)^j\|_2 \neq \frac{1}{\alpha}$ $\forall j = 1, ..., d$, then $(W^*, |W_d^*|_2)$ is a local solution to (5.31).

Proof : a) (a) is a consequence of convergence properties of general DC programs and the facts that the objective function of (5.30) is bounded below by 0 and the sequence generated by $\ell_{2,0}$(DCA1) is bounded.

b) For $p = 2$, (5.23) 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 $(W^*, |W_d^*|_2)$. Therefore, $(W^*, |W_d^*|_2)$ is a local solution to (5.31). □

5.4.1 Numerical experiments

5.4.1.1 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 standard algorithms using the $\ell_{2,1}$-regularization: SOS_GLASSO (Leng, 2008; Merchante et al., 2012) and GS_MSVM (Blondel et al., 2013). We also compare the four DCA based algorithms to study the performance of the two regularizations as well as the two different DC decompositions.

Sparse optimal scoring using group lasso (SOS_GLASSO):

GLASSO used the $\ell_{2,1}$-norm instead of the $\ell_{p,0}$-norm in the problem (5.29), that is

$$\min_{W\in\mathbb{R}^{d\times L}} \left\{ \frac{1}{2n}||Y\Theta_0 - XW||_F^2 + \lambda\|W\|_{2,1} \right\}. \tag{5.39}$$

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

Group sparse multiclass support vector machine (GS_MSVM):

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 Blondel et al. (2013) is to estimate a row-wise sparse solution W by solving the following problem.

$$\min \left\{ \sum_{i=1}^{n} \sum_{k\neq y_i} \max(1 - (w_{y_i}^T x_i - w_k^T x_i), 0)^2 + \lambda\|W\|_{2,1} \right\}. \tag{5.40}$$

In Blondel 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.

5.4.1.2 Datasets

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 5.1.

1. http://research.nhgri.nih.gov/microarray/Supplement/
2. http://www-genome.wi.mit.edu
3. http://datam.i2r.a-star.edu.sg/datasets/krbd/
4. http://research.dfci.harvard.edu/korsmeyer/Supp_pub/Supp_Armstrong_Main.html
5. http://featureselection.asu.edu/

Table 5.1: Real datasets used in experiments.

Datasets	#Features	#Samples	#Classes	Area
SRBCT[1] (SRB)	2308	83	4	Microarray, Bio
Pencillium (PEN)	3754	36	3	Microarray, Bio
ALL/AML[2] (ALL)	7129	72	3	Microarray, Bio
Leukemia[3] (LEU)	12558	248	6	Microarray, Bio
MLL-Leukemia[4] (MLL)	12582	72	3	Microarray, Bio
CLL-SUB-111[5] (CLL)	11340	111	3	Microarray, Bio
TOX-171[5] (TOX)	5748	171	4	Microarray, Bio
AR10P[5] (AR1)	2400	130	10	Image, Face
PIX10P[5] (PIX)	10000	100	10	Image, Face
PIE10P[5] (PIE)	2420	210	10	Image, Face
ORL10P[5] (ORL)	10304	100	10	Image, Face

5.4.1.3 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 dataset is split into a training set containing 2/3 of the samples and a test set containing 1/3 of the samples. This process is repeated 10 times, each with a random choice of training set and test set.

The tuning parameters are α, λ and L (number of used discriminant vectors). We fixed $\alpha = 5$, and λ, L were chosen via 5-fold cross-validation procedure on the training set from the sets of candidates 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 $\mathcal{L} = \{1, ..., C-1\}$, respectively. By this way, we avoid performing tuning parameter selection on a three-dimensional grid. The test set is used to measure the accuracy of various classifiers given by the training procedure. 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 starting point of DCA is zero. The bound M is set to 10^3. We use the same stopping criterion of the (block) coordinate descent method in Chapter 3 with tolerance 10^{-4}.

5.4.1.4 Numerical results

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), SOS_GLASSO and

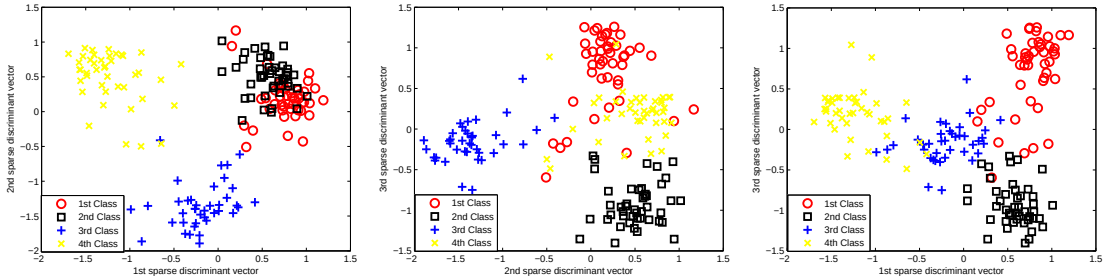

Figure 5.1: The TOX dataset was projected onto the first three sparse discriminant vectors. The samples in each class are shown by using a distinct symbol.

GS_MSVM are reported in Tables 5.2-5.3. 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 5.1.

Comments on computational results.

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 SOS_GLASSO and GS_MSVM on 10/11 datasets. SOS_GLASSO is slightly better than the DCA based algorithms on one dataset (ORL). $\ell_{1,0}$(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 SOS_GLASSO and GS_MSVM select from 0.91% to 14.63% and from 0.71% to 19.78% of features, respectively. Comparing between the $\ell_{1,0}$-regularization and $\ell_{2,0}$-regularization, we see that $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) select less features than $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) on ten out of eleven datasets. 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 DCA based algorithms not only provide a good performance in terms of feature selection, but also give a high accuracy of classifiers. The accuracy of classifiers attained the DCA based algorithms are better than SOS_GLASSO and GS_MSVM on most of datasets. Comparing between the $\ell_{1,0}$-regularization and $\ell_{2,0}$-regularization, $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) are better than $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) on 7/11 datasets while $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) are better than $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) on 3/11 datasets. On the remaining dataset (PEN), they obtain equal performance. We also see that $\ell_{1,0}$(DCA1) and $\ell_{1,0}$(DCA2) are comparable. More precisely, $\ell_{1,0}$(DCA1) is the best on 7/11 datasets and $\ell_{1,0}$(DCA2) is the best on 3/10 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

Table 5.2: Comparative results of $\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2), $\ell_{2,0}$(DCA1), $\ell_{2,0}$(DCA2), SOS_GLASSO and GS_MSVM 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)	SOS_GLASSO	GS_MSVM
SRB	42.8 (13.14) 1.85 (0.56)	**35.1** (16.07) **1.52** (0.69)	90.24 (37.15) 3.9 (1.6)	87.1 (42.34) 3.77 (1.83)	100.3 (4.24) 4.34 (0.18)	123.3 (23.11) 5.34 (1)
PEN	7.7 (4.37) 0.2 (0.11)	**3.5** (0.7) **0.09** (0.01)	7.4 (4.22) 0.19 (0.11)	14.8 (3.76) 0.49 (0.1)	41.6 (3.97) 1.1 (0.1)	48.3 (5.07) 1.28 (0.13)
ALL	31.9 (12.69) 0.45 (0.17)	**31.8** (14.57) **0.44** (0.2)	52.5 (8.12) 0.73 (0.11)	70.8 (11.8) 0.99 (0.16)	83.8 (5.75) 1.17 (0.08)	366.5 (40.16) 5.14 (0.56)
LEU	88.6 (40.32) 0.7 (0.32)	97.1 (50.11) 0.77 (0.39)	**50.1** (18.95) **0.39** (0.15)	109.4 (49.25) 0.87 (0.39)	243.9 (11.1) 1.94 (0.09)	2485.1 (1051.35) 19.78 (8.37)
MLL	35.1 (10.02) 0.27 (0.07)	**32.4** (11.11) **0.25** (0.08)	56.9 (5.38) 0.45 (0.04)	51.4 (17.89) 0.41 (0.14)	126.6 (6.85) 1.01 (0.05)	162.43 (95.62) 1.29 (0.76)
CLL	40.1 (21.66) 0.35 (0.19)	**35.6** (30.6) **0.31** (0.26)	79.3 (23.07) 0.69 (0.2)	83.3 (64.47) 0.73 (0.56)	103.8 (5) 0.91 (0.04)	80.7 (7.78) 0.71 (0.06)
TOX	**129.9** (23.15) **2.25** (0.4)	132.2 (27.42) 2.29 (0.47)	164.6 (24.31) 2.86 (0.42)	226.7 (64.11) 3.94 (1.11)	269.8 (12.53) 4.69 (0.21)	262 (7.81) 4.55 (0.13)
AR1	168.3 (49.77) 7.01 (2.07)	**88.8** (15.18) **3.7** (0.63)	342.1 (141.43) 14.29 (5.89)	231.6 (79.31) 9.65 (3.3)	351 (9.26) 14.63 (0.38)	302.8 (18.09) 12.61 (0.75)
PIX	116.1 (20.22) 1.16 (0.2)	**19.9** (9.12) **0.19** (0.09)	200.9 (18.57) 2.01 (0.18)	435.7 (92.31) 4.35 (0.92)	281.9 (16.17) 2.81 (0.16)	209.2 (20.96) 2.09 (0.2)
PIE	91 (19.31) 3.76 (0.79)	**30.5** (9.38) **1.2** (0.38)	201.9 (13.97) 8.34 (0.57)	237.1 (18.95) 9.79 (0.78)	281.3 (8.4) 11.62 (0.35)	65.3 (3.65) 2.69 (0.15)
ORL	327.5 (73.49) 3.17 (0.71)	254.4 (101.79) 2.46 (0.98)	315.9 (18.2) 3.06 (0.17)	276.8 (35.53) 2.68 (0.34)	**236.8** (8.35) **2.29** (0.08)	353.1 (27.27) 3.42 (0.26)

Table 5.3: $\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2), $\ell_{2,0}$(DCA1), $\ell_{2,0}$(DCA2), SOS_GLASSO and GS_MSVM 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 (the data is projected onto an L-dimensional space) (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)	SOS_GLASSO	GS_MSVM
SRB	**100** (0)	**100** (0)	99.1 (1.56)	99.64 (1.12)	99.61 (1.21)	97.47 (4.54)
	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)
PEN	**100** (0)	**100** (0)	**100** (0)	**100** (0)	**100** (0)	**100** (0)
	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)
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)
	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)
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)
	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)
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)
	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)
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)
	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)
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)
	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)
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)
	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)
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)
	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)
PIE	**100** (0)	**100** (0)	99 (1.35)	99.71 (0.9)	**100** (0)	98.57 (1.16)
	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)
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)
	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)

separated into independent sub-problems. Moreover, the sub-problems at each iteration are solved in parallel. We observe that SOS_GLASSO runs faster than $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) on 8/11 datasets. This is because SOS_GLASSO only solves a convex problem while $\ell_{2,0}$(DCA1) and $\ell_{2,0}$(DCA2) have to solve several problems of the same type as SOS_GLASSO.

5.5 Application to estimation of multiple covariance matrices

In recent years, much interest has focused on estimating a covariance matrix on the basis of an $n \times d$ data matrix X, where n is the number of observations and d is the number of features. Suppose that the observations $x_1, ..., x_n \in \mathbb{R}^d$ are independent and identically distributed $\mathcal{N}(0, \Sigma)$, where Σ is a positive definite $d \times d$ matrix. A natural way to estimate the covariance matrix Σ is via minimizing negative log-likelihood. The resulting optimization problem is

$$\min_{\Sigma \succ 0} \left\{ \log\det \Sigma + \mathrm{tr}(\Sigma^{-1} S) \right\}, \tag{5.41}$$

where $S = 1/n \sum_{i=1}^{n} x_i x_i^T$ is the sample covariance matrix and the notation $\Sigma \succ 0$ means that Σ is symmetric positive definite.

As mentioned in Chapter 4, estimation of sparse covariance matrix plays an important role in various areas of statistical analysis. In that chapter, we have used the ℓ_0-norm in the regularization term that leads to the following sparse covariance matrix estimation problem:

$$\min_{\Sigma \succ 0} \left\{ \log\det \Sigma + \mathrm{tr}(\Sigma^{-1} S) + \|\Sigma\|_0 \right\}, \tag{5.42}$$

where λ is a non-negative tuning parameter and $||\Sigma||_0$ denotes the ℓ_0-norm of Σ, i.e., the number of nonzero elements of matrix Σ.

In this chapter we extend the results of Chapter 4 to deal with the case of multiple classes. Suppose that we have a dataset with Q classes. For the k-th class, let X^k be an $n_k \times d$ matrix consisting of n_k observations with the number of features d common to all classes. Furthermore, we assume that the observations within each class are independent and identically distributed according to $\mathcal{N}(0, \Sigma^k)$. Let $S^k = \frac{1}{n_k}(X^k)^T X^k$ be the sample covariance matrix for the k-th class. The Q covariance matrices are estimated via minimizing negative log-likelihood

$$\min_{\{\Sigma\} \in \Omega} \left\{ \sum_{k=1}^{Q} n_k \left[\log\det \Sigma^k + \mathrm{tr}((\Sigma^k)^{-1} S^k) \right] \right\}, \tag{5.43}$$

where $\Omega = \{\{\Sigma\} := \{\Sigma^1, ..., \Sigma^Q\} : \Sigma^k \succeq \delta_k I, k = 1, ..., Q\}$. Here, I denotes the $d \times d$ identity matrix, and the notation $\Sigma^k \succeq \delta_k I$ means that $\Sigma^k - \delta_k I$ is symmetric positive semidefinite.

We define the $\ell_{p,0}$-norm of $\{\Sigma\}$ by

$$\|\{\Sigma\}\|_{p,0} = \sum_{i,j} s(\|(\Sigma_{ij}^1, ..., \Sigma_{ij}^Q)\|_p), \tag{5.44}$$

where s is the step function defined by $s(t) = 1$ if $t \neq 0$ and $s(t) = 0$ otherwise. In this chapter, we propose to estimate sparse multiple covariance matrices using the ℓ_0+$\ell_{p,0}$ regularization. The resulting group sparse multiple covariance matrix estimation problem takes the form:

$$\min_{\{\Sigma\}\in\Omega} \left\{ \sum_{k=1}^{Q} n_k \left[\log\det \Sigma^k + \mathrm{tr}((\Sigma^k)^{-1} S^k) + \lambda \|\Sigma^k\|_0 \right] + \gamma \|\{\Sigma\}\|_{p,0} \right\}, \tag{5.45}$$

where λ and γ are non-negative tuning parameters. An ℓ_0 penalty is applied to the elements of the covariance matrices and an $\ell_{p,0}$ penalty is applied to the (i, j) element across all Q covariance matrices. The first regularization term encourages sparsity within each covariance matrix Σ^k while the second one encourages a similar pattern of sparsity across all the covariance matrices. This is referred as bi-level variable selection in the estimation of multiple covariance matrices.

In the literature, there exists a number of methods that seek a sparse covariance matrix or its inverse (see e.g. Meinshausen and Buhlmann (2006); Yuan and Lin (2007); Banerjee et al. (2008); Friedman et al. (2008); Rothman et al. (2008); Danaher et al. (2014); Cai et al. (2011); Zhang and Zou (2014); Deng and Tsui (2013); Liu et al. (2014); Rothman et al. (2009); Rothman (2012); Xue et al. (2012); Bien and Tibshirani (2011); Lam and Fan (2009)). These methods only estimate single sparse covariance matrix or its inverse. We can apply these methods to separately estimate each covariance matrix Σ^k. However, these approaches can be less accurate than the jointly approaches. Recently, Danaher et al. (2014); Huang and Chen (2015); Guo et al. (2011); Lee and Liu (2015) have developed the methods based on group lasso or fused lasso for finding sparse multiple inverse covariance matrices.

In this section, we apply the proposed algorithms to solve the problem (5.42). We also study the convex approximation approaches of the ℓ_0-norm and $\ell_{p,0}$ which are the ℓ_1-norm (lasso) and the $\ell_{2,1}$-norm (group lasso), respectively. However, the resulting problem is still non-convex, and then we apply DCA to solve this problem. Among the $\ell_{p,0}$-regularizations, we continue showing that $\ell_{1,0}$ is the most interesting group regularization for DCA.

