Published in final edited form as:

Stat Anal Data Min. 2016 April ; 9(2): 75–88. doi:10.1002/sam.11300.

Composite large margin classifiers with latent subclasses for heterogeneous biomedical data

Guanhua Chen,

Assistant Professor, Department of Biostatistics, Vanderbilt University, Nashville, TN 37203

Yufeng Liu,

Professor, Department of Statistics and Operations Research, Department of Genetics, and Department of Biostatistics, University of North Carolina, Chapel Hill, NC 27599

Dinggang Shen, and

Professor, Department of Radiology and BRIC, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599; he is also with Department of Brain and Cognitive Engineering, Korea University, Seoul 02841, Republic of Korea

Michael R. Kosorok

W. R. Kenan, Jr. Distinguished Professor and Chair, Department of Biostatistics, and Professor, Department of Statistics and Operations Research, University of North Carolina, Chapel Hill, NC 27599

Guanhua Chen: g.chen@vanderbilt.edu; Yufeng Liu: yfliu@email.unc.edu; Dinggang Shen: dgshen@med.unc.edu; Michael R. Kosorok: kosorok@unc.edu

Abstract

High dimensional classification problems are prevalent in a wide range of modern scientific applications. Despite a large number of candidate classification techniques available to use, practitioners often face a dilemma of choosing between linear and general nonlinear classifiers. Specifically, simple linear classifiers have good interpretability, but may have limitations in handling data with complex structures. In contrast, general nonlinear classifiers are more flexible, but may lose interpretability and have higher tendency for overfitting. In this paper, we consider data with potential latent subgroups in the classes of interest. We propose a new method, namely the Composite Large Margin Classifier (CLM), to address the issue of classification with latent subclasses. The CLM aims to find three linear functions simultaneously: one linear function to split the data into two parts, with each part being classified by a different linear classifier. Our method has comparable prediction accuracy to a general nonlinear classifier, and it maintains the interpretability of traditional linear classifiers. We demonstrate the competitive performance of the CLM through comparisons with several existing linear and nonlinear classifiers by Monte Carlo experiments. Analysis of the Alzheimer's disease classification problem using CLM not only provides a lower classification error in discriminating cases and controls, but also identifies subclasses in controls that are more likely to develop the disease in the future.

Keywords

Classification; Large margin; Latent subclasses; Principal component analysis

1. Introduction

In biomedical research, it is important to distinguish the patients with a high risk of developing a certain disease from the patients with a low risk of that disease using biomarkers, as the follow-up treatment plan likely depends on such diagnosis. For example, Alzheimer's disease (AD) is one of the most common mental diseases that causes memory, thinking, and behavior problems. Between normal aging and Alzheimer's disease, there exists a transitional stage, known as amnestic mid cognitive impairment (MCI). Although AD is not currently curable, proper early therapy can slow the progress and alleviate symptoms. Thus, it is vital to accurately diagnose AD, especially for MCI. To discriminate AD from normal aging, we can build classifiers with biomarkers coming from various resources such as microarray or imaging data (fMRI).

In the literature, there are a large number of classifiers available, for example, linear discriminant analysis [4], Support Vector Machines (SVMs) [29], Random Forests [1], Distance Weighted Discrimination (DWD) [21], and Large Margin Unified Machines (LUMs) [19]. Hastie et al. [8] provides a comprehensive review of many machine learning techniques. Among various classification tools, linear classifiers are popular especially for high dimensional problems, due to their simplicity and good interpretability. While widely used, linear classifiers can be suboptimal for many practical problems [10]. One main practical problem we are interested in this paper is classification in the presence of latent subgroups. For the AD problem, there are two subgroups for MCI patients: the group that developed into AD in the follow-up and the other group that did not develop AD [24]. However, such subgroup information is latent (unknown) when only the baseline information is available. Another important biological application is the cancer subtype classification. The existing cancer subtypes reported in the literature may be further refined by the profile of high-throughput data, e.g. further subtypes identified by genomic data [25]. For both examples mentioned above, the latent subclasses have biological or clinical interpretation. In practice, such refined class labels are typically unavailable and they correspond to the latent subclasses. The main contribution of this paper is to develop new classification techniques which can handle classification of heterogeneous data with potential latent subclasses. As a remark, we want to point out the difference of latent subgroups considered in this paper from the batch effect in biology [16], which occurs as a result of different laboratory conditions and is independent of the biological or scientific variables in the study [14]. For such batch effects, the batch labels are often available and one can remove such effect directly before performing further analysis such as classification.

To further illustrate the problem of interest, we show a simple two dimensional toy example in Figure 1. This is a binary classification problem with X_1 and X_2 as predictors. One class is labeled in gray, the other is labeled in black. As seen from the plots, each class has two latent subclasses. A linear SVM model is fitted to the data, and its decision boundary is shown as the solid line in Figure 1(a). The linear SVM model completely fails to classify the two classes. Note that although traditional linear classifiers, such as the linear SVM, are not able to effectively capture the difference between classes in this example, the classification task becomes much easier if we divide the data by the line $X_2 = 0$ and classify each part separately. This motivates us to introduce the idea of a splitting function to divide the data

into two parts so that we can handle the classification task with two separate linear classifiers.

As mentioned earlier, despite having good properties, such as simplicity and interpretability of linear classifiers, they are insufficient for problems with nonlinear decision boundaries. Using the kernel trick with a nonlinear kernel function (see [8] for details), one can extend a linear large margin classifier to a non-linear classifier with more flexible classification boundaries. However, the corresponding functional space is much larger, and there is high risk of overfitting. Although regularization is commonly used to control overfitting, finding the optimal tuning parameters is difficult for nonlinear kernel classifiers, especially for high dimensional data. Furthermore, compared to simple linear classifiers, results from non-linear classifiers are generally more difficult to interpret. As shown in Figure 1, two non-linear classifiers, quadratic (panel b) and Gaussian (panel c) kernel SVMs work reasonably well. But their decision boundaries are quite complicated. Moreover, we will show in Section 4 that the performance of kernel SVMs can deteriorate rapidly when the dimension increases.

To solve the classification problem with complex structures, we propose a new group of methods, namely the Composite Large Margin Classifier (CLM). The CLM aims to find three linear functions simultaneously: one linear function to split the data into two parts, with each part being classified by a different linear classifier. We denote these three linear functions as $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$, respectively. Because of the split function, the CLM method provides a natural solution to the classification problem with latent subgroups. In Figure 1(d), we present the three lines: $\widehat{f_1}(\mathbf{x})=0$, $\widehat{f_2}(\mathbf{x})=0$ and $\widehat{f_3}(\mathbf{x})=0$ estimated from CLM. The function $\widehat{f_1}(\mathbf{x})$ helps to capture the hidden structure and divide the data into two parts, one part with $\widehat{f_1}(\mathbf{x})>0$ and the other part with $\widehat{f_1}(\mathbf{x})<0$. With this division, we can use two separate linear classifiers on each part. Thus, our CLM method makes use of three linear functions simultaneously to classify the data and capture the latent subclasses.

The CLM method has some advantages over both traditional linear and nonlinear methods. Compared to linear methods, the CLM is more flexible for classifying data with complex structure. On the other hand, unlike general nonlinear methods, the CLM only depends on the linear combination of the features and thus retains most of the simplicity and interpretability of linear methods. Furthermore, the functional space of interest for the CLM is much smaller than the space for general kernel methods. As a consequence of its relative simplicity, the CLM performs better than the kernel based method in high dimensions as shown in Section 4. In addition, the splitting function for the CLM has a natural latent variable interpretation, and it is also a change-plane problem – an extension of the well studied change-point problem [3] and the recent change-line problem [12]. The latent variable identified by our methods in the AD example appears to be scientifically meaningful and is useful for disease prognosis predication and treatment selection.

Although our CLM method is motivated by large margin classifiers, the fundamental concept is more general and is applicable to many other linear classifiers as well. Furthermore, besides classification, we can also generalize the CLM method to regression.

In this article, we focus on the implementation of the LUM loss and the logistic loss for classification and use them as examples to illustrate how the CLM method works.