We observe that the problem (5.45) takes the form of (5.1) where the function f is given by

$$f(\{\Sigma\}) = \sum_{k=1}^{Q} n_k \left[\log\det \Sigma^k + \mathrm{tr}((\Sigma^k)^{-1} S^k) \right] + \lambda \sum_{k=1}^{Q} \sum_{ij} \eta_\alpha(\Sigma_{ij}^k),$$

and the corresponding approximate problem is

$$\min_{\{\Sigma\}\in\Omega} \left\{ F_p(\{\Sigma\}) = f(\{\Sigma\}) + \gamma \sum_{ij} \eta_\alpha(\|(\Sigma_{ij}^1, ..., \Sigma_{ij}^Q)\|_p) \right\}, \tag{5.46}$$

We note that $\log\det\Sigma^k$ is concave function while $\text{tr}((\Sigma^k)^{-1}S^k)$ is convex in Σ^k. Hence we also have a natural DC decomposition of f (see Chapter 4). However, in Chapter 4, we have showed the efficient of the special DC formulation by moving $\text{tr}((\Sigma^k)^{-1}S^k)$ to the second DC component. We have a special DC decomposition of the function f as follows:

$$f(\{\Sigma\})) = g(\{\Sigma\}) - h(\{\Sigma\}), \tag{5.47}$$

where

$$g(\{\Sigma\}) = \sum_{k=1}^{Q}\frac{\mu_k}{2}\|\Sigma^k\|_F^2 + \chi_\Omega(\{\Sigma\}) + \lambda\alpha\sum_{k=1}^{Q}\|\Sigma^k\|_1,$$

$$h(\{\Sigma\}) = \sum_{k=1}^{Q}\left[\frac{\mu_k}{2}\|\Sigma^k\|_F^2 - n_k\text{tr}((\Sigma^k)^{-1}S^k) - n_k\log\det\Sigma^k\right] + \lambda\sum_{k=1}^{Q}\sum_{ij}r_\alpha(\Sigma_{ij}^k),$$

are convex functions in $\{\Sigma\}$ when $\mu_1, ..., \mu_Q$ are large enough. For estimating $\mu_1, ..., \mu_Q$, we have the following lemma.

Lemma 5.1 *If $\mu_k \geq n_k\|S^k\|_2\delta_k^{-3}$ for $k = 1, ..., Q$, then $h(\{\Sigma\})$ is convex in $\{\Sigma\}$.*

Proof : From the proof of Lemma 4.1, if $\mu_k \geq n_k\|S^k\|_2\delta_k^{-3}$ then

$$\frac{\mu_k}{2}\|\Sigma^k\|_F^2 - n_k\text{tr}((\Sigma^k)^{-1}S^k)$$

is convex in Σ^k. Moreover, we note that

$$-\sum_{k=1}^{Q}n_k\log\det\Sigma^k + \lambda\sum_{k=1}^{Q}\sum_{ij}r_\alpha(\Sigma_{ij}^k)$$

is convex. Hence $h(\{\Sigma\})$ is convex since the sum of convex functions is also convex. The proof of lemma is complete. □

Remark 5.3 *From the Lemma 5.1, we can choose $\mu_k = n_k\|S^k\|_2\delta_k^{-3}, k = 1, ..., Q$.*

5.5.1 DCA for solving the problem (5.46) with $p = 1$

According to DCA1 with $p = 1$, at each iteration l, we have to compute $\{D^l\} \in \partial h(\{\Sigma^l\})$, $\{C^l\} \in \partial\gamma\sum_{ij}r_\alpha(\|((\Sigma^1)_{ij}^l, ..., (\Sigma^Q)_{ij}^l)\|_1)$, and $\{\Sigma^{l+1}\}$ as a solution to the problem

$$\min_{\{\Sigma\}\in\Omega}\left\{\sum_{k=1}^{Q}\left[\frac{\mu_k}{2}\|\Sigma^k\|_F^2 + \lambda\alpha\|\Sigma^k\|_1 - \langle(V^k)^l, \Sigma^k\rangle\right] + \gamma\alpha\sum_{ij}\|(\Sigma_{ij}^1, ..., \Sigma_{ij}^Q)\|_1\right\}, \tag{5.48}$$

where $\{V^l\} = \{D^l\} + \{C^l\}$. Computing $\{D^l\}$ can be split into two parts as follows: $\{D^l\} = \{A^l\} + \{B^l\}$, where

$$\{A^l\} \in \partial \sum_{k=1}^{Q} \left[\frac{\mu_k}{2} \|(\Sigma^k)^l\|_F^2 - n_k \mathrm{tr}(((\Sigma^k)^l)^{-1} S^k) - n_k \log\det(\Sigma^k)^l \right],$$

$$\{B^l\} \in \partial\lambda \sum_{k=1}^{Q} \sum_{ij} r_\alpha((\Sigma^k)^l_{ij}),$$

respectively computed by

$$(A^k)^l = \mu_k(\Sigma^k)^l + n_k[(\Sigma^k)^l]^{-1} S^k [(\Sigma^k)^l]^{-1} - n_k[(\Sigma^k)^l]^{-1}, \tag{5.49}$$

$$(B^k)^l_{ij} = \begin{cases} \lambda\alpha \mathrm{sgn}(\Sigma^k)^l_{ij} & \text{if } \alpha|(\Sigma^k)^l_{ij}| \geq 1 \\ 0 & \text{otherwise} \end{cases}. \tag{5.50}$$

The computation of $\{C^l\}$ is defined by

$$(C^k)^l_{ij} = \begin{cases} \gamma\alpha \mathrm{sgn}(\Sigma^k)^l_{ij} & \text{if } \alpha\|(\Sigma^1)^l_{ij}, ..., (\Sigma^Q)^l_{ij}\|_1 \geq 1 \\ 0 & \text{otherwise} \end{cases}.$$

For computing $\{\Sigma^{l+1}\}$, we notice that $\sum_{ij} \|(\Sigma^1_{ij}, ..., \Sigma^Q_{ij})\|_1 = \sum_{k=1}^{Q} \|\Sigma\|_1$ is separable. Hence the problem (5.48) can be separated into Q independent sub-problems of the same form,

$$\min_{\Sigma^k \succeq \delta_k I} \left\{ \ell_k(\Sigma^k) := \frac{\mu_k}{2} \|\Sigma^k\|_F^2 + \lambda\alpha\|\Sigma^k\|_1 + \gamma\alpha\|\Sigma^k\|_1 - \langle (V^k)^l, \Sigma^k \rangle \right\}. \tag{5.51}$$

For solving each convex sub-problem (5.51), we use the alternating direction method of multipliers (ADMM) Boyd et al. (2011). The augmented Lagrangian function of this problem is

$$L_1(\Sigma^k, X, Y) = \frac{\mu_k}{2} \|\Sigma^k\|_F^2 - \langle (V^k)^l, \Sigma^k \rangle + (\lambda + \gamma)\alpha\|X\|_1 + \langle Y, \Sigma^k - X \rangle + \frac{\rho}{2} \|\Sigma^k - X\|_F^2.$$

More specifically, ADMM solves the following problems at each iteration m:

$$\Sigma^{k,l,m+1} = \arg\min_{\Sigma \succeq \delta_k I} L_1(\Sigma, X^m, Y^m) \tag{5.52}$$

$$X^{m+1} = \arg\min_{X \in \mathbb{R}^{p \times p}} L_1(\Sigma^{k,l,m+1}, X, Y^m) \tag{5.53}$$

$$Y^{m+1} = Y^m + \rho(\Sigma^{k,l,m+1} - X^{m+1}). \tag{5.54}$$

The solutions (5.52) and (5.53) can be explicitly computed as follows:

$$\begin{aligned} \Sigma^{k,l,m+1} =& U D_{\delta_k} U^T \text{ where } D_{\delta_k} = \mathrm{diag}(\max(D_{ii}, \delta_k)) \text{ and} \\ & ((V^k)^l - Y^m + \rho X^m)/(\mu_k + \rho) = U D U^T, \\ X^{m+1} =& \mathcal{S}\left(\Sigma^{k,l,m+1} + Y^m/\rho, \frac{(\lambda + \gamma)\alpha}{\rho} \right), \end{aligned}$$

where $\mathcal{S}$ be the elementwise soft-thresholding operator defined by $\mathcal{S}(A,B)_{ij} = \text{sgn}(A_{ij})(|A_{ij}| - B_{ij})_+$. DCA for solving (5.46) with $p = 1$ is summarized in the following algorithm.

$\ell_0/\ell_{1,0}$**(DCA1)** (DCA1 with $p = 1$ for solving (5.46))

Initialization: Choose $\{\Sigma^0\} \in \Omega$, $l \leftarrow 0$, and τ, ϵ tolerances sufficient small.
repeat

1. Compute $(V^k)^l = (A^k)^l + (B^k)^l + (C^k)^l$, where

$$(A^k)^l = \mu_k(\Sigma^k)^l + n_k[(\Sigma^k)^l]^{-1}S^k[(\Sigma^k)^l]^{-1} - n_k[(\Sigma^k)^l]^{-1},$$

$$(B^k)^l_{ij} = \begin{cases} \lambda\alpha\text{sgn}(\Sigma^k)^l_{ij} & \text{if } \alpha|(\Sigma^k)^l_{ij}| \geq 1 \\ 0 & \text{otherwise} \end{cases},$$

$$(C^k)^l_{ij} = \begin{cases} \gamma\alpha\text{sgn}(\Sigma^k)^l_{ij} & \text{if } \alpha\|((\Sigma^1)^l_{ij}, ..., (\Sigma^Q)^l_{ij}\|_1 \geq 1 \\ 0 & \text{otherwise} \end{cases}.$$

2. For $k = 1, ..., Q$ compute $(\Sigma^k)^{l+1}$ by ADMM:
Initialization: $m = 0$, $(X^k)^0 = \mathcal{S}\left((V^k)^l, \alpha(\lambda + \gamma)\right)$, $Y^0 = 0$.
repeat
+ Compute $\Sigma^{k,l,m+1} = UD_{\delta_k}U^T$ where $D_{\delta_k} = \text{diag}(\max(D_{ii}, \delta_k))$ and $((V^k)^l - Y^m + \rho X^m)/(\mu_k + \rho) = UDU^T$,
+ Compute $X^{m+1} = \mathcal{S}\left(\Sigma^{k,l,m+1} + Y^m/\rho, \frac{(\lambda+\gamma)\alpha}{\rho}\right)$,
+ Compute $Y^{m+1} = Y^m + \rho(\Sigma^{k,l,m+1} - X^{m+1})$,
+ $m \leftarrow m + 1$.
until $|\ell_k(\Sigma^{k,l,m}) - \ell_k(\Sigma^{k,l,m-1})| \leq \epsilon$
3. $l \leftarrow l + 1$.

until $\|\{\Sigma^l\} - \{\Sigma^{l-1}\}\| \leq \tau(\|\{\Sigma^{l-1}\}\| + 1)$ or $|F_1(\{\Sigma^l\}) - F_1(\{\Sigma^{l-1}\})| \leq \tau(|F_1(\{\Sigma^{l-1}\})| + 1)$

5.5.2 DCA for solving the problem (5.46) with $p = 2$

According to DCA1 with $p = 2$, at each iteration l, we have to compute $\{D^l\} \in \partial h(\{\Sigma^l\})$, $\{C^l\} \in\in \partial\gamma \sum_{ij} r_\alpha(\|((\Sigma^1)^l_{ij}, ..., (\Sigma^Q)^l_{ij})\|_2)$, and $\{\Sigma^{l+1}\}$ as a solution to the following problem:

$$\min_{\{\Sigma\}\in\Omega} \left\{ \ell(\{\Sigma\}) := \sum_{k=1}^{Q} \left[\frac{\mu_k}{2}\|\Sigma^k\|_F^2 + \lambda\alpha\|\Sigma^k\|_1 - \langle (V^k)^l, \Sigma^k\rangle\right] + \gamma\alpha \sum_{ij} \|(\Sigma^1_{ij}, ..., \Sigma^Q_{ij})\|_2 \right\}, \tag{5.55}$$

where $\{V^l\} = \{D^l\} + \{C^l\}$. Computing $\{D^l\}$ can be split into two parts as follows: $\{D^l\} = \{A^l\} + \{B^l\} + \{C^l\}$, where $\{A^l\}, \{B^l\}$ are respectively computed by using (5.49) and (5.50), and

$$\{C^l\} \in \partial\gamma \sum_{ij} r_\alpha(\|((\Sigma^1)^l_{ij}, ..., (\Sigma^Q)^l_{ij})\|_2),$$

computed by

$$(C^k)^l_{ij} = \begin{cases} \gamma\alpha(\Sigma^k)^l_{ij}/\|(\Sigma^1)^l_{ij}, ..., (\Sigma^Q)^l_{ij}\|_2 & \text{if } \alpha\|(\Sigma^1)^l_{ij}, ..., (\Sigma^Q)^l_{ij}\|_2 \geq 1 \\ 0 & \text{otherwise} \end{cases}.$$

The problem (5.55) cannot be separated into independent sub-problems as the previous case. Here we apply the ADMM algorithm for solving this problem. The augmented Lagrangian function of the problem (5.55) is

$$\begin{aligned} L_2(\{\Sigma\}, \{X\}, \{Y\}) = &\sum_{k=1}^{Q} [\frac{\mu_k}{2}\|\Sigma^k\|_F^2 - \langle (V^k)^l, \Sigma^k\rangle + \lambda\alpha\|X^k\|_1 + \langle Y^k, \Sigma^k - X^k\rangle \\ &+ \frac{\rho}{2}\|\Sigma^k - X^k\|_F^2] + \gamma\alpha \sum_{ij} \|(X^1_{ij}, ..., X^Q_{ij})\|_2. \end{aligned}$$

More specifically, ADMM solves the following problems at each iteration m:

$$\{\Sigma^{l,m+1}\} = \arg\min_{\{\Sigma\}\in\Omega} L_2(\{\Sigma\}, \{X^m\}, \{Y^m\} \tag{5.56}$$

$$\{X^{m+1}\} = \arg\min_{\{X\}} L_2(\{\Sigma^{l,m+1}\}, \{X\}, \{Y^m\}) \tag{5.57}$$

$$\{Y^{m+1}\} = \{Y^m\} + \rho(\{\Sigma^{l,m+1}\} - \{X^{m+1}\}). \tag{5.58}$$

The solution (5.56) can be explicitly computed as follows, $k = 1, ..., Q$,

$$\begin{aligned} \Sigma^{k,l,m+1} = &UD_{\delta_k}U^T \text{ where } D_{\delta_k} = \text{diag}(\max(D^k_{ii}, \delta_k)) \text{ and} \\ &\left((V^k)^l - (Y^k)^m + \rho(X^k)^m\right)/(\mu_k + \rho) = UD^kU^T. \end{aligned}$$

The solution (5.57) can be computed as follows. We denote $X_{ij} = (X^1_{ij}, ..., X^Q_{ij})$. Combining the subgradient conditions with basic algebra, we have $(X_{ij})^{m+1} = 0$ if

$$\|\mathcal{S}\left((\Sigma_{ij})^{l,m+1} + (Y_{ij})^m/\rho, \lambda\alpha/\rho\right)\|_2 \leq \alpha\gamma/\rho, \tag{5.59}$$

and otherwise $(X_{ij})^{m+1}$ satisfies

$$\left(1 + \frac{\lambda\gamma}{\rho\|(X_{ij})^{m+1}\|_2}\right)(X_{ij})^{m+1} = \mathcal{S}\left((\Sigma_{ij})^{l,m+1} + (Y_{ij})^m/\rho, \lambda\alpha/\rho\right). \tag{5.60}$$

Taking the norm of both sides, we obtain

$$\|(X_{ij})^{m+1}\|_2 = \|\mathcal{S}\left((\Sigma_{ij})^{l,m+1} + (Y_{ij})^m/\rho, \lambda\alpha/\rho\right)\|_2 - \alpha\gamma/\rho. \tag{5.61}$$

Substituting this expression for $\|(X_{ij})^{m+1}\|_2$ into Eq. (5.60) and simplifying gives

$$(X_{ij})^{m+1} = \left(1 - \frac{\alpha\gamma/\rho}{\|\mathcal{S}\left((\Sigma_{ij})^{l,m+1} + (Y_{ij})^m/\rho, \lambda\alpha/\rho\right)\|_2}\right) \mathcal{S}\left((\Sigma_{ij})^{l,m+1} + (Y_{ij})^m/\rho, \lambda\alpha/\rho\right).$$

For summary, we describe the algorithm for (5.46) with $p = 2$ in the algorithm below.

$\ell_0/\ell_{2,0}$**(DCA1)** (DCA1 with $p = 2$ for solving (5.46))

Initialization: Choose $\{\Sigma^0\} \in \Omega$, $l \leftarrow 0$, and let τ, ϵ be tolerances sufficient small.
repeat

1. Compute $(V^k)^l = (A^k)^l + (B^k)^l + (C^k)^l$ with

$$(A^k)^l = \mu_k(\Sigma^k)^l + n_k[(\Sigma^k)^l]^{-1}S^k[(\Sigma^k)^l]^{-1} - n_k[(\Sigma^k)^l]^{-1},$$

$$(B^k)^l_{ij} = \begin{cases} \lambda\alpha\text{sgn}(\Sigma^k)^l_{ij} & \text{if } \alpha|(\Sigma^k)^l_{ij}| \geq 1 \\ 0 & \text{otherwise} \end{cases},$$

$$(C^k)^l_{ij} = \begin{cases} \gamma\alpha(\Sigma^k)^l_{ij}/\|(\Sigma^1)^l_{ij}, ..., (\Sigma^Q)^l_{ij}\|_2 & \text{if } \alpha\|(\Sigma^1)^l_{ij}, ..., (\Sigma^Q)^l_{ij}\|_2 \geq 1 \\ 0 & \text{otherwise} \end{cases}.$$

2. Compute $\{\Sigma^{l+1}\}$ by ADMM:
Initialization: $m = 0$, $\{Y^0\} = 0$, $(X_{ij})^0 = 0$ if $\|\mathcal{S}\left((V_{ij})^l, \lambda\alpha/\max_k \mu_k\right)\|_2 \leq \alpha\gamma/\max_k \mu_k$, and otherwise

$$(X_{ij})^0 = \left(1 - \frac{\alpha\gamma/\max_k \mu_k}{\|\mathcal{S}\left((V_{ij})^l, \lambda\alpha/\max_k \mu_k\right)\|_2}\right) \mathcal{S}\left((V_{ij})^l, \lambda\alpha/\max_k \mu_k\right).$$

repeat
+ For $k = 1, ..., Q$ compute $\Sigma^{k,l,m+1} = UD_{\delta_k}U^T$ with $D_{\delta_k} = \text{diag}(\max(D^k_{ii}, \delta_k))$ and $\left((V^k)^l - (Y^k)^m + \rho(X^k)^m\right)/(\mu_k + \rho) = UD^kU^T$,
+ Set $(X_{ij})^{m+1} = 0$ if $\|\mathcal{S}\left((\Sigma_{ij})^{l,m+1} + (Y_{ij})^m/\rho, \lambda\alpha/\rho\right)\|_2 \leq \alpha\gamma/\rho$, and otherwise

$$(X_{ij})^{m+1} = \left(1 - \frac{\alpha\gamma/\rho}{\|\mathcal{S}\left((\Sigma_{ij})^{m+1} + (Y_{ij})^m/\rho, \lambda\alpha/\rho\right)\|_2}\right) \mathcal{S}\left((\Sigma_{ij})^{l,m+1} + (Y_{ij})^m/\rho, \lambda\alpha/\rho\right).$$

+ $\{Y^{m+1}\} = \{Y^m\} + \rho(\{\Sigma^{l,m+1}\} - \{X^{m+1}\})$,
+ $m \leftarrow m + 1$.
until $|\ell(\{\Sigma^{l,m}\}) - \ell(\{\Sigma^{l,m-1}\})| \leq \epsilon$
3. $l \leftarrow l + 1$.
until $\|\{\Sigma^l\} - \{\Sigma^{l-1}\}\| \leq \tau(\|\{\Sigma^{l-1}\}\| + 1)$ or $|F_2(\{\Sigma^l\}) - F_2(\{\Sigma^{l-1}\})| \leq \tau(|F_2(\{\Sigma^{l-1}\})| + 1)$

Theorem 5.3 (Convergence properties of $\ell_0/\ell_{1,0}$(DCA1) and $\ell_0/\ell_{2,0}$(DCA1)) *Let $\{\{\Sigma^l\}\}$ be the sequence generated by $\ell_0/\ell_{1,0}$(DCA1) (resp. $\ell_0/\ell_{2,0}$(DCA1)), we have*
(a) $\{F_1(\{\Sigma^l\})\}$ (resp. $\{F_2(\{\Sigma^l\})\}$) is decreasing.
(b) $\{\{\Sigma^l\}\}$ is bounded.
(c) $\sum_{l=0}^{+\infty} \|\{\Sigma^l\} - \{\Sigma^{l+1}\}\|_F^2 < +\infty$, and hence $\|\{\Sigma^l\} - \{\Sigma^{l+1}\}\|_F \to 0$ as $l \to +\infty$.
(d) The sequence $\{\{\Sigma^l\}\}$ has at least one limit point and every limit point of this sequence is a critical point of the problem (5.46).

Proof : The theorem is proved analogously to Theorem 4.1. □

5.5.3 Group variable selection using $\ell_1/\ell_{2,1}$-regularization

In this section, we study the convex approximation approaches of ℓ_0-norm and $\ell_{p,0}$-norm which are respectively ℓ_1-norm (lasso) and $\ell_{2,1}$-norm (group lasso). By replacing the $\ell_0/\ell_{p,0}$-norm with the $\ell_1/\ell_{2,1}$-norm, the resulting problem is

$$\min_{\{\Sigma\}\in\Omega} \left\{ \sum_{k=1}^{Q} n_k \left[\log\det\Sigma^k + \mathrm{tr}((\Sigma^k)^{-1}S^k)\right] + \lambda \sum_{k=1}^{Q} \|\Sigma^k\|_1 + \gamma \sum_{ij} \|(\Sigma_{ij}^1, ..., \Sigma_{ij}^Q)\|_2 \right\}. \tag{5.62}$$

This problem is still non-convex. We use DCA for solving it. Similar to the previous section, (5.62) can be reformulated as the DC program

$$\min_{\{\Sigma\}} \{F(\{\Sigma\} := G(\{\Sigma\}) - H(\{\Sigma\})\}, \tag{5.63}$$

where

$$G(\{\Sigma\}) = \sum_{k=1}^{Q} \frac{\mu_k}{2}\|\Sigma^k\|_F^2 + \chi_\Omega(\{\Sigma\}) + \lambda\alpha \sum_{k=1}^{Q} \|\Sigma^k\|_1 + \gamma\alpha \sum_{ij} \|(\Sigma_{ij}^1, ..., \Sigma_{ij}^Q)\|_2,$$

and

$$H(\{\Sigma\}) = \sum_{k=1}^{Q} \left[\frac{\mu_k}{2}\|\Sigma^k\|_F^2 - n_k \mathrm{tr}((\Sigma^k)^{-1}S^k) - n_k \log\det\Sigma^k\right].$$

DCA applied to DC program (5.63) is similar to $\ell_0/\ell_{2,0}$(DCA1). We simply replace the computation of $\{V^l\} \in \partial H(\{\Sigma^l\})$ with $\{V^l\} = \{A^l\}$ computed by (5.49). The DCA for solving (5.63) is described as follows.

$\ell_1/\ell_{2,1}$(DCA) (DCA for solving (5.63))

Initialization: Choose $\{\Sigma^0\} \in \Omega$, $l \leftarrow 0$.
repeat
1. Compute $\{V^l\} \in \partial H(\{\Sigma^l\})$ with
$$(V^k)^l = \mu_k(\Sigma^k)^l + n_k[(\Sigma^k)^l]^{-1}S^k[(\Sigma^k)^l]^{-1} - n_k[(\Sigma^k)^l]^{-1}.$$
2. Compute $\{\Sigma^{l+1}\}$ by ADMM:
Initialization: $m = 0$, $\{Y^0\} = 0$, $(X_{ij})^0 = 0$ if $\|\mathcal{S}\left((V_{ij})^l, \lambda/\max_k \mu_k\right)\|_2 \leq \gamma/\max_k \mu_k$, and otherwise
$$(X_{ij})^0 = \left(1 - \frac{\gamma/\max_k \mu_k}{\|\mathcal{S}\left((V_{ij})^l, \lambda/\max_k \mu_k\right)\|_2}\right)\mathcal{S}\left((V_{ij})^l, \lambda/\max_k \mu_k\right).$$
repeat
+ For $k = 1, ..., Q$ compute $\Sigma^{k,l,m+1} = UD_{\delta_k}U^T$ with $D_{\delta_k} = \text{diag}(\max(D^k_{ii}, \delta_k))$ and $\left((V^k)^l - (Y^k)^m + \rho(X^k)^m\right)/(\mu_k + \rho) = UD^kU^T$,
+ Set $(X_{ij})^{m+1} = 0$ if $\|\mathcal{S}\left((\Sigma_{ij})^{l,m+1} + (Y_{ij})^m/\rho, \lambda/\rho\right)\|_2 \leq \lambda/\rho$, and otherwise
$$(X_{ij})^{m+1} = \left(1 - \frac{\gamma/\rho}{\|\mathcal{S}\left((\Sigma_{ij})^{m+1} + (Y_{ij})^m/\rho, \lambda/\rho\right)\|_2}\right)\mathcal{S}\left((\Sigma_{ij})^{l,m+1} + (Y_{ij})^m/\rho, \lambda/\rho\right).$$
+ $\{Y^{m+1}\} = \{Y^m\} + \rho(\{\Sigma^{l,m+1}\} - \{X^{m+1}\})$,
+ $m \leftarrow m + 1$.
until Stopping criterion.
3. $l \leftarrow l + 1$.
until Stopping criterion.

The stopping criterion of $\ell_1/\ell_{2,1}$(DCA) is used as in $\ell_0/\ell_{2,0}$(DCA1)

Theorem 5.4 (Convergence properties of $\ell_1/\ell_{2,1}$(DCA)) *Let $\{\{\Sigma^l\}\}$ be the sequence generated by $\ell_1/\ell_{2,1}$(DCA), we have*
(a) $\{F(\{\Sigma^l\})\}$ is decreasing.
(b) $\{\{\Sigma^l\}\}$ is bounded.
(c) $\sum_{l=0}^{+\infty} \|\{\Sigma^l\} - \{\Sigma^{l+1}\}\|_F^2 < +\infty$, and hence $\|\{\Sigma^l\} - \{\Sigma^{l+1}\}\|_F \to 0$ as $l \to +\infty$.
(d) The sequence $\{\{\Sigma^l\}\}$ has at least one limit point and every limit point of this sequence is a critical point of the problem (5.63).

Proof : The theorem is proved analogously to Theorem 4.1. □

5.5.4 Numerical experiments

The numerical experiments aim to evaluate the performance of the three approaches: $\ell_0/\ell_{1,0}$(DCA1), $\ell_0/\ell_{2,0}$(DCA1), and the standard approach based on the $\ell_1 + \ell_{2,1}$-regularization model (5.62) ($\ell_1/\ell_{2,1}$(DCA)).

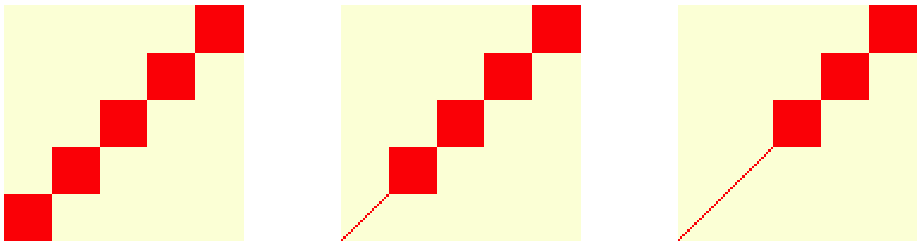

Figure 5.2: Sparse multiple covariance matrices in Model 1

5.5.4.1 Experimental setups

The proposed algorithms are implemented in R software and all algorithms are performed on a PC Intel i7 CPU3770, 3.40 GHz of 8GB RAM.

In experiments, we set the stop tolerances $\epsilon = \tau = 10^{-4}$. The starting point $\{\Sigma^0\}$ of DCA is the sample covariance matrices $\{S^1, ..., S^Q\}$. The values of parameter λ and γ are chosen through a 5-fold cross-validation procedure on training set. The approximation parameter α of the Capped-ℓ_1 is set 1. Note that the $\ell_{1,0}$ regularization also promotes sparsity within the group. Hence, we set $\lambda = 0$ in $\ell_0/\ell_{1,0}$(DCA1) to avoid performing tuning this parameter.

5.5.4.2 Experiments on synthetic datasets

We evaluate the performance of $\ell_0/\ell_{1,0}$(DCA1) and $\ell_0/\ell_{2,0}$(DCA1) on two synthetic datasets. We consider two types of covariance graphs with three-class:

Model 1: We generate a covariance matrix for the first class as follows. $\Sigma^1 = \text{diag}(\Sigma_1, ..., \Sigma_5)$, where $\Sigma_1, ..., \Sigma_5$ are dense matrices. We create Σ^2 by resetting one of its 5 sub-network blocks to the identity, i.e., $\Sigma^2 = \text{diag}(I, \Sigma_2, ..., \Sigma_5)$. Resetting an additional sub-network block to the identity, we have $\Sigma^3 = \text{diag}(I, I, \Sigma_3, ..., \Sigma_5)$. A example is showed in Figure 5.2.

Model 2: $\Sigma^1 = \text{diag}(\Sigma_1, ..., \Sigma_5)$ again, however each submatrix Σ_k is zero except the elements in the last row and the last column. This corresponds to a sub-graph with five connected components each of which has all nodes connected to one particular node. Similarly to model 1, we create $\Sigma^2 = \text{diag}(I, \Sigma_2, ..., \Sigma_5)$ and $\Sigma^3 = \text{diag}(I, I, \Sigma_3, ..., \Sigma_5)$. A example is showed in Figure 5.3.

The nonzero entries of matrices $\Sigma^k, k = 1, 2, 3$ are randomly drawn in the set $\{+1, -1\}$. Finally, for each class we generate independent, identically distributed observations $X^k =$

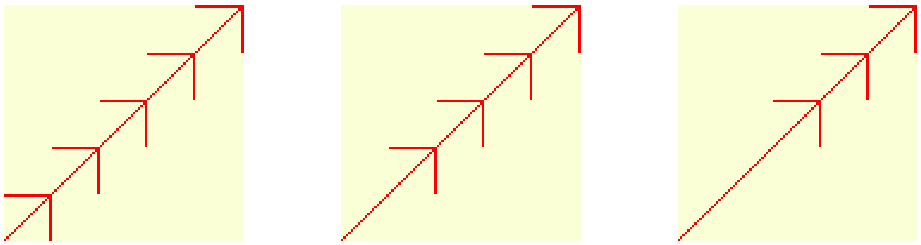

Figure 5.3: Sparse multiple covariance matrices in Model 2

$[x_1^k, ..., x_{n_k}^k]$ from an $\mathcal{N}(0, \Sigma^k)$ distribution. In this experiment, for each model, we generate 10 training sets with size $n_1 = n_2 = n_3 = 200, d = 100$.

To evaluate the performance of each method, we consider three loss functions which are the average root-mean-square error (ARMSE), the average entropy loss (AEN), and the average Kullback-Leibler loss (AKL), respectively.

$$\text{ARMSE} = \frac{1}{Q}\sum_{k=1}^{Q} ||\hat{\Sigma}^k - \Sigma^k||_F/d,$$

$$\text{AEN} = \frac{1}{Q}\sum_{k=1}^{Q} \left[-\log\det(\hat{\Sigma}^k(\Sigma^k)^{-1}) + \text{tr}(\hat{\Sigma}^k(\Sigma^k)^{-1}) - d\right],$$

$$\text{AKL} = \frac{1}{Q}\sum_{k=1}^{Q} \left[-\log\det((\hat{\Sigma}^k)^{-1}\Sigma^k) + \text{tr}((\hat{\Sigma}^k)^{-1}\Sigma^k) - d\right],$$

where $\hat{\Sigma}^k$ is a sparse estimate of the covariance matrix Σ^k.

The experimental results on synthetic datasets are given in Table 5.4. In this Table, the average of root-mean-square error (ARMSE), entropy loss (AEN), Kullback-Leibler loss (AKL), number of nonzero elements on each covariance matrix (NZ1, NZ2, NZ3) and their sum (NZ), CPU time in seconds, and their standard deviations over 10 samples are reported.

Table 5.4: Comparative results of $\ell_0/\ell_{1,0}$(DCA1), $\ell_0/\ell_{2,0}$(DCA1), and $\ell_1/\ell_{2,1}$(DCA) in terms of the average of root-mean-square error (ARMSE), entropy loss (AEN), Kullback-Leibler loss (AKL), number of nonzero elements, CPU time in second (and their standard deviations) over 10 runs. Bold fonts indicate the best result in each row.