The rest of paper is organized as follows. In Section 2.1, we briefly review binary classification methods. The CLM framework is introduced in Section 2.2, and the properties of the CLM and its connection to existing methods are also discussed. In Section 3, we present a principal component analysis (PCA) based computational strategy for non-sparse solutions and a refitting procedure for sparse solutions. We demonstrate the effectiveness of our method with simulated data and then apply the method to the analysis of Alzheimer's disease and cancer data in Sections 4 and 5, respectively. Concluding comments are given in Section 6.

2 Methodology

We first review binary classification and large margin classifiers in Section 2.1. Due to the limitation of existing large margin classifiers for classification with latent subclasses, we propose the Composite Large Margin (CLM) classifier in Section 2.2. We will also discuss the properties of the CLM with two particular loss functions, the LUM and logistic losses.

2.1 Review of Binary Classification

Suppose we have a training data set $\{(\mathbf{x}_i, y_i); i = 1, 2, \ldots n\}$ available. The class label $y \in \{\pm 1\}$, and the predictor $\mathbf{x}$ is a p-dimensional vector. Our goal is to build a classifier based on the training data for prediction of data points with $\mathbf{x}$ only. For a given binary classification problem, there are many techniques available in the literature and our focus in this paper is on large margin classifiers [8]. Given the training data set, a large margin classifier is trained to obtain $f(\mathbf{x}) : \mathcal{R}^k \to \mathcal{R}$, such that the predicted class label is assigned using the sign of $f(\mathbf{x})$. Note that we correctly predict the class label of $\mathbf{x}$ when $yf(\mathbf{x})$ is positive. The term $yf(\mathbf{x})$ is known as the functional margin. In general, the objective function of a large margin classifier can be written in the regularization framework of a loss plus a penalty. The loss is a measure of the goodness of fit between the model and data, and the penalty controls the complexity of the model to avoid overfitting. Specifically, we express the optimization problem of a large margin classifier as follows:

$$\min_{f \in \mathcal{F}} J(f) + \lambda \sum_{i=1}^{n} L(y_i f(\mathbf{x}_i)),$$

where $\mathcal{F}$ is the function class that all candidate solution functions belong to, $J(f)$ is a regularization term penalizing the complexity of f, $L(\cdot)$ is the loss function, and λ is a tuning parameter balancing the two terms. When the function $f(\mathbf{x})$ is linear with the form $w^T\mathbf{x}$, a common choice of $J(f)$ is the ℓ_2 penalty, i.e. $\|w\|_2^2$. A natural loss function is the so called 0–1 loss with value 1 if $yf(\mathbf{x}) \le 0$, and 0 otherwise, i.e. $L_{0-1}(yf(\mathbf{x})) = I\{yf(\mathbf{x}) \le 0\}$. However, the 0–1 loss is difficult for optimization due to its nonconvexity. Consequently, various convex surrogate loss functions have been proposed in the literature to alleviate the computational problem [35]. For example, SVM uses the hinge loss, penalized logistic regression uses the logistic loss, and AdaBoost uses the exponential loss [6].

Recently, Liu et al. [19] proposed a unified large margin machine (LUM) with a family of convex loss functions which contains DWD and SVM as special cases. The LUM loss function is differentiable everywhere, hence it has some computational advantage. As an important component of our proposed method, we will describe the LUM loss function in detail. The LUM loss is indexed by two parameters a and c with the following explicit form:

$$V(u)=\begin{cases} 1-u & \text{if } u \le \frac{c}{1+c}, \\ \frac{1}{1+c}\left(\frac{a}{(1+c)u-c+a}\right)^a & \text{if } u > \frac{c}{1+c}. \end{cases} \quad (2.1)$$

The left piece of $V(u)$ with $u \le \frac{c}{1+c}$ is the same as the hinge loss used in the SVM. The right piece is a convex curve whose shape is controlled by c with rate of decay controlled by a. With $a > 0$ and $c \to \infty$, LUM is equivalent to the standard SVM. With $a \to \infty$ and fixed c, the LUM loss is a hybrid of SVM and AdaBoost.