		$\ell_0/\ell_{1,0}$(DCA1)	$\ell_0/\ell_{2,0}$(DCA1)	$\ell_1/\ell_{2,1}$(DCA)
Model 1	ARMSE	**0.381** (0.004)	0.415 (0.007)	0.445 (0.002)
	AEN	**12.97** (1.07)	17.53 (1.12)	18.08 (2.62)
	AKL	**17.31** (2.24)	18.58 (3.1)	31.86 (2.75)
	NZ1	**1903.2** (298.61)	2138.6 (313.7)	2242.4 (271.5)
	NZ2	1781.6 (307.96)	2172.18 (281.6)	**1464.6** (316.4)
	NZ3	**1728.2** (288.12)	2058.37 (215.5)	1918.8 (251.2)
	NZ	**5413** (892.69)	6369.15 (810.8)	5625.8 (839.1)
	CPUs	**642.11** (3.74)	3180.56 (7.92)	3471.66 (5.18)
Model 2	ARMSE	**0.082** (0.003)	0.09 (0.007)	0.094 (0.005)
	AEN	**3.57** (0.52)	18.02 (1.31)	28.08 (2.16)
	AKL	**3.93** (0.56)	6.65 (1.66)	7.76 (1.82)
	NZ1	**352.8** (13.51)	394.18 (52.47)	448.6 (52.3)
	NZ2	**259** (13.61)	347.45 (38.52)	378.27 (36.1)
	NZ3	**255.8** (11.18)	359.72 (12.98)	264.61 (31.6)
	NZ	**867.6** (38.3)	1101.35 (103.97)	1091.48 (120)
	CPUs	**267.22** (39.33)	3843.57 (27.37)	5593.15 (24.61)

We observe from Table 5.4 that in the both models, $\ell_0/\ell_{1,0}$(DCA1) gives the best results in terms of three losses. In terms of the sparsity, the number of the nonzero elements, this approach also achieves better performances than the other approaches. The second and third performing approaches with respect to the losses and the sparsity are $\ell_0/\ell_{2,0}$(DCA1) and $\ell_1/\ell_{2,1}$(DCA), respectively.

Regarding the training time, $\ell_0/\ell_{1,0}$(DCA1) is much faster than the other algorithms. This can be explained by the fact that $\ell_0/\ell_{1,0}$(DCA1) leads to the sequence of convex problems which can be separated into the independent sub-problems.

5.5.4.3 Experiments on real datasets

We illustrate the use of the sparse covariance matrix estimation problem via a real application: a classification problem based sparse quadratic discriminant analysis (SQDA). This application requires estimates of the covariance matrices.

Let X be an $n \times d$ training data matrix with observations on the rows and features on the columns. We assume that the n_k observations x_i^k $(i = 1, ..., n_k)$ within the k-th class C_k are normal distributed $\mathcal{N}(\mu_k, \Sigma_k)$. We denote the prior probability of the k-th class by π_k. The quadratic discriminant function is

$$\delta_k(x) = -\frac{1}{2}\log\det\Sigma_k - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k. \tag{5.64}$$

Then the predicted class for a new observation x is $\arg\max_k \delta_k(x)$. The decision boundary between each pair of classes k and l is described by a quadratic equation $\{x : \delta_k(x) = \delta_l(x)\}$.

In practice we do not know π_k, μ_k, Σ_k, and will need to estimate them using the training data. In this work, we directly estimate the covariance matrices $\Sigma_1, ..., \Sigma_Q$ by using $\ell_0/\ell_{1,0}$(DCA), $\ell_0/\ell_{2,0}$(DCA), and $\ell_1/\ell_{2,1}$(DCA). Note that if S^k is singular, then we replace it by $S^k + \epsilon I_p$, where ϵ is chosen through a 5-fold cross-validation.

For the experiment, we evaluate the proposed algorithms on two datasets from UCI Machine Learning Repository (Ionosphere and Waveform 2). We use the cross-validation scheme to validate the performance of various approaches on these two datasets. The dataset is split into a training set containing 2/3 of the samples and a test set containing 1/3 of the samples. This process is repeated 10 times, each with a random choice of training set and test set. The parameter λ and γ are chosen via 5-fold cross-validation.

Table 5.5: Comparative results of Ionosphere and Waveform 2 datasets in terms of the average of percentage of testing errors, training errors, training time in second and their standard deviations over 10 training/test set splits. The bold font indicates the best result in each column.

		Testing error (%)	Training error (%)	Training time (s)
Ionosphere	$\ell_0/\ell_{1,0}$(DCA1)	**5.13** (1.3)	**3.41** (0.48)	**0.094** (0.02)
	$\ell_0/\ell_{2,0}$(DCA1)	**5.13** (0.54)	3.84 (0.72)	0.94 (0.01)
	$\ell_1/\ell_{2,1}$(DCA)	6.79 (1.78)	4.27 (0.82)	0.97 (0.04)
Waveform 2	$\ell_0/\ell_{1,0}$(DCA1)	**13.01** (0.25)	**11.41** (1.3)	**3.28** (1.14)
	$\ell_0/\ell_{2,0}$(DCA1)	14.64 (0.38)	12.68 (0.32)	157.64 (23.99)
	$\ell_1/\ell_{2,1}$(DCA)	15.6 (1.01)	14.57 (0.39)	259.28 (84.06)

The computational results are reported in Table 5.5. We observe that, on the Ionosphere dataset, $\ell_0/\ell_{1,0}$(DCA1) and $\ell_0/\ell_{2,0}$(DCA1) are comparable and better than $\ell_1/\ell_{2,1}$(DCA) in terms of the testing error and training error. On the Waveform 2, $\ell_0/\ell_{1,0}$(DCA1) gives better testing error and training error than the both algorithms $\ell_0/\ell_{2,0}$(DCA1) and $\ell_1/\ell_{2,1}$(DCA). In terms of training time, $\ell_0/\ell_{1,0}$(DCA1) is significantly faster than $\ell_0/\ell_{2,0}$(DCA1) and $\ell_1/\ell_{2,1}$(DCA) on these two datasets.

5.6 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 approach has been investigated from both a theoretical and an algorithmic point of view. Using the Capped-ℓ_1 approximation function, we have proved that the Capped-ℓ_1 approximate problem is equivalent to the original problem with suitable parameter α. Considering the two equivalent formulations of the approximate problem we have developed DC programming and DCA 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, three of four DCA schemes ($\ell_{1,0}$(DCA1), $\ell_{1,0}$(DCA2) and $\ell_{2,0}$(DCA2)) have the interesting convergence properties: they converge after a finite number of iterations to local solution. 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) 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.

In the estimation problem of multiple covariance matrices, we use two regularization terms in order to encourage simultaneously sparsity within each covariance matrix and across all the covariance matrices. Among the proposed DCA schemes, $\ell_0/\ell_{1,0}$(DCA1) is the best interesting. At iteration, this scheme solves independent convex sub-problems. We also propose explicitly algorithms for solving the convex sub-problems. Numerical experiments on both simulation and real datasets have shown that our methods are promising.

For the future works, we plan to study group variable selection for other applications. We believe that the success of the $\ell_{1,0}$-regularization motivates and opens up a new avenue for the group variable selection problems.

Part III

Stochastic Learning

Chapter 6

Stochastic DCA and Application to Latent Log-Linear Model

Abstract: *In this chapter, we introduce stochastic DCA for minimizing a large sum of non-convex functions, a problem of utmost importance in machine learning. With appropriate DC components, we present two special versions of the stochastic DCA: stochastic proximal DCA and stochastic proximal Newton DCA. We also show that stochastic gradient descent algorithm and stochastic proximal descent algorithm are special versions of stochastic DCA. As an application, we apply the proposed algorithm to a log-linear model incorporating latent variables. Parameter estimation of this model often results in an optimization problem involving a rational function of mixtures of exponential terms. It is a non-convex and large-scale problem which is very hard to solve. Experiments on the some real datasets show the efficiency of the proposed algorithms.*

6.1 Introduction

The increase of applications using big data causes the difficulty in computation, especially when both the number of features and samples are large. Among methods proposed to address this problem, stochastic has been recently widely used as an efficient technique. In this chapter, we introduce a stochastic scheme based on DCA for solving a large sum of non-convex functions:

$$\min_{\Theta\in\Omega}\left\{f(\Theta)=\frac{1}{n}\sum_{i=1}^{n}f^i(\Theta)\right\}, \tag{6.1}$$

where $f^i:\mathbb{R}^d\to\mathbb{R}$ are DC functions, and Ω is a convex subset of $\mathbb{R}^d$. Θ represents some model parameters and each function f^i measures the adequacy of the parameters Θ to an observed data point indexed by i.

There are many application problems of the form (6.1) in machine learning, signal processing and other domains. The problem (6.1) also arises in the minimization of an expected loss that depends on Θ and some random vector. Hence, the objective function is either

an expected loss with respect to a discrete distribution or is a finite sample approximation to an expected loss. Let us mention some important application problems corresponding to the model (6.1).

Sparse logistic regression problem: Given n data points $(y_i, x_i)_{i=1}^n$ where observation vector $x_i \in \mathbb{R}^d$ and label $y_i \in \{-1, 1\}$. We consider the sparse logistic regression problem, which can be formulated as follows:

$$\min_{\Theta \in \Omega} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, x_i^T \Theta) + \lambda \sum_{i=1}^{d} \eta_\alpha(\Theta_i) \right\}, \tag{6.2}$$

where $\ell(y, x^T\Theta) = \log(1 + \exp(-yx^T\Theta))$, and η_α is a DC approximation function of the step function such as the Capped-ℓ_1. This problem takes the form of (6.1) with $f^i(\Theta) = \ell(y_i, x_i^T\Theta) + n\lambda \sum_{j=1}^{d} \eta_\alpha(\Theta_j)$.

Problem with many constraints: We consider the problem

$$\begin{array}{ll} \min\limits_{x \in \mathbb{X}} & f(x) \\ \text{subject to} & g_i(x) \leq 0, i = 1, ..., n, \end{array} \tag{6.3}$$

where the number of constraints n is very large in many practical problems. One way can tackle this problem is to use a penalty function. The resulting penalty problem takes the form of (6.1):

$$\min_{x \in \mathbb{X}} f(x) + \tau \sum_{i=1}^{n} p(g_i(x)), \tag{6.4}$$

where τ is a positive penalty parameter and p is a penalty function such as the nondifferentiable penalty $p(t) = \max\{0, t\}$.

Minimizing an expected loss in stochastic programming: Considering the minimization problem of an expected loss

$$\min_{\Theta \in \Omega} \mathbb{E}[f(\Theta, \xi)], \tag{6.5}$$

where f is a function of Θ and a random variable ξ. When ξ has a discrete distribution or we use the sample average approximation method, the problem can be rewritten of the form (6.1):

$$\min_{\Theta \in \Omega} \frac{1}{n} \sum_{i=1}^{n} f(\Theta, \xi_i), \tag{6.6}$$

where $\xi_1, ..., \xi_n$ are independent samples of the random variables ξ.

Latent log-linear model: Let $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^N$ be a set of labeled training data with observation vector $x_i \in \mathbb{R}^d$ and labels $y_i \in \mathcal{Y} = \{1, ..., Q\}$. A log-linear model with the parameters $(\theta_i, \lambda_i) \in \mathbb{R}^d \times \mathbb{R}, i = 1, ..., Q$ is a model for the class-posterior probabilities of the form

$$P(y|x) = \frac{\exp\left(\langle x, \theta_y \rangle + \lambda_y\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\langle x, \theta_{y'} \rangle + \lambda_{y'}\right)}. \tag{6.7}$$

The predicted class for a new observation x is

$$\arg\max_y P(y|x) = \arg\max_y \{\langle x, \theta_y \rangle + \lambda_y\}. \tag{6.8}$$

Therefore, the decision boundary between each pair of classes i and j is described by a linear equation $\{x : \langle x, \theta_i - \theta_j \rangle + \lambda_i - \lambda_j = 0\}$. In the supervised learning setup with the labeled data $\mathcal{D}_{train}$, learning the log-linear model corresponds to maximizing the conditional log-likelihood on $\mathcal{D}_{train}$, namely

$$\max \left\{ \frac{1}{N} \sum_{i=1}^{N} \log P(y_i|x_i) \right\}. \tag{6.9}$$

In order to extend this model is to create a latent log-linear model by incorporating a latent (or hidden) variable h (Deselaers et al., 2012). The latent log-linear model is an extension of log-linear model to increase the flexibility of the model. The posterior probability for a label y is defined by

$$P_\Theta(y|x_i) = \frac{\sum_h \exp(\langle x_i, \theta_y^h \rangle + \lambda_y^h)}{\sum_{y' \in \mathcal{Y}} \sum_h \exp(\langle x_i, \theta_{y'}^h \rangle + \lambda_{y'}^h)}, \tag{6.10}$$

where h is a discrete latent variable and $\Theta = \{(\theta_y^h, \lambda_y^h)\}$. Then the classification rule is to assign a new observation x to class $\arg\max_y \sum_h \exp(\langle x, \theta_y^h \rangle + \lambda_y^h)$. For learning model parameters Θ, we minimize the negative conditional log-likelihood on $\mathcal{D}_{train}$, namely

$$\min_\Theta \left\{ -\frac{1}{N} \sum_{i=1}^{N} \log P_\Theta(y_i|x_i) \right\}. \tag{6.11}$$

The problem (6.11) takes the form of (6.1) in which the DC function f^i is defined by

$$f^i(\Theta) = \log \sum_{y \in \mathcal{Y}} \sum_h \exp\left(\langle x_i, \theta_y^h \rangle + \lambda_y^h\right) - \log \sum_h \exp\left(\langle x_i, \theta_{y_i}^h \rangle + \lambda_{y_i}^h\right). \tag{6.12}$$

Many other application problems in practice take the form of (6.1) such as sparse support vector machine, deep neutral networks, regularized least squares, sensor networks, other maximum likelihood problems, etc.

For solving the problem (6.1), batch approaches are more natural and well-known. However, when n is large, the above problem becomes more challenge in machine learning. The per-iteration costs of the batch approaches might be more expensive because they use all the functions f^i. In the last few years, stochastic optimization techniques have proven to be useful in machine learning for solving problems with a large number n of training data points (s.g. (Bottou, 2004, 1998; Duchi and Singer, 2009; Xiao, 2010; Mairal, 2013)). The typical stochastic optimization method is the stochastic gradient method. In the context minimizing a differentiable function f without constraint $\Omega = \mathbb{R}^d$, the update step of the stochastic gradient method is given by

$$\Theta_{l+1} \leftarrow \Theta_l - \gamma_l \nabla f^{i_l}(\Theta_l), \tag{6.13}$$

where the index i_l is randomly chosen from $\{1, ..., n\}$ and γ_l is a positive step size. At each iteration, this method only involves the computation of the gradient $\nabla f^{i_l}(\Theta_l)$ corresponding to one sample, hence its per-iteration cost is very cheap. Other works are closely related to our works: online expectation-maximization algorithm (Cappé and Moulines, 2009), stochastic majorization-minimization algorithm (Mairal, 2013), stochastic successive minimization method (Razaviyayn et al., 2016), incremental method (Bertsekas, 2011; Mairal, 2015).

There are some practical and theoretical advantages of stochastic over batch approaches for solving the large-scale machine learning problems. However, in the fact that batch approaches possess some intrinsic advantages. Motivated by this we combine the best properties of the stochastic approach and one of the most powerful tools in optimization. We introduce in this chapter a stochastic scheme based on DCA called stochastic DCA for the problem (6.1) where Ω is a convex set in $\mathbb{R}^d$ and f is a large sum of DC functions f^i. At each iteration, the stochastic DCA requires to solve the convex surrogate problem of only one small subset of the DC functions, which results in obtaining a low computation cost per iteration. Secondly, we exploit the particular structure of the problem and provide two special versions of the stochastic DCA: stochastic proximal DCA and stochastic proximal Newton DCA. In many practical problems, these algorithms can provide an explicit solution at each iteration. In some cases of objective functions, we also point out that the stochastic gradient descent and stochastic proximal descent are special variants of the proposed algorithms. Finally, we apply the stochastic DCA to solve the latent log-linear model (6.11) and propose two DCA for this problem. In order to evaluate the performance of the proposed methods, an empirical experiment is conducted.

The rest of the chapter is organized as follows. In Section 6.2, we present a genetic stochastic DCA and show that the stochastic gradient descent algorithm is as a special version of the stochastic DCA. We illustrate how to apply stochastic DCA to solve the latent log-linear model in Section 6.3. The DCA based algorithms are also proposed in Section 6.4. The numerical experiments are reported in Section 6.5 and Section 6.6 concludes the chapter.

6.2 Solution method based on DCA

6.2.1 Stochastic DCA

In this section, we introduce a stochastic DCA that exploits the structure of the objective f being a large sum of n DC functions $f^i = g^i - h^i$, where g^i and h^i are convex functions. At each iteration l, a small number of functions is used, and an iteration of DCA is performed. Every function f^i is DC function, hence we have a DC decomposition of $\frac{1}{|s_l|}\sum_{i\in s_l} f^i$ as follows:

$$\frac{1}{|s_l|}\sum_{i\in s_l} f^i(\Theta) = \bar{g}^l(\Theta) - \bar{h}^l(\Theta),$$

where $\bar{g}^l$ and $\bar{h}^l$ are convex functions defined by

$$\bar{g}^l = \frac{1}{|s_l|}\sum_{i\in s_l} g^i(\Theta) \quad \text{and} \quad \bar{h}^l = \frac{1}{|s_l|}\sum_{i\in s_l} h^i(\Theta),$$

We propose a stochastic scheme that at each iteration l we randomly choose one small subset of the indexes $s_l \subset \{1,...,n\}$ and compute $v^l \in \partial\bar{h}^l(\Theta_l)$, and then solve the following convex problem:

$$\min_{\Theta\in\Omega}\left\{\bar{g}^l(\Theta) - \langle v^l, \Theta\rangle\right\}.$$

The generic stochastic DCA is described in Algorithm 6.1.

Algorithm 6.1 Generic stochastic DCA

Initialization: Choose $\Theta_0 \in \Omega$.

For $l = 0, 1, ...$ **do**

1. Randomly choose a small subset $s_l \subset \{1,...,n\}$.
2. Compute $v^l \in \partial\frac{1}{|s_l|}\sum_{i\in s_l} h^i(\Theta_l)$.
3. Compute Θ_{l+1} by solving the convex problem:

$$\min_{\Theta\in\Omega}\left\{\frac{1}{|s_l|}\sum_{i\in s_l} g^i(\Theta) - \langle v^l, \Theta\rangle\right\}. \tag{6.14}$$

End for.

Remark 6.1 *We consider an extension of the stochastic DCA (6.1). In the step 3, the update rule of Θ_{l+1} can be replaced with*

- *step 3'.*

$$\Theta_{l+1} \in \arg\min_{\Theta\in\Omega}\left\{\frac{1}{l}\sum_{i=1}^{l}\left[\bar{g}^i(\Theta) - \langle v^i, \Theta\rangle\right]\right\}. \tag{6.15}$$

6.2.2 Special versions of stochastic DCA

In this section, we discuss about several versions of the stochastic DCA base on special DC decompositions of f^i. The proposed algorithms in this section is very useful in practice because the solution to the convex problem (6.14) can be explicitly computed. We also show some existing stochastic algorithms are special versions of our algorithm. We now consider the first DC component g^i expressed as follows:

$$g^i(\Theta) = g_1^i(\Theta) + g_2^i(\Theta), \tag{6.16}$$

where g_1^i and g_2^i are convex functions.

6.2.2.1 Stochastic proximal DCA

In order to discuss the first special version of (6.1), we assume that there exists a positive number ρ_i such that $\frac{\rho_i}{2}\|\Theta\|^2 - g_1^i(\Theta)$ is convex, for example, g_1^i is differentiable with L - Lipschitz gradient. Therefore, we can choose DC components of $\frac{1}{|s_l|}\sum_{i\in s_l} f^i$ by

$$\bar{g}^l(\Theta) = \bar{g}_2^l(\Theta) + \frac{\bar{\rho}_l}{2}\|\Theta\|^2,$$
$$\bar{h}^l(\Theta) = \frac{\bar{\rho}_l}{2}\|\Theta\|^2 - \frac{1}{|s_l|}\sum_{i\in s_l} g_1^i(\Theta) + \frac{1}{|s_l|}\sum_{i\in s_l} h^i(\Theta),$$

where $\bar{g}_2^l(\Theta) = \frac{1}{|s_l|}\sum_{i\in s_l} g_2^i(\Theta)$ and $\bar{\rho}_l = \frac{1}{|s_l|}\sum_{i\in s_l}\rho_i$. Following the generic stochastic DCA 6.1, at each iteration l we have to compute $v^l \in \partial\bar{h}^l(\Theta_l)$ and

$$\Theta_{l+1} = \arg\min_{\Theta\in\Omega}\left\{\bar{g}_2^l(\Theta) + \frac{\bar{\rho}_l}{2}\|\Theta\|^2 - \langle v^l, \Theta\rangle\right\} = \arg\min_{\Theta\in\Omega}\left\{\bar{g}_2^l(\Theta) + \frac{\bar{\rho}_l}{2}\|\Theta - \frac{v^l}{\bar{\rho}_l}\|^2\right\}$$
$$:= \text{prox}_{\bar{\rho}_l}^{\bar{g}_2^l+\chi_\Omega}\left(\frac{v^l}{\bar{\rho}_l}\right),$$

where χ_Ω is defined by $\chi_\Omega(t) = 0$ if $t \in \Omega$ and 0 otherwise, and $\text{prox}_{\bar{\rho}_l}^{\bar{g}_2^l+\chi_\Omega}$ denotes the proximal operator associated to $\bar{g}_2^l + \chi_\Omega$. The first special version of the stochastic DCA is described in Algorithm 6.2. We can call this algorithm as the stochastic proximal DCA.

Algorithm 6.2 Stochastic Proximal DCA (SPDCA)

Initialization: Choose $\Theta_0 \in \Omega$.

For $l = 0, 1, ...$ **do**

1. Randomly choose a small subset $s_l \subset \{1, ..., n\}$.
2. Compute $v^l \in \partial\bar{h}^l(\Theta_l)$.
3. Compute $\Theta_{l+1} = \text{prox}_{\bar{\rho}_l}^{\bar{g}_2^l+\chi_\Omega}\left(\frac{v^l}{\bar{\rho}_l}\right)$.

End for.

Remark 6.2 *We consider a special case in which $g_2^i \equiv 0$, $i = 1, ..., n$, hence $\bar{g}_2^l \equiv 0$. In the step 3 of the algorithm (6.2), the proximal operator $prox_{\bar{\rho}_l}^{\chi_\Omega}\left(\frac{v^l}{\bar{\rho}_l}\right)$ becomes a projection of $\frac{v^l}{\bar{\rho}_l}$ on Ω. When Ω falls into one of the following cases: a simplex, a ball, a box, a hyperplane or the intersection of a box and a hyperplane, this projection is possible to be explicitly computed.*

In many practical problems such as regularization problems, we have $\Omega = \mathbb{R}^d$ and $g_2^i(\Theta) = \lambda\|\Theta\|_1$. The proximal map $prox_{\bar{\rho}_l}^{\lambda\|.\|_1}\left(\frac{v^l}{\bar{\rho}_l}\right)$ can be computed in closed form:

$$prox_{\bar{\rho}_l}^{\lambda\|.\|_1}\left(\frac{v^l}{\bar{\rho}_l}\right) = \mathcal{S}\left(\frac{v^l}{\bar{\rho}_l}, \lambda/\bar{\rho}_l\right),$$

where $\mathcal{S}$ is the elementwise soft-thresholding operator defined by $\mathcal{S}(v, w)_j = sgn(v_j)(|v_j| - w_j)_+$.

6.2.2.2 Stochastic gradient descent and incremental proximal methods are special versions of SPDCA

When $\Omega = \mathbb{R}^d, g_2^i = 0$, g_1^i, h^i are differentiable, and at each iteration we randomly choose one index $s_l = \{i_l\}$, we have

$$v^l = \nabla \bar{h}^l(\Theta_l) = \rho_{i_l}\Theta_l - \nabla f^{i_l}(\Theta_l).$$

Hence, the updated rule in step 3 can be rewritten as follows:

$$\Theta_{l+1} = \Theta_l - \frac{1}{\rho_{i_l}}\nabla f^{i_l}(\Theta_l).$$

This is the updated rule of the stochastic gradient descent algorithm with the step-size $\frac{1}{\rho_{i_l}}$ at the iteration l. Therefore, the stochastic gradient descent algorithm is a special version of SPDCA.

When $h^i \equiv 0$ and at each iteration we randomly choose one index $s_l = \{i_l\}$, we have

$$v^l = \rho_{i_l}\Theta_l - \tilde{\nabla} g_1^{i_l}(\Theta_l),$$

where $\tilde{\nabla} g_1^{i_l}(\Theta_l)$ is an arbitrary subgradient of $g_1^{i_l}$ at Θ_l. Hence, the updated rule in step 3 becomes

$$\Theta_{l+1} = \arg\min_{\Theta\in\Omega}\left\{g_2^{i_l}(\Theta) + \frac{\rho_{i_l}}{2}\|\Theta - \Theta_l + \frac{1}{\rho_{i_l}}\tilde{\nabla} g_1^{i_l}(\Theta_l)\|^2\right\}.$$

This update rule can be rewritten as below.

$$\Theta_{l+1} = \arg\min_{\Theta\in\Omega}\left\{g_2^{i_l}(\Theta) + g_1^{i_l}(\Theta_l) + \langle\tilde{\nabla} g_1^{i_l}(\Theta_l), \Theta - \Theta_l\rangle + \frac{\rho_{i_l}}{2}\|\Theta - \Theta_l\|^2\right\}.$$

This is the update rule of the incremental proximal method proposed in Bertsekas (2011) for solving convex problems. The incremental proximal method is also special version of SPDCA.

6.2.2.3 Stochastic Proximal Newton DCA

The second special version of Algorithm 6.1 comes from the assumption that

$$\frac{1}{2}\Theta^T H_i\Theta - g_1^i(\Theta),$$

is convex for some positive definite matrix H_i. We have a DC decomposition of $\frac{1}{|s_l|}\sum_{i\in s_l} f^i$ as follows:

$$\frac{1}{|s_l|}\sum_{i\in s_l} f^i(\Theta) = \bar{g}^l(\Theta) - \bar{h}^l(\Theta), \tag{6.17}$$

where

$$\bar{g}^l(\Theta) = \bar{g}_2^l(\Theta) + \frac{1}{2}\Theta^T \bar{H}_l \Theta,$$
$$\bar{h}^l(\Theta) = \frac{1}{2}\Theta^T \bar{H}_l \Theta - \frac{1}{|s_l|}\sum_{i\in s_l} g_1^i(\Theta) + \frac{1}{|s_l|}\sum_{i\in s_l} h^i(\Theta),$$

with $\bar{g}_2^l(\Theta) = \frac{1}{|s_l|}\sum_{i\in s_l} g_2^i(\Theta)$ and $\bar{H}_l = \frac{1}{|s_l|}\sum_{i\in s_l} H_i$ is also a positive definite matrix. According to the genetic stochastic DCA scheme, at each iteration l, we have to compute $v^l \in \partial \bar{h}^l(\Theta_l)$ and

$$\begin{aligned}\Theta_{l+1} &= \arg\min_{\Theta\in\Omega}\left\{\bar{g}_2^l(\Theta) + \frac{1}{2}\Theta^T\bar{H}_l\Theta - \langle v^l, \Theta\rangle\right\}\\ &= \arg\min_{\Theta\in\Omega}\left\{\bar{g}_2^l(\Theta) + \frac{1}{2}\left(\Theta - \bar{H}_l^{-1}v^l\right)^T\bar{H}_l\left(\Theta - \bar{H}_l^{-1}v^l\right)\right\}\\ &:= \text{prox}_{\bar{H}_l}^{\bar{g}_2^l+\chi_\Omega}\left(\bar{H}_l^{-1}v^l\right),\end{aligned}$$

where $\text{prox}_{\bar{H}_l}^{\bar{g}_2^l+\chi_\Omega}$ denotes the proximal Newton operator associated to $\bar{g}_2^l + \chi_\Omega$. Hence the second special version of (6.1) is similarly to the algorithm (6.2) in which the step 3 is replaced with the following rule:

$$3.\quad \Theta_{l+1} = \text{prox}_{\bar{H}_l}^{\bar{g}_2^l+\chi_\Omega}\left(\bar{H}_l^{-1}v^l\right).$$

The algorithm using this rule can be named as stochastic proximal Newton DCA. Note that the positive definite matrix $\bar{H}_l$ can be computed by using a quasi-Newton method.

6.3 Application to latent log-linear model

Latent variables have long been used to model observations in various models such as hidden Markov models (see e.g. Juang and Rabiner (1985); Starner and Pentland (1995); Durbin et al. (2002); Kim and Pavlovic (2006); Rabiner (1989)), hidden conditional random fields (see e.g. Lafferty et al. (2001); Gunawardana et al. (2004); Wang et al. (2006); van der Maaten et al. (2011); Quattoni et al. (2007)) and log-linear model incorporated latent variables (see e.g. Deselaers et al. (2012); Heigold et al. (2008); Tsiligkaridis et al. (2013)). Recently, there are some works on structural SVMs with latent variables (see e.g. Yu and Joachims (2009); Ping et al. (2014)). These latent variables models are motivated by numerous applications in areas, for examples, speech recognition, information retrieval, natural language processing, object recognition, gesture recognition, object detection, document-level sentiment classification and link prediction.

In this section, we apply the stochastic DCA for solving the latent log-linear model (6.11) which is an extension of log-linear model by incorporating a latent variable. This problem takes the form of (6.1):

$$\min_{\Theta}\left\{f(\Theta) = \frac{1}{n}\sum_{i=1}^{n} f^i(\Theta)\right\}, \tag{6.18}$$

where

$$f^i(\Theta) = \log \sum_{y\in\mathcal{Y}} \sum_h \exp\left(\langle x_i, \theta_y^h\rangle + \lambda_y^h\right) - \log \sum_h \exp\left(\langle x_i, \theta_{y_i}^h\rangle + \lambda_{y_i}^h\right).$$

Every f^i is DC function, hence the problem (6.18) is a DC programming. A natural DC decomposition of f is

$$f(\Theta) = G(\Theta) - H(\Theta), \tag{6.19}$$

where

$$G(\Theta) = \frac{1}{n}\sum_{i=1}^n \log \sum_{y\in\mathcal{Y}} \sum_h \exp\left(\langle x_i, \theta_y^h\rangle + \lambda_y^h\right),$$
$$H(\Theta) = \frac{1}{n}\sum_{i=1}^n \log \sum_h \exp\left(\langle x_i, \theta_{y_i}^h\rangle + \lambda_{y_i}^h\right).$$

In Tsiligkaridis et al. (2013), the authors have proposed a concave-convex procedure (CCCP) based on this DC decomposition and a proximal term (ProxCCCP) for solving the problem (6.18). Note that CCCP is a special version of DCA.

In this section, we exploit the particular structure of the problem (6.18) and propose a special stochastic DCA for solving it. At each iteration l, we randomly choose a small subset of functions f^i and perform one iteration of DCA. We consider a special DC decomposition of $\frac{1}{|s_l|}\sum_{i\in s_l} f^i$ as follows:

$$\bar{g}^l(\Theta) = \frac{\bar{\rho}_l}{2}\|\Theta\|^2,$$
$$\bar{h}^l(\Theta) = \frac{\bar{\rho}_l}{2}\|\Theta\|^2 - \frac{1}{|s_l|}\sum_{i\in s_l}\left(\log\sum_{y,h}\exp(\langle x_i, \theta_y^h\rangle + \lambda_y^h) - \log\sum_h \exp(\langle x_i, \theta_{y_i}^h\rangle + \lambda_{y_i}^h)\right),$$

are convex function when $\bar{\rho}_l$ is large enough. For estimating $\bar{\rho}_l$, we state the following lemma.