The techniques discussed above work well in many traditional classification problems. For large margin classifiers, it is common to use linear learning. One advantage of linear learning is its simple interpretation. Once the function $f(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$ is obtained, one can examine the importance of each dimension in $\mathbf{x}$ through its corresponding coefficients $\mathbf{w}$. When a linear function is insufficient, one can map the original linear space to a higher dimensional nonlinear space using kernel methods [8]. Despite its flexibility, it is typically more difficult to interpret. Our goal is to propose a class of classifiers that maintain sufficient flexibility to incorporate latent subclasses without losing the interpretability of linear classifiers.

2.2 The Composite Large Margin (CLM) framework

In this section, we describe the CLM framework for binary classification with latent subclasses in detail. We assume that due to the existence of heterogenous subclasses, a global single linear classifier that can effectively separate the positive and the negative classes does not exist. However, if we divide the data into two parts using a simple function (e.g. a linear function), then the classification task for each part becomes relatively easy.

Next, we describe our proposed CLM method. To that end, we first define the generalized $0-1$ latent classification loss as $W_{0-1}(y, \mathbf{x}) = I(f_1(\mathbf{x}) \le 0)I(yf_2(\mathbf{x}) \le 0) + I(f_1(\mathbf{x}) > 0)I(yf_3(\mathbf{x}) \le 0)$, where $f_1(\mathbf{x})$ is the splitting function, $f_2(\mathbf{x})$ is the classifier for data points with $f_1(\mathbf{x}) \le 0$, and $f_3(\mathbf{x})$ is the classifier for data points with $f_1(\mathbf{x}) > 0$.

The generalized $0-1$ latent classification loss is the composition of two $0-1$ standard binary classification loss functions with weights $I(f_1(\mathbf{x}) \le 0)$ and $I(f_1(\mathbf{x}) > 0)$. Similar to the standard $0-1$ loss, it is hard to optimize the generalized $0-1$ loss directly due to its discontinuity. In practice, a surrogate loss function is often used instead. For illustration, we use logistic and LUM losses as surrogate loss functions for the indicators $I(yf_2(\mathbf{x}) \le 0)$ and $I(yf_3(\mathbf{x}) \le 0)$. For weight functions $I(f_1(\mathbf{x}) \le 0)$ and $I(f_1(\mathbf{x}) > 0)$, we use $G(-f_1(\mathbf{x}))$ and $G(f_1(\mathbf{x}))$ as their corresponding smooth approximations, where $G(u)$ is defined as:

$$G(u)=\begin{cases} 1 & \text{if } u \geq \varepsilon, \\ 1-0.5(1-u/\varepsilon)^2 & 0 \leq u \leq \varepsilon \\ 0.5(1+u/\varepsilon)^2 & -\varepsilon \leq u \leq 0 \\ 0 & u \leq -\varepsilon, \end{cases} \quad (2.2)$$

ε is a parameter, which we can set to be small. Note that $G(u) + G(-u) = 1$, also as $\varepsilon \to 0$, $G(u)$ converges to $I(u > 0)$ pointwisely. Note that this choice is not unique, and there are many other possible approximations, such as sigmoid functions.

The loss functions W_{log} and W_{lum} for the latent classification problem are defined as follows:

$$W_{lum}(y_i, \mathbf{x}_i)=\alpha_i V(y_i f_2(\mathbf{x}_i))+(1-\alpha_i)V(y_i f_3(\mathbf{x}_i)), \quad (2.3)$$

$$W_{log}(y_i, \mathbf{x}_i)=\alpha_i \log(1+e^{-y_i f_2(\mathbf{x}_i)})+(1-\alpha_i)\log(1+e^{-y_i f_3(\mathbf{x}_i)}), \quad (2.4)$$

where $\alpha_i = G(-f_1(\mathbf{x}_i))$. Note that the W_{log} and W_{lum} losses are the compositions of two logistic and LUM loss functions, respectively. The α_i and $1 - \alpha_i$ are the weights.