Lemma 6.1 *If* $\bar{\rho}_l \geq \frac{2}{|s_l|}\sum_{i\in s_l}(||x_i||^2 + 1)$ *then* $\bar{h}^l(\Theta)$ *is convex.*

Proof : We have

$$\bar{h}^l(\Theta) = \frac{\bar{\rho}_l}{2}\|\Theta\|^2 - \frac{1}{|s_l|}\sum_{i\in s_l}\log\sum_{y,h}\exp(\langle x_i, \theta_y^h\rangle + \lambda_y^h) + \frac{1}{|s_l|}\sum_{i\in s_l}\log\sum_h \exp(\langle x_i, \theta_{y_i}^h\rangle + \lambda_{y_i}^h).$$

Since the function $\frac{1}{|s_l|}\sum_{i\in s_l}\log\sum_h \exp(\langle x_i, \theta_{y_i}^h\rangle + \lambda_{y_i}^h)$ is convex and the sum of two convex functions is also convex, it sufficient to show that

$$\frac{\bar{\rho}_l}{2}\|\Theta\|^2 - \frac{1}{|s_l|}\sum_{i\in s_l}\log\sum_{y,h}\exp(\langle x_i, \theta_y^h\rangle + \lambda_y^h),$$

becomes convex, i.e., $\bar{\rho}_l$ is greater than the spectral radius of the Hessian matrix of

$$\bar{k}^l(\Theta) := \frac{1}{|s_l|}\sum_{i\in s_l}\log\sum_{y,h}\exp(\langle x_i,\theta_y^h\rangle+\lambda_y^h).$$

We have

$$\nabla^2_{(y,h)^2}\bar{k}^l(\Theta)=\frac{1}{|s_l|}\sum_{i\in s_l}\left(\frac{\exp(\langle x_i,\theta_y^h\rangle+\lambda_y^h)}{\sum_{y',h'}\exp(\langle x_i,\theta_{y'}^{h'}\rangle+\lambda_{y'}^{h'})}-\frac{\exp^2(\langle x_i,\theta_y^h\rangle+\lambda_y^h)}{(\sum_{y',h'}\exp(\langle x_i,\theta_{y'}^{h'}\rangle+\lambda_{y'}^{h'}))^2}\right)(x_i,1)(x_i,1)^T,$$

and let $(B,t)\neq(A,y)$, we also have

$$\nabla_{t,B}\nabla_{y,A}\bar{k}^l(\Theta)=-\frac{1}{|s_l|}\sum_{i\in s_l}\frac{\exp(\langle x_i,\theta_y^A\rangle+\lambda_y^A).\exp(\langle x_i,\theta_t^B\rangle+\lambda_t^B)}{(\sum_{y',A'}\exp(\langle x_i,\theta_{y'}^{A'}\rangle+\lambda_{y'}^{A'}))^2}(x_i,1)(x_i,1)^T.$$

Hence we get the following inequality.

$$\begin{aligned}||\nabla^2\bar{k}^l(\Theta)||_2 &\leq \frac{1}{|s_l|}\sum_{i\in s_l}\left(1+\frac{\sum_{A,y,B,t}\exp(\langle x_i,\theta_y^A\rangle+\lambda_y^A).\exp(\langle x_i,\theta_t^B\rangle+\lambda_t^B)}{(\sum_{y,h}\exp(\langle x_i,\theta_y^h\rangle+\lambda_y^h))^2}\right)||(x_i,1)(x_i,1)^T||_2\\ &\leq \frac{2}{|s_l|}\sum_{i\in s_l}(||x_i||^2+1).\end{aligned}$$

On the other hand, the spectral radius of the Hessian matrix of $\bar{k}^l(\Theta)$ is smaller than $||\nabla^2\bar{k}^l(\Theta))||_2$. It follows that ρ_{s_l} is larger than the spectral radius of the Hessian matrix. The lemma has been proved. □

Remark 6.3 *From the Lemma 6.1, we can choose* $\bar{\rho}_l=\frac{2}{|s_l|}\sum_{i\in s_l}(||x_i||^2+1)$.

Following the special stochastic DCA scheme, at each iteration l we have to randomly choose $s_l\subset\{1,...,n\}$, compute $v^l\in\partial\bar{h}^l(\Theta_l)$ and

$$\Theta_{l+1}=\arg\min_{\Theta}\left\{\frac{\bar{\rho}_l}{2}\|\Theta\|^2-\langle v^l,\Theta\rangle\right\}=\frac{v^l}{\bar{\rho}_l}.$$

Let δ be the function defined by

$$\delta(y,x_i)=\begin{cases}1 & \text{if} \quad y=y_i\\ 0 & \text{otherwise}\end{cases}. \tag{6.20}$$

The computation of v^l is described as follows.

$$\begin{aligned}v^l_{y,h}&=\bar{\rho}_l(\theta_y^h)_l-\frac{1}{|s_l|}\sum_{i\in s_l}\exp(\langle x_i,(\theta_y^h)_l\rangle+(\lambda_y^h)_l)\\ &\times\left(\frac{1}{\sum_{y',h'}\exp(\langle x_i,(\theta_{y'}^{h'})_l\rangle+(\lambda_{y'}^{h'})_l)}-\frac{\delta(y,x_i)}{\sum_{h'}\exp(\langle x_i,(\theta_{y_i}^{h'})_l\rangle+(\lambda_{y_i}^{h'})_l)}\right)(x_i,1).\end{aligned} \tag{6.21}$$

The stochastic DCA for solving (6.18) is described in the following algorithm.

SDCA: Stochastic DCA for solving (6.18)

Initialization: Choose Θ_0.
For $l = 0, 1, ...$ **do**
1. Randomly choose a small subset $s_l \subset \{1, ..., n\}$.
2. Compute $v^l \in \partial \bar{h}^l(\Theta_l)$ using (6.21).
3. Compute $\Theta_{l+1} = \frac{v^l}{\bar{\rho}_l}$.

End for.

Founded on the results for the stochastic gradient descent (see Bottou (1998)), we prove the convergence properties of SDCA.

Theorem 6.1 *Assume that the data is bounded. If* $\sum_{l=1}^{+\infty} \frac{1}{\bar{\rho}_l^2} < +\infty$, *then SDCA generates the sequence* $\{\Theta_l\}_l$ *such that*
i) $\{f(\Theta_l)\}_l$ *converges almost surely.*
ii) $\mathbb{E}\left[\sum_{l=1}^{+\infty} \frac{1}{\bar{\rho}_l} \|\nabla f(\Theta_l)\|^2\right] < +\infty$ *if* $\mathbb{E}_{s_l}[\|\frac{1}{|s_l|}\sum_{i \in s_l} \nabla f^i(\Theta_l)\|^2] \leq A + B\|\nabla f(\Theta_l)\|^2$ *for some* $A \geq 0$ *and* $B > 0$.

Proof : First of all, we prove that the gradient $\nabla f^i(\Theta)$ and Hessian functions $\nabla^2 f^i(\Theta)$ are bounded. We have

$$\nabla_{(y,h)} f^i(\Theta) = \left(\frac{\exp(\langle x_i, \theta_y^h \rangle + \lambda_y^h)}{\sum_{y',h'} \exp(\langle x_i, \theta_{y'}^{h'} \rangle + \lambda_{y'}^{h'})} - \frac{\delta(y, x_i) \exp(\langle x_i, \theta_{y_i}^h \rangle + \lambda_{y_i}^h)}{\sum_{h'} \exp(\langle x_i, \theta_{y_i}^{h'} \rangle + \lambda_{y_i}^{h'})} \right) (x_i, 1),$$

and

$$\begin{aligned}\nabla^2_{(y,h)^2} f^i(\Theta) = &(1 - \frac{\exp(\langle x_i, \theta_y^h \rangle + \lambda_y^h)}{\sum_{y',h'} \exp(\langle x_i, \theta_{y'}^{h'} \rangle + \lambda_{y'}^{h'})}) \frac{\exp(\langle x_i, \theta_y^h \rangle + \lambda_y^h)(x_i, 1)(x_i, 1)^T}{\sum_{y',h'} \exp(\langle x_i, \theta_{y'}^{h'} \rangle + \lambda_{y'}^{h'})} \\ &- \delta(y, x_i)(1 - \frac{\exp(\langle x_i, \theta_{y_i}^h \rangle + \lambda_{y_i}^h)}{\sum_{h'} \exp(\langle x_i, \theta_{y_i}^{h'} \rangle + \lambda_{y_i}^{h'})}) \frac{\exp(\langle x_i, \theta_{y_i}^h \rangle + \lambda_{y_i}^h)(x_i, 1)(x_i, 1)^T}{\sum_{h'} \exp(\langle x_i, \theta_{y_i}^{h'} \rangle + \lambda_{y_i}^{h'})},\end{aligned}$$

$$\begin{aligned}\nabla_{t,B} \nabla_{y,A} f^i(\Theta) = &- \frac{\exp(\langle x_i, \theta_y^A \rangle + \lambda_y^A). \exp(\langle x_i, \theta_t^B \rangle + \lambda_t^B)}{\left(\sum_{y',A'} \exp(\langle x_i, \theta_{y'}^{A'} \rangle + \lambda_{y'}^{A'})\right)^2} (x_i, 1)(x_i, 1)^T \\ &+ \delta(y, x_i)\delta(t, x_i) \frac{\exp(\langle x_i, \theta_{y_i}^A \rangle + \lambda_{y_i}^A). \exp(\langle x_i, \theta_{y_i}^B \rangle + \lambda_{y_i}^B)}{\left(\sum_{A'} \exp(\langle x_i, \theta_{y_i}^{A'} \rangle + \lambda_{y_i}^{A'})\right)^2} (x_i, 1)(x_i, 1)^T,\end{aligned}$$

for $(B,t) \neq (A,y)$. Hence we obtain the following inequalities

$$\|\nabla f^i(\Theta)\|_2 \leq 2\|(x_i,1)\|_2, \tag{6.22}$$

and

$$\begin{aligned}||\nabla^2 f^i(\Theta)||_2 &\leq (2 + \frac{\sum_{y,A,t,B} \exp(\langle x_i, \theta_y^A\rangle + \lambda_y^A).\exp(\langle x_i, \theta_t^B\rangle + \lambda_t^B)}{(\sum_{y,h} \exp(\langle x_i, \theta_y^h\rangle + \lambda_y^h))^2} \\ &+ \frac{\sum_{A,B} \exp(\langle x_i, \theta_{y_i}^A\rangle + \lambda_{y_i}^A).\exp(\langle x_i, \theta_{y_i}^B\rangle + \lambda_{y_i}^B)}{(\sum_h \exp(\langle x_i, \theta_{y_i}^h\rangle + \lambda_{y_i}^h))^2})||(x_i,1)(x_i,1)^T||_2 \\ &\leq 4(||x_i||^2 + 1).\end{aligned}$$

Therefore, we get

$$||\nabla^2 f(\Theta)||_2 \leq \frac{1}{n}\sum_{i=1}^n ||\nabla^2 f^i(\Theta)||_2 \leq \frac{4}{n}\sum_{i=1}^n (||x_i||^2 + 1). \tag{6.23}$$

From the above results and similar arguments in Bottou (1998), the proof of theorem is completed.

□

6.4 DCA for solving the latent log-linear model

In this section, we propose two DCA for solving the problem (6.18). The first DCA bases on the natural DC decomposition (6.19) while the second DCA bases on a special DC decomposition of f. Firstly, the corresponding DC formulation of the problem (6.18) is

$$\min_{\Theta} \{f(\Theta) = G(\Theta) - H(\Theta)\}, \tag{6.24}$$

According to the generic DCA scheme, DCA applied on (6.24) consists of computing, at each iteration l, a gradient $V^l = \nabla H(\Theta_l)$ and solving the convex program of the form (P_l)

$$\min_{\Theta} \left\{G(\Theta) - \langle V^l, \Theta\rangle\right\}. \tag{6.25}$$

The computation of $V^l = \nabla H(\Theta_l)$ is described as follows.

$$V_{y,h}^l = \frac{1}{n}\sum_{i=1}^n \frac{\delta(y,x_i)\exp\left(\langle x_i, (\theta_{y_i}^h)_l\rangle + (\lambda_{y_i}^h)_l\right)}{\sum_{h'}\exp\left(\langle x_i, (\theta_{y_i}^{h'})_l\rangle + (\lambda_{y_i}^{h'})_l\right)}(x_i,1). \tag{6.26}$$

The algorithm is described as follows.

DCA1: DCA for solving (6.24)

Initialization: Let τ tolerance sufficient small, set $l = 0$ and choose Θ_0.
repeat

1. Compute $V^l = \nabla H(\Theta_l)$ by

$$V^l_{y,h} = \frac{1}{n}\sum_{i=1}^{n} \frac{\delta(y, x_i)\exp\left(\langle x_i, (\theta^h_{y_i})_l\rangle + (\lambda^h_{y_i})_l\right)}{\sum_{h'}\exp\left(\langle x_i, (\theta^{h'}_{y_i})_l\rangle + (\lambda^{h'}_{y_i})_l\right)}(x_i, 1).$$

2. Solve the following convex problem to obtain Θ_{l+1}

$$\min_{\Theta}\left\{\frac{1}{n}\sum_{i=1}^{n}\log\sum_{y\in\mathcal{Y}}\sum_{h}\exp\left(\langle x_i, \theta^h_y\rangle + \lambda^h_y\right) - \langle V^l, \Theta\rangle\right\} \tag{6.27}$$

3. $l \leftarrow l + 1$.

until $||\Theta_l - \Theta_{l-1}|| \leq \tau\left(||\Theta_{l-1}|| + 1\right)$ or $|f(\Theta_l) - f(\Theta_{l-1})| \leq \tau\left(|f(\Theta_{l-1})| + 1\right)$.

For solving the convex subproblem (6.27), we use the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization algorithm (Nocedal and Wright, 1999).

We remark that DCA1 might be quite expensive because we have to use an iterative method for solving the convex subproblem (6.27). This motivates us to consider another DC formulation of the problem (6.18) that the corresponding DCA scheme has explicit solution at each iteration. We now consider a special DC decomposition of $f(\Theta)$ as follows.

$$f(\Theta) = G'(\Theta) - H'(\Theta), \tag{6.28}$$

where

$$G'(\Theta) = \frac{\mu}{2}||\Theta||^2 \tag{6.29}$$

$$H'(\Theta) = \frac{\mu}{2}||\Theta||^2 - f(\Theta), \tag{6.30}$$

are convex functions when μ is large enough. Hence we have a DC formulation below.

$$\min_{\Theta}\left\{F(\Theta) = G'(\Theta) - H'(\Theta)\right\}. \tag{6.31}$$

For estimating μ, we state the following lemma.

Lemma 6.2 *If* $\mu \geq \frac{2}{n}\sum_{i=1}^{n}(||x_i||^2 + 1)$ *then* $H'(\Theta)$ *is convex.*

Proof : This lemma is proved analogously to Lemma 6.1 by replacing the subset s_l with the set $\{1, .., n\}$. □

Remark 6.4 *From the Lemma 6.2, we can choose* $\mu = \frac{2}{n}\sum_{i=1}^{n}(||x_i||^2 + 1)$.

Like DCA1, DCA applied on (6.31) consist of computing, at each iteration l, a gradient $V^l = \nabla H'(\Theta_l)$, and then solving the following convex problem to obtain Θ_{l+1}.

$$\min_{\Theta} \left\{ G'(\Theta) - \langle V^l, \Theta \rangle \right\}, \tag{6.32}$$

whose solution is $\Theta_{l+1} = \frac{V^l}{\mu}$. The computation of V^l is described as follows.

$$\begin{aligned} V^l_{y,h} = \mu(\theta^h_y)^l - \frac{1}{n}\sum_{i=1}^{n} \exp(\langle x_i, (\theta^h_y)_l\rangle + (\lambda^h_y)_l) \\ \left(\frac{1}{\sum_{y',h'} \exp(\langle x_i, (\theta^{h'}_{y'})_l\rangle + (\lambda^{h'}_{y'})_l)} - \frac{\delta(y, x_i)}{\sum_{h'} \exp(\langle x_i, (\theta^{h'}_{y_i})_l\rangle + (\lambda^{h'}_{y_i})_l)} \right) (x_i, 1). \end{aligned} \tag{6.33}$$

DCA for solving (6.31) is described as follows.

DCA2: DCA for solving (6.31)

Initialization: Let τ tolerance sufficient small, set $l = 0$, compute μ and choose Θ_0.
repeat
1. Compute $V^l = \nabla H'(\Theta_l)$ using (6.33).
2. Compute $\Theta_{l+1} = \frac{V^l}{\mu}$.
3. $l \leftarrow l + 1$.

until $||\Theta_l - \Theta_{l-1}|| \leq \tau\,(||\Theta_{l-1}|| + 1)$ or $|f(\Theta_l) - f(\Theta_{l-1})| \leq \tau\,(|f(\Theta_{l-1})| + 1)$.

The convergence properties of DCA1 and DCA2 are given in the following theorem.

Theorem 6.2 *(i) DCA1 (resp. DCA2) generates the sequence* $\{\Theta_l\}_l$ *such that* $\{f(\Theta_l)\}_l$ *is decreasing.*
(ii) DCA2 generates the sequence $\{\Theta_l\}_l$ *such that* $\sum_{l=0}^{+\infty} \|\Theta_l - \Theta_{l+1}\|_2^2 < +\infty$ *and* $\|\Theta_l - \Theta_{l+1}\| \to 0$ *as* $l \to +\infty$.
(iii) Every limit point of the sequence generated by DCA1 (resp. DCA2) is a critical point of the problem (6.24) *(resp. the problem* (6.31)*).*

Proof : (i) and (iii) are consequences of convergence properties of general DC programs and the facts that the objective function of (6.18) is bounded from below by 0.

(ii) From the step 1 of DCA2, we have $V^l = \nabla H'(\Theta_l)$. It follows that

$$H'(\Theta_{l+1}) \geq H'(\Theta_l) + \langle V^l, \Theta_{l+1} - \Theta_l \rangle. \tag{6.34}$$

By the step 3 of DCA2, we have $V^l = \mu\Theta_{l+1}$. Substituting this and $G'(\Theta) = \frac{\mu}{2}\|\Theta\|_2^2$ into (6.34), we obtain

$$\begin{aligned}
&H'(\Theta_{l+1}) \geq H'(\Theta_l) + \mu\langle\Theta_{l+1}, \Theta_{l+1} - \Theta_l\rangle \\
\Rightarrow &H'(\Theta_{l+1}) \geq H'(\Theta_l) - \frac{\mu}{2}\|\Theta_l\|_2^2 + \frac{\mu}{2}\|\Theta_{l+1}\|_2^2 + \frac{\mu}{2}\|\Theta_l - \Theta_{l+1}\|_2^2 \\
\Rightarrow &H'(\Theta_{l+1}) \geq H'(\Theta_l) - G'(\Theta_l) + G'(\Theta_{l+1}) + \frac{\mu}{2}\|\Theta_l - \Theta_{l+1}\|_2^2 \\
\Rightarrow &f(\Theta_l) - f(\Theta_{l+1}) \geq \frac{\mu}{2}\|\Theta_l - \Theta_{l+1}\|_2^2.
\end{aligned}$$

Moreover, $\mu > 0$. Hence, we have

$$\|\Theta_l - \Theta_{l+1}\|_2^2 \leq \frac{2}{\mu}\left(f(\Theta_l) - f(\Theta_{l+1})\right). \tag{6.35}$$

Let N be a positive integer. Summing (6.35) from $l = 0$ to N, we get

$$\sum_{l=0}^{N} \|\Theta_l - \Theta_{l+1}\|_2^2 \leq \frac{2}{\mu}\left(f(\Theta_0) - f(\Theta_{N+1})\right). \tag{6.36}$$

On the other hand, we have $f(\Theta_{N+1}) \geq 0$. Combining this and (6.36) we get

$$\sum_{l=0}^{N} \|\Theta_l - \Theta_{l+1}\|_2^2 \leq \frac{2}{\mu} f(\Theta_0).$$

Taking the limit as $N \to +\infty$, we obtain

$$\sum_{l=0}^{+\infty} \|\Theta_l - \Theta_{l+1}\|_2^2 < +\infty, \tag{6.37}$$

and hence $\lim_{l\to+\infty} \|\Theta_l - \Theta_{l+1}\|_2 = 0$. □

6.5 Numerical experiments

We will compare the three proposed algorithms (one stochastic DCA scheme (SDCA) and two DCA schemes (DCA1 and DCA2)) to the two methods that aim to solve the problem (6.11) (L-BFGS (Nocedal and Wright, 1999) and ProxCCCP (Tsiligkaridis et al., 2013)) and another state-of-the-art method for the classification problems (Kernel SVM).

The L-BFGS procedure is a well known optimization technique that directly optimizes the objective function of the problem (6.11). The L-BFGS procedure is performed by the **lbfgs** package of R software.

Table 6.1: Real Datasets.

Datasets	#train	#test	#feature	#class
isolet[1]	6,238	1,559	616	26
usps[2]	7,291	2,007	256	10
ijcnn1[3]	49,990	91,701	22	2
webspam[3]	280,000	70,000	254	2

ProxCCCP is a modification of the concave-convex procedure (CCCP). Note that CCCP is a special version of DCA. ProxCCCP added a proximal term to each convex subproblem in CCCP, namely, at each iteration l, ProxCCCP has to solve the following problem

$$\min_{\Theta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \log \sum_{y \in \mathcal{Y}} \sum_{h} \exp\left(\langle x_i, \theta_y^h \rangle + \lambda_y^h\right) + \langle \nabla E_{cave}(\Theta_l), \Theta \rangle + \frac{c_l}{2} \|\Theta - \Theta_l\|^2 \right\}$$

where $E_{cave}(\Theta) = -\frac{1}{n} \sum_{i=1}^{n} \log \sum_h \exp\left(\langle x_i, \theta_{y_i}^h \rangle + \lambda_{y_i}^h\right)$. In the experiments, we set $c_l = c = 5 \times 10^{-4}$ as suggested in Tsiligkaridis et al. (2013).

Kernel SVM uses the Radial Basis (Gaussian) kernel function. It is included in the **kernlab** package of R software.

We have used four real-world datasets for our comparison, and their information is shown in Table 6.1.

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

We set $\tau = 10^{-5}$ for the stop tolerance of DCA. The starting point $\Theta_0 = \{(\theta_k^h)_0 \in \mathbb{R}^p, (\lambda_k^h)_0) \in \mathbb{R}\}$ of the algorithms is chosen as follows: $(\theta_k^h)_0 = \Sigma_k^{-1}\mu_k$ and $(\lambda_k^h)_0 = \log(n_k/n) - n/2 \log(2\pi|\Sigma_k|) - 1/2\mu_k^T\Sigma_k^{-1}\mu_k$, where Σ_k, μ_k and n_k are respectively the sample covariance matrix, mean vector and number of observations of the class k.

In all experiments, the number of latent variables is set to 10. We fix the number of indexes at each iteration. The test set is used to measure the accuracy of various classifiers trained on the training set.

The computational results of SDCA, DCA1, DCA2, L-BFGS, ProxCCCP and Kernel SVM are given in Table 6.2. We are interested in the accuracy of classifiers as well as the rapidity of the algorithms: the percentage of accuracy of classifiers and the training time in second are reported.

Comments on computational results:

The classification accuracy of SDCA, DCA1 and DCA2 is better than that of the compared algorithms on 3 out of 4 datasets. In comparison between stochastic DCA and

1. https://archive.ics.uci.edu/ml/datasets/ISOLET
2. http://statweb.stanford.edu/~tibs/ElemStatLearn/data.html
3. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Table 6.2: Comparative results of DCA1, DCA2, L-BFGS, ProxCCCP and Kernel SVM. Bold fonts indicate the best results in each row.

		SDCA	DCA1	DCA2	L-BFGS	ProxCCCP	Kernel SVM
ACC (%)	isoslet	93.17	95	**96.02**	94.99	95.02	95.95
	usps	91.85	92.28	93.67	90.88	91.13	**94.12**
	ijcnn1	96.11	95.14	**98.73**	93.51	95.27	95.86
	webspam	**98.72**	97.69	98.25	95.19	96.02	97.5
CPU (s)	isoslet	**4.93**	232.37	156.45	183.62	797.37	161.46
	usps	**2.38**	36.02	28.16	32.89	392.97	33.68
	ijcnn1	**16.42**	97.38	43.22	105.16	197.72	216.5
	webspam	**190.28**	2938.8	1518.68	2483.91	2173.13	6838.54

DCA, DCA1 and DCA2 slightly outperform the stochastic DCA on the first two datasets where the number of observations is not large enough. However, on the remaining two datasets with a large number of observations, DCA1, DCA2 and the stochastic DCA are comparable in term of the classification accuracy. The training time of the stochastic DCA is quite short: less than 191 seconds. SDCA not only obtains the quite good results in term of the classification accuracy, but also takes a short time on the all datasets. More precisely, SDCA runs much faster than the other algorithms, and this can be explained by the fact that this approach leads to the sequence of the convex sub-problems which only use one mall subset of observations and have explicit solutions. The second best algorithm according to the running time is DCA2 in which the explicit solution is computed at each iteration.

6.6 Conclusion

In this chapter, we have introduced a stochastic scheme based DCA for solving large scale parameter estimation problems in which the objective function is a large sum of DC functions. At each iteration, we only use one small subset of the DC functions and run one iteration of the corresponding DC program. As an application, we have investigated the structure of the the latent log-linear model and proposed a stochastic DCA for solving it. At each iteration of this stochastic DCA, we can compute the explicit solution to the convex sub-problem. We have also investigated DC programming and DCA for solving the latent log-linear model. We propose two DC formulation of the latent log-linear model and develop two DCA based algorithms.

The robustness and the effectiveness of our algorithms have been demonstrated through the computational results on real datasets. The nice effect of DC decomposition has been exploited: the second DC decomposition seems to be very suitable since it leads to an efficient, fast and scalable stochastic DCA scheme. In the experimental results, the stochastic DCA have obtained the quite good results in terms of accuracy of classifiers, and have taken the shortest time for training. The second best algorithm in terms of the training time is DCA2 which have achieved the best performance in terms of accuracy of classifier.

In future works, we plan to study the convergence properties of the generic stochastic DCA and its variants. We will also apply the stochastic DCA for solving the other models, especially, sparse SVM, spare logistic regression, sparse matrix factorization.

Chapter 7

Conclusion

In this thesis, we have analyzed how the problems of the sparsity in high dimensional setting and the stochastic learning can be addressed from various aspects including theory, algorithms and applications. The main algorithmic methodologies applied in the thesis are DC (Difference of Convex functions) programming and DCA (DC Algorithms) which are considered one of the state of the arts and powerful tools in optimization.

In the first part of the thesis, we introduced and evaluated a methodology supporting variable selections by using DC programming and DCA. In particular, we have first investigated DC programming and DCA for the sparse Fisher linear discriminant analysis (SFLDA) problem using the ℓ_0-regularization. In order to tackle the ℓ_0-norm, we analyzed DC approximation approaches, and among several existing sparse inducing functions we decided to use the Capped-ℓ_1 and the piecewise exponential concave function. The resulting problems have been formulated as DC programs, and then DCA have been applied. Consequently, we have proposed two DCA schemes for two different formulations of a common model to both the approximation functions. The robustness and effectiveness of our DCA based algorithms have been demonstrated through the experiments conducted on both the simulated and real datasets, in which we compared our approaches with three standard algorithms that use the ℓ_1-regularization.

Next, we concentrated on investigating alternating schemes based on DC programming and DCA for solving the sparse optimal scoring (SOS) problem. By using two DC approximations of the ℓ_0-norm and considering two DC formulations of each resulting approximate SOS problem, we have proposed alternating schemes for solving the four approximate problems. In accordance, four DCA schemes have been studied to deal with DC programs w.r.t w_k in each step of the alternating algorithms. The important point here, is that we have proved that the main algorithms converge to a critical point of the approximate problems. The efficiency of the four proposed methods have been compared with five standard algorithms which use the ℓ_1 regularization. The computational results have showed that the proposed algorithms have produced much better sparsity as well as higher classification accuracy than the standard algorithms on both simulated datasets and high-dimensional real datasets. Besides, we have provided recommendations on how

to best use the proposed algorithms. We have also compared the proposed methods for SOS, SFLDA problems with the sparse multiclass support vector machine (SMSVM) using the ℓ_0-regularization. Numerical results have showed that the proposed methods have outperformed SMSVM.

Our third contribution lies in the investigation of the sparse covariance matrix estimation (SCME) problem. Specifically, the capped-ℓ_1 and piecewise exponential concave functions are continued to be chosen for modeling the sparsity, however we face the difficulty in the non-convexity of the negative log-likelihood function. Thus, we have proposed two DC formulations of the approximate SCME problem based on two DC decompositions of its objective function to overcome the problem. The first results are obtained from a natural DC decomposition while the second is introduced to exploit nice effects of DC decompositions. It turns out that the complexity of two corresponding DCA schemes is significantly different and the ratio of gain between them in terms of CPU times in our numerical experiments is up to 44 times. This is explained by the fact that the convex subproblems in the second DCA scheme can be solved by an extremely inexpensive algorithm. Applying DCA on two DC formulations with two approximations, we then have four DCA based algorithms for the approximate SCME problem. Special convergence analysis results of our algorithms have been provided. Additionally, we have considered two important applications of the SCME problem in our experiments, which are respectively the quadratic discriminant analysis using sparse covariance matrices estimated by the proposed algorithms and the portfolio optimization problem. Numerical experiments have been carefully achieved on several test experiments on both simulated datasets and real datasets with eleven algorithms including seven state-of-the-art methods and the four proposed DCA schemes.

In the second part of this thesis, we turned our attention to the problem of group variable selection. We have studied the $\ell_{p,0}$ regularization ($p \geq 1$) for enforcing group sparsity. Using a DC approximation of the $\ell_{p,0}$-norm, we have indicated that the approximate problem is equivalent to the original problem with suitable parameters. By considering two equivalent formulations of the approximate problem we have developed DCA based algorithms to solve them. Among $\ell_{p,0}$ regularizations, we have show that the $\ell_{1,0}$ is the most interesting regularization with several advantages in both theoretical and computational aspects. Regarding applications, we have implemented the proposed algorithms for group variable selection in optimal scoring problem and multiple covariance matrices estimation problem. In the first application, sparsity is obtained by using the $\ell_{p,0}$ regularization that can select the same variables in all discriminant vectors. The resulting sparse discriminant vectors have provided a more interpretable low-dimensional representation of data. In the second application where multiple covariance matrices share some common structures such as the locations or weights of non-zero elements, we have combined the ℓ_0-norm and the $\ell_{p,0}$-norm to enforce sparsity on each covariance matrix and across multiple covariance matrices, respectively.

Finally, we analyzed and applied the stochastic technique based DC programming and DCA to large scale parameter estimation problems in which the objective function is a large sum of DC functions. At each iteration, we only use one small subset of the DC

functions and run one iteration of the corresponding DC program. We have also presented two special versions of the stochastic DCA: stochastic proximal DCA and stochastic proximal Newton DCA that regard some standard stochastic algorithms as special versions. As application, we have investigated the structure of the the latent log-linear model and proposed a special stochastic DCA in which the solution to each convex sub-problem can be explicitly computed. We have also taken DC programming and DCA into account in order to solve the latent log-linear model.

This thesis has explored some issues relating to modeling sparsity and stochastic learning, and we believe several follow-up studies for the future can be derived from this research. First of all, concerning variable selection, the DCA based approaches presented in this thesis could be useful to develop efficient algorithms for other sparse optimization problems in high-dimensional setting. Moreover, in the scope of this thesis, we have just studied the Fisher's discriminant problem for the linear classification, so it is interesting to extend the proposed techniques to more complex settings, such as the case where the observations from each class are drawn from a mixture of Gaussian distributions resulting in nonlinear separations between classes. We also plan to study more extensive applications of these problems.

Regarding group variable selection, we believe that the success of the $\ell_{p,0}$-regularization motivate and open up a new avenue for the group variable selection problems. To be more specific, we will study this regularization to other models such multiclass support vector machine, principal component analysis, compressed sensing, etc. By considering a common DC approximation of the $\ell_{p,0}$-norm, we also intend to investigate the consistency between global minimums (local minimums) of approximate and original problems. Moreover, the combination of the ℓ_0-norm and the $\ell_{p,0}$-norm to obtain the sparsity at both group and individuals in group levels will be more explored.

Last but not least, our research study on the stochastic schemes based on DCA is just the beginning of the ongoing work. Therefore, in the future, the convergence properties of the generic stochastic DCA and its variants will be investigated along with the update rule (6.15). More concretely, we plan to study the stochastic DCA using this update rule and others, as well as explore the use of the stochastic DCA for solving the other models, especially, sparse SVM, spare logistic regression, sparse matrix factorization, group variable selection in latent log-linear model.

Appendix A

Appendix

A.1 Bounded optimal solution set of the problem (5.29)

Let $\mathcal{P}$ be the optimal solution set of the problem (5.29). In the sequel, $x^1, x^2, ..., x^d$ denote the columns of the data matrix X. We will prove that $\mathcal{P}$ is bounded.

Lemma A.1 *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 (5.29). The proof of Lemma A.1 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 A.1 *The optimal solution set of the problem (5.29) 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 A.1, if W is an optimal solution to the problem (5.29), 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 (5.29), 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 A.1 has been proved. □

Bibliography

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96(12):6745–6750.

Argyriou, A., Evgeniou, T., and Pontil, M. (2008). Convex multi-task feature learning. *Machine Learing*, 73(2):243–272.

Banerjee, O., Elghaoul, L. E., and D'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. Mach. Learn. Res.*, 9:485–516.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.*, 2:183–202.

Bertsekas, D. P. (2011). Incremental proximal methods for large scale convex optimization. *Mathematical Programming*, 129(2):163–195.

Bhatia, R. (1997). *Matrix Analysis*. Springer-Verlag New York.

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

Bickel, P. J. and Levina, E. (2004). Some theory for fisher's linear discriminant function, naive bayes, and some alternatives when there are many more variables than observations. *Bernoulli*, 10(6):989–1010.

Bien, J. and Tibshirani, R. (2011). Sparse estimation of a covariance matrix. *Biometrika*, 98(4):807–820.

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

Bottou, L. (1998). *Online Learning in Neural Networks*, chapter Online Learning and Stochastic Approximations, pages 9–42. Cambridge University Press, New York, NY, USA.

Bottou, L. (2004). *Stochastic Learning*, pages 146–168. Springer Berlin Heidelberg, Berlin, Heidelberg.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, P. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundat. Trends Mach. Learn.*, 3(1):1–122.

Boyd, S. and Vanderberghe, L. (1979). *Convex Optimization*. Cambridge University Press, Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, Sao Paolo, Delhi.