Furthermore, we assume that f_1, f_2, f_3 are all linear, i.e. $f_j(\mathbf{x})=\mathbf{x}\mathbf{w}_j^T+b_j; j = 1, 2, 3$, to maintain the interpretability of linear classifiers.

With the loss function $L(y, \mathbf{x})$ defined, we can express the optimization problem for CLM as $\min_{\mathbf{w},\mathbf{b}} Q(\mathbf{w}, \mathbf{b}|\boldsymbol{Y}, \mathbf{X})=\frac{1}{2}\sum_{j=1}^{3}\|\mathbf{w}_j\|_2^2+\lambda\sum_{i=1}^{n} L(y_i, \mathbf{x}_i)$, where λ is the tuning parameter, and $(\mathbf{w}, \mathbf{b}) = (\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, b_1, b_2, b_3)$. We will discuss the algorithm for obtaining the CLM solution in Section 3.

As a remark, we would like to point that in the literature, there are two main categories of methods to handle data heterogeneity and identify potential subtypes: tree-based methods [2], and likelihood based mixture models [5].

For tree-based methods, several techniques [13, 20] were proposed to overcome the potential problems of splitting variables without global effects. For example, GUIDE allows the search for linear splits using two variables at one time when the marginal effects of both covariates are weak, and the pairwise interaction effect is strong [20]. As illustrated in Section 4, both GUIDE and CLM can work well for certain problems, while traditional tree-based methods cannot. Unlike the tree-based GUIDE, CLM is a composite large margin classifier motivated by latent variables. Furthermore, with the use of three linear functions, the interpretation of CLM is relatively simple. Lastly, our numerical examples indicate that CLM is more competitive for high dimensional data.

Likelihood based mixture models assume that the data come from several mixture components (with the number of components known), and the model usually has a hierarchical structure. An example of the hierarchical mixture of experts (HME) introduced by [11] is described below. We assume there are two layers with four components, and the parametric likelihood of Y given X is

$$P(Y|X,\theta)=\sum_{i=1}^{2} g(i)\sum_{j=1}^{2} g(j|i)\mu_{ij}^{I(y=1)}(1-\mu_{ij})^{I(y=-1)},$$

where $g(i) = \exp(q_i(x,\theta))/(\exp(q_1(x,\theta)) + \exp(q_2(x,\theta)))$,

$$g(j|i)=\exp(q_{j|i}(x))/(\exp(q_{1|i}(x,\theta))+\exp(q_{2|i}(x,\theta))), \mu_{ij}=E(I(Y=1)|i,j,X,\theta).$$

The $g(i)$ and $g(j/i)$ are the proportions for the four components, and μ_{ij} is the model for a given component. The task is to calculate the MLE for θ, and techniques, such as the EM algorithm, are usable. If we consider a specific form of CLM without the penalty term, for example, using logistic loss function for $f_2(x)$ and $f_3(x)$, and approximating $I(f_1(x) > 0)$ with $\exp(f_1(x))/(\exp(f_1(x))+\exp(f_1(-x)))$, then the one layer HME model has the same objective function as that of CLM. Despite the interesting connection, the motivations of CLM and HME are different. In particular, the CLM method is motivated from the perspective of latent subclasses and is a generalization of change-point models to the change-plane, while the HME is a likelihood based mixture model. When making a decision, CLM is similar to a "hard" classifier in the sense that it directly estimates the decision boundary of the latent classification problem represented by $I(f_1(x) > 0)$. In contrast, HME is similar to a "soft" classifier which first estimates the conditional class probability, then converts the probability into the decision. More details about "hard" and "soft" classifiers are described in [30]. In addition, the CLM is broader than likelihood-based methods and allows for more general loss functions. As shown in Section 4, a general loss function for CLM may provide better classification performance for complex problems. We show that CLM with LUM delivers smaller classification errors than CLM with the logistic loss in both of our application settings studied in Section 5.

Our proposed CLM also has interesting connections with the piecewise linear regression method for problems with continuous outcomes. The piecewise linear regression method splits the covariate space into subspaces by identifying the change point, and within each subspace, a linear model is assumed. Our method with the logistic loss splits the covariate space into subspaces by a linear function, and within each subspace, a generalized linear model is used.

3 Computational Algorithms for CLM

In this section, we discuss implementation of CLM. In particular, we describe a gradient based algorithm in Section 3.1. To tackle the difficulty of high dimensional problems, a PCA

based algorithm is given in Section 3.2. Based on this algorithm, we further describe a refitting procedure to achieve variable selection in Section 3.3.

3.1 Gradient based algorithm for the CLM

To describe the algorithm, we use CLM with the logistic loss as an example. The corresponding objective function is

$$Q_1^{\lambda}(\mathbf{w},\mathbf{b})=\frac{1}{2}\sum_{j=1}^{3}\|\mathbf{w}_j\|_2^2+\lambda\sum_{i=1}^{n}[\alpha_i\log(1+e^{-y_i f_2(\mathbf{x}_i)})+(1-\alpha_i)\log(1+e^{-y_i f_3(\mathbf{x}_i)})], \tag{3.1}$$

where λ, α_i and $f_j(\mathbf{x})$ are as defined in (2.4). Since the objective function is continuously differentiable, many general optimization algorithms, such as the conjugate gradient method or the quasi-Newton method [26], are applicable.

To apply these algorithms, we first need to derive the corresponding gradient functions. Once the gradient is given, we can iteratively update the solution. For example, for the gradient descent algorithm, the $(t+1)$-th step solution $(\mathbf{w}^{t+1}, \mathbf{b}^{t+1})$ based on the t-th step solution $(\mathbf{w}^{t+1}, \mathbf{b}^{t+1})$ is given by $\mathbf{w}_i^{t+1}=\mathbf{w}_i^t-\gamma\frac{\partial Q_1^{\lambda}(\mathbf{w}^t,\mathbf{b}^t)}{\partial \mathbf{w}_i}$, $b_i^{t+1}=b_i^t-\gamma\frac{\partial Q_1^{\lambda}(\mathbf{w}^t,\mathbf{b}^t)}{\partial b_i}$; $i=\ldots 3$, where γ is a small positive number known as the learning rate. Similar calculations can be done for other loss functions such as the W_{lum} loss in (2.3).

3.2 PCA algorithm for the CLM

The direct optimization strategy in Section 3.1 works well for low or moderate-size dimensional problems. However, direct optimization encounters significant challenges for high dimensional data since the computational burden of most general optimization methods increases dramatically with the dimension. To overcome the challenge of this problem, we incorporate the principal component idea. In particular, we predict the class label using the CLM method by a reduced rank design matrix instead of the original design matrix. The reduced rank matrix comprises the first k principal component scores of the original design matrix with $k < d$. The steps of this PCA-based algorithm for CLM are given in Algorithm 1 below.

Denote the design matrix as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T = [\mathbf{X}_1, \ldots, \mathbf{X}_d]$, which is an $n \times d$ matrix. The eigen-decomposition of $\mathbf{X}$ is $\mathbf{X}^T\mathbf{X} = \mathbf{P}\Lambda\mathbf{P}^T$, where the $\mathbf{P}$ is a matrix with orthonormal columns ($[\mathbf{P}_1, \ldots, \mathbf{P}_d]$) and Λ is the diagonal matrix with the eigenvalues as the diagonal elements. Furthermore, we define $\mathbf{P}^k = [\mathbf{P}_1, \ldots, \mathbf{P}_k]$, and $\mathbf{X}^k=\mathbf{X}\mathbf{P}^k=[\mathbf{x}_1^k,\ldots,\mathbf{x}_n^k]^T$. Our idea is to work with the k-dimensional space spanned by the first k principal component dimensions. In particular, instead of working with the d-dimensional $\mathbf{x}$, we work with the k-dimensional $\mathbf{x}^k$. If we replace the corresponding elements in Q_1^{λ} in (3.1) with the new linear functions $\tilde{f}_j(\mathbf{x}_i)=\mathbf{x}_i^k\tilde{\mathbf{w}}_j^T+\tilde{b}_j$ $(j=1,2,3)$, then we can obtain a new objective function $\tilde{Q}_1^{k,\lambda}$:

$$\tilde{Q}_1^{k,\lambda}(\tilde{\mathbf{w}},\tilde{\mathbf{b}})=\frac{1}{2}\sum_{j=1}^{3}\|\tilde{w}_j\|_2^2+\lambda\sum_{i=1}^{n}W_{log}(y_i,\mathbf{x}_i^k). \quad (3.2)$$