Bradley, P. S. and Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In *Proceeding of international conference on machine learning ICML'98*.

Cai, T. and Liu, W. (2011). A direct estimation approach to sparse linear discriminant analysis. *Journal of the American Statistical Association*, 106(496):1566–1577.

Cai, T., Liu, W., and Luo, X. (2011). A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106:594–607.

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

Cappé, O. and Moulines, E. (2009). Online expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society. Series B*, 71(3):593–613.

Chambolle, A., Devore, R. A., Lee, N. Y., and Lucier, B. J. (1998). Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEETrans Image Process*, 7:319–335.

Chaudhuri, S., Drton, M., and Richardson, T. S. (2007). Estimation of a covariance matrix with zeros. *Biometrika*, 94:199–216.

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

Chen, X., Xu, F. M., and Ye, Y. (2010). Lower bound theory of nonzero entries in solutions of l2-lp minimization. *SIAM J. Sci. Comp.*, 32(5):2832–2852.

Chun, H. and Keles, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society - Series B*, 72(1):3–25.

Clemmensen, L., Hansen, M., Ersboll, B., and 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., and Ersbøll, B. (2011). Sparse discriminant analysis. *Technometrics*, 53(4):406–413.

Collobert, R., Sinz, F., Weston, J., and Bottou, L. (2006). Trading convexity for scalability. In *Proceedings of the 23rd international conference on Machine learning*, pages 201–208, NY, USA.

Cotter, S. F., Rao, B. D., Engan, K., and Kreutz-Delgado, K. (2005). Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, 53:2477–2488.

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

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc Series B Stat Methodol*, 39:1–38.

Deng, X. and Tsui, K. W. (2013). Penalized covariance matrix estimation using a matrix-logarithm transformation. *Journal of Computational and Graphical Statistics*, 22(2):494–512.

Deselaers, T., Gass, T., Heigold, G., and Ney, H. (2012). deslat. *Latent Log-linear Models for Handwritten Digit Classification*, 34(6):1105–1117.

Dobra, A., Hans, C., Jones, B., Nevins, J. R., Yao, G., and West, M. (2004). Sparse graphical models for exploring gene expression data. *J. Multiv. Anal.*, 90:196–212.

Duchi, J. and Singer, Y. (2009). Efficient online and batch learning using forward backward splitting. *J. Mach. Learn. Res.*, 10:2899–2934.

Durbin, R., Eddy, S., Krogh, A., and Mitchenson, G. (2002). *Biological Sequence Analysis*. Cambridge University Press.

Efron, B. (2010). Correlated z-values and the accuracy of large-scale statistical estimates. *J. Am. Statist. Ass.*, 105:1042–1055.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Stat. Ass.*, 96(456):1348–1360.

Fan, J., Liao, Y., and Mincheva, M. (2013). Large covariance estimation by thresholding principal orthogonal complements. *J. R. Statist. Soc.B*, 74:603–680.

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

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

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

Friedman, J., Hastie, T. J., and Tibshirani, R. J. (2008). Sparse inverse covariance estimationwith the graphical lasso. *Biostatistics*, 9:432–441.

Gasso, G., Rakotomamonjy, A., and Canu, S. (2009). Recovering sparse signals with a certain family of nonconvex penalties and DC programming. *IEEE Trans. Sign. Proc.*, 57:4686–4698.

Gordon, G. J., Jensen, R. V., Hsiao, L. L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., Richards, W. G., Sugarbaker, D. J., and Bueno, R. (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gege expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62:4963–4967.

Grosenick, L., Greer, S., and Knutson, B. (2008). Interpretable classifers for fmri improve prediction of purchases. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(6):539–547.

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

Gunawardana, A., Mahajan, M., Acero, A., and Platt, C. J. (2004). Hidden conditional random fields for phone classification. In *International Conference on Speech Communication and Technology*.

Guo, J., Levina, E., Michailidis, G., and Zhu, J. (2011). Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15.

Guo, Y., Hastie, T., and Tibshirani, R. (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1):86–100.

Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized discriminant analysis. *The annals of statistics*, 23(1):73–102.

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

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Verlag, New York.

Heigold, G., Lehnen, P., Schlueter, R., and Ney, H. (2008). On the equivalence of gaussian and log-linear hmms. In *Proc. Inter-speech*.

Huang, F. and Chen, S. (2015). Joint learning of multiple sparse matrix gaussian graphical models. *IEEE Transactions on Neural Networks and Learning Systems*, 26(11):2606–2620.

Huang, J., Breheny, P., and Ma, S. (2012). A selective review of group selection in high-dimensional models. *Statist Sci.*, 27.

Jagannathan, R. and Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. *Journal of Finance*, 58:1651–1684.

Juang, B. H. and Rabiner, L. R. (1985). A probabilistic distance measure for hidden markov models. *AT&T Tech. J.*, 64:391–408.

Kha, Z., Shafait, F., and Mian, A. (2015). Joint group sparse pca for compressed hyperspectral imaging. *IEEE Trans. Ima. Process.*, 24(12):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., and Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using expression profiling and artificial neural networks. *Nat. Med.*, 7:673–679.

Kim, M. and Pavlovic, V. (2006). Discriminative learning of mixture of bayesian network classifiers for sequence classification. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 268–275, New York.

Kourtis, A., Dotsis, G., and Markellos, R. N. (2012). Parameter uncertainty in portfolio selection: Shrinking the inverse covariance matrix. *Journal of Banking & Finance*, 36:2522–2531.

Krause, N. and Singer, Y. (2004). Leveraging the margin more carefully. In *Proceedings of the twenty first international conference on Machine learning*, NY, USA.

Krzanowski, W., Jonathan, P., Mccarthy, W., and Thomas, M. (1995). Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. *Journal of the Royal Statistical Society*, 44(1):101–115.

Lafferty, J., Mccallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Lai, T. L., Xing, H., and Chen, Z. (2011). Mean–variance portfolio optimization when means and covariances are unknown. *The Annals of Applied Statistics*, 5:798–823.

Lam, C. and Fan, J. (2009). Sparsistency and rates of convergence in large covariance matrix estimation. *The Annals of Statistics*, 37:4254–4278.

Lan, X., Ma, A., Yuen, P., and Chellappa, R. (2015). Joint sparse representation robust feature-level fusion for multi-cue visual tracking. *IEEE Trans Image Process*, 24(12):5826–5841.

Le Hoai, M., Le Thi, H. A., Pham Dinh, T., and Huynh, V. N. (2013). Block clustering based on difference of convex functions (dc) programming and dc algorithms. *Neural Computation*, 25:259–278.

Le Thi, H. (1994). *Analyse numérique des algorithmes de l'optimization DC. Approches locale et globale. Codes et simulations numériques en grande dimension. Appplications.* PhD thesis, Université de Rouen.

Le Thi, H. and Nguyen, M. (2013). Efficient algorithms for feature selection in multi-class support vector machine. *Advanced Computational Methods for Knowledge Engineering, Studies in Computational Intelligence 479, Springer*.

Le Thi, H. A. (2000). An efficient algorithm for globally minimizing a quadratic function under convex quadratic constraints. *Math. Program.*, 87:401–426.

Le Thi, H. A., Le Hoai, M., Nguyen, V. V., and 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(3):259–278.

Le Thi, H. A., Le Hoai, M., and Pham Dinh, T. (2007). Optimization based DC programming and DCA for hierarchical clustering. *European Journal of Operational Research*, 183:1067–1085.

Le Thi, H. A., Le Hoai, M., and 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., Le Hoai, M., and Pham Dinh, T. (2014b). New and efficient DCA based algorithms for minimum sum-of-squares clustering. *Patter Recognition*, 47:388–401.

Le Thi, H. A. and Nguyen, M. C. (2014). Self-organizing maps by difference of convex functions optimization. *Data Mining and Knowledge Discovery*, 28:1336–1365.

Le Thi, H. A., Nguyen, V. V., and Ouchani, S. (2009). Gene selection for cancer classification using dca. *Journal of Frontiers of Computer Science and Technology*, 3:612–620.

Le Thi, H. A. and Pham Dinh, T. (1998). A branch and bound method via d.c. optimization algorithms and ellipsoidal technique for box constrained nonconvex quadratic problems. *Journal of Global Optimization*, 13(2):171–206.

Le Thi, H. A. and 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., and Huynh, V. N. (2012). Exact penalty and error bounds in DC programming. *Journal of Global Optimization*, 52(3):509–535.

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

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

Le Thi (Website), H. A. Dc programming and dca. http://www.lita.univ-lorraine.fr/~lethi/index.php/dca.html. Accessed in August 2016.

Ledoit, O. and Wolf, M. (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10:603–621.

Ledoit, O. and Wolf, M. (2004). Honey, i shrunk the sample covariance matrix. *J. Port-folio Management*, 30:110–119.

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

Lee, W. and Liu, Y. (2015). Joint estimation of multiple precision matrices with common structures. *Journal of Machine Learning Research*, 16:1035–1062.

Leek, J. and Storey, J. (2008). A general framework for multiple testing dependence. *Proc. Natn. Acad. Sci. USA*, 105:18718–18723.

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

Liu, H., Palatucci, M., and Zhang, J. (2009a). Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *ICML*.

Liu, H., Wang, L., and Zhao, T. (2014). Sparse covariance matrix estimation with eigenvalue contraints. *Journal of Computational and Graphical Statistics*, 23(2):439–459.

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

Liu, Y., Shen, X., and Doss, H. (2005). Multicategory ψ-learning and support vector machine: Computational tools. *Journal of Computational and Graphical Statistics*, 14:219–236.

Mai, Q. and Zou, H. (2013). A note on the connection and equivalence of three sparse linear discriminant analysis methods. *Technometrics*, 55(2):243–246.

Mai, Q., Zou, H., and Yuan, M. (2012). A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika*, 99(1):29–42.

Mairal, J. (2013). Stochastic majorization-minimization algorithms for large-scale optimization. In *NIPS 2013*.

Mairal, J. (2015). Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. Academic Press, London - New York - Toronto - Sydney - San Francisco.

Markowitz, H. (1952). Portfolio selection. *Journal of Finance*, 7:77–91.

Meinshausen, N. and Buhlmann, P. (2006). High dimensional graphs and variable selection with the lasso. *Ann. Statist.*, 34:1436–1462.

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

Nakayama, R., Nemoto, T., Takahashi, H., Ohta, T., Kawai, A., Yoshida, T., Toyama, Y., Ichikawa, H., and Hasegama, T. (2007). Gene expression analysis of soft tissue sarcomas: characterization and reclassification of malignant fbrous histiocytoma. *Modern Pathology*, 20(7):749–759.

Neumann, J., Schnorr, G., and Steidl, G. (2005). Combined svm-based feature selection and classification. *Machine Learning*, 61:129–150.

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

Nocedal, J. and Wright, S. (1999). *Numerical Optimization*. Springer, New York, NY, USA.

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

Obozinski, G., Taskar, B., and Jordan, M. (2010). Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252.

Ong, C. and Le Thi, H. A. (2013a). Learning sparse classifiers with difference of convex functions algorithms. *Optimization Methods and Software*, 28(4):830–854.

Ong, C. S. and Le Thi, H. A. (2013b). Learning sparse classifers with difference of convex cunctions algorithms. *Optimization Methods and Software*, 28(4):830–854.

Peleg, D. and Meir, R. (2008). A bilinear formulation for vector sparsity optimization. *Signal Processing*, 88(2):375–389.

Pham Dinh, T. and Le Thi, H. A. (1997). Convex analysis approach to D.C. programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355.

Pham Dinh, T. and Le Thi, H. A. (1998). A DC optimization algorithm for solving the trust-region subproblem. *SIAM Journal of Optimization*, 8(2):476–505.

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

Ping, W., Liu, Q., and Ihler, A. (2014). Marginal structured svm with hidden variables. In *ICML*.

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

Quattoni, A., Wang, S., Morency, L. P., Collins, M., and Darrell, T. (2007). Hidden conditional random fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(10):1848–1852.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Razaviyayn, M., Sanjabi, M., and Luo, Z.-Q. (2016). A stochastic successive minimization method for nonsmooth nonconvex optimization with applications to transceiver design in wireless communication networks. *Mathematical Programming*, 157(2):515–545.

Rothman, A. (2012). Positive definite estimators of large covariance matrices. *Biometrika*, 99:733–740.

Rothman, A., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electron. J. Statist.*, 2:494–515.

Rothman, A. J., Levina, E., and Zhu, J. (2009). Generalized thresholding of large covariance matrices. *J. Am. Statist. Assoc.*, 104:177–186.

Schafer, J. and Strimmer, K. (2005a). An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21:754–764.

Schafer, J. and Strimmer, K. (2005b). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4:1–30.

Sentana, E. (2009). The econometrics of mean-variance efficiency tests: a survey. *Econometr. J.*, 12:65–101.

Shao, J., Wang, Y., Deng, X., and Wang, S. (2011). Sparse linear discriminant analysis by thresholding for high dimensional data. *Ann. Statist.*, 39(2):1241–1265.

Starner, T. and Pentland, A. (1995). Real-time american sign language recognition from video using hidden markov models. In *International Symposium on Computer Vision*, pages 265–270.

Sun, J. and Zhao, H. (2015). The application of sparse estimation of covariance matrix to quadratic discriminant analysis. *BMC Bioinformatics*, 16:1–9.

Sun, L., Hui, A., Su, Q., Vortmeyer, A., Kotliarov, Y., Pastorino, S., Passaniti, A., Menon, J., Wlling, J., Bailey, R., Rosenblum, M., Mikkelsen, T., and Fine, H. (2006). Neuronal and glioma-derived stem cell factor induces angiogenesis within the brain. *Cancer Cell*, 9:287–300.

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

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc.*, 58:267–288.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl. Acad. Sci.*, 99:6567–6572.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2003). Class prediction by nearest shrunken centroids, with applications to DNA microarrays. *Statistical Science*, 18(1):104–117.

Toh, H. and Horimoto, K. (2002). Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics*, 18:287–297.

Trendafilov, N. T. and Jolliffe, I. T. (2007). Dalass: Variable selection in discriminant analysis via the lasso. *Computational Statistics and Data Analysis*, 51:3718–3736.

Tsiligkaridis, T., Marcheret, E., and Goel, V. (2013). A difference of convex functions approach to large-scale log-linear model estimation. *IEEE Transaction on audio, Speech, and Language processing*, 21(11):2255–2266.

van der Maaten, L., Welling, M., and Saul, L. (2011). Hidden-unit conditional random fields. In *Artificial Intelligence & Statistics*, pages 479–488.

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

Wang, S., Quattoni, A., Morency, L., Demirdjian, D., and Darrell, T. (2006). Hidden conditional random fields for gesture recognition. *Computer Vision and Pattern Recognition*, 2:1521–1527.

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

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

Weston, J. and Watkins, C. (1999). Support vector machines for multi-class pattern recognition. In *Proceedings of European symposium on artifical neural networks, computational intelligence and machine learning*, pages 219–224.

Witten, D. and Tibshirani, R. (2011). Penalized classification using Fisher's linear discriminant. *Journal Royal Statistical Society B*, 73:753–772.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534.

Wu, M., Zhang, L., Wang, Z., Christiani, D., , and Lin, X. (2009). Sparse linear discriminant analysis for simultaneous testing for the significance of a gene set/pathway and gene selection. *Bioinformatics*, 25.

Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.*, 11:2543–2596.

Xiong, H., Goulding, E., Carlson, E. J., Tecott, L., McCulloch, C. E., and Sen, S. (2011). A flexible estimating equations approach for mapping function-valued traits. *Genetics*, 189:305–316.

Xu, P., Brock, G. N., and Parrish, R. S. (2009). Modified linear discriminant analysis approaches for classification of high-dimensional microarray data. *Computational Statistics and Data Analysis*, 53:1674–1687.

Xue, L., Ma, S., and Zuo, H. (2012). Positive-definite ℓ_1-penalized estimation of large covariance matrices. *Journal of the American Statistical Association*, 107:1480–1491.

Yap, J. S., Fan, J., and Wu, R. (2009). Nonparametric modeling of longitudinal covariance structure in functional mapping of quantitative trait loci. *Biometrics*, 65:1068–1077.

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., and al. (2002). Classification, subtype discovery, and prediction of outcome in pediatric lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1:133–143.

Yu, C. and Joachims, T. (2009). Learning structural svms with latent variables. In *ICML*, pages 1169–1176.

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

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika*, 94:19–35.

Yuille, A. L. and Rangarajan, A. (2003). The concave-convex procedure (CCCP). *Neural Comput*, 15:915–936.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.*, 38(2):894–942.

Zhang, T. and Zou, H. (2014). Sparse precision matrix estimation via lasso penalized d-trace loss. *Biometrika*, 101:103–120.

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

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