We minimize $\tilde{Q}_1^{k,\lambda}$ instead of Q_1^{λ} and get the minimizer ($\tilde{\mathbf{w}}^*$, $\tilde{\mathbf{b}}^*$). Consequently, we can calculate the solution for the original problem ($\mathbf{w}^*$,$\mathbf{b}^*$) by setting $\mathbf{w}^* = \tilde{\mathbf{w}}^*[\mathbf{P}^k]^T$, $\mathbf{b}^* = \tilde{\mathbf{b}}^*$. This strategy reduces the dimension of the problem from $3d$ to $3k$. If $k \ll d$, then we can handle high dimensional data relatively efficiently.

We would like to point out that although reducing the dimension greatly helps the computational efficiency, we may lose important classification information. Thus the choice of k is very important. We assume that the important classification information is mostly contained in the space spanned by the first k PC dimensions. Under this assumption, finding the right k helps to eliminate noise dimensions and improve computational efficiency as well as accuracy of the resulting classifier. Therefore, we need to measure the information of Y contained in $\mathbf{X}^k$ for various k. The traditional Pearson correlation is not appropriate for this purpose, since it restricts the two random vectors to be one dimensional and only measures the linear dependence. To address this problem, we use a recently proposed "distance correlation" (dcor) [28] for choosing the number of leading principal components k. The dcor measures arbitrary types of dependence between two random vectors. In particular, the distance covariance (dcov) between two random vectors $\mathbf{u}$ and $\mathbf{v}$ with finite first moments is written as $\mathrm{dcov}(\mathbf{u},\mathbf{v})=\sqrt{\int_{R^{d_u+d_v}}\|\phi_{\mathbf{u},\mathbf{v}}(\mathrm{t},\mathrm{s})-\phi_{\mathbf{u}}(\mathrm{t})\phi_{\mathbf{v}}(\mathrm{s})\|^2 w(\mathrm{t},\mathrm{s})\,d\mathrm{t}\,d\mathrm{s}}$, where $\varphi(.)$ represents the characteristic function, d_u and d_v are the dimensions of $\mathbf{u}$ and $\mathbf{v}$, and $w(\mathrm{t},\mathrm{s})$ is a properly defined weight function. The distance correlation between $\mathbf{u}$ and $\mathbf{v}$ is defined as $\mathrm{dcor}(\mathbf{u},\mathbf{v})=\mathrm{dcov}(\mathbf{u},\mathbf{v})/\sqrt{\mathrm{dcov}(\mathbf{u},\mathbf{u})\mathrm{dcov}(\mathbf{v},\mathbf{v})}$. Unlike the Pearson correlation, dcor is 0 if and only if the two random vectors are independent and it does not restrict the dimensions of those random vectors [28]. From its properties, the dcor is feasible and robust for screening information for classification problems [15]. We measure the information of Y contained in the leading k-PCs of $\mathbf{X}$ as dcor(Y, $\mathbf{X}^k$). Consequently, the optimal $k_{opt}=\underset{k=1,2,\ldots,d}{\mathrm{argmax}}\,\mathrm{dcor}(Y,\mathbf{X}^k)$. In practice, we find that using $\mathbf{X}^{k_{opt}}$ as the predictor may not always yield the lowest classification error, so in the implementation, we treat k as a tuning parameter from the set $\{k_{opt}, k_{opt}+1, k_{opt}+2\}$. This strategy appears to work well in practice.

PCA algorithm for the CLM method

(0) *Initialization:* Denote the training data set as $\{\mathbf{Y}_1, \mathbf{X}\}$ and the tuning data set $\{\mathbf{Y}_2, \mathbf{U}\}$. Let $k_{opt}=\underset{k=1,2,\ldots,d}{\mathrm{argmax}}\,\mathrm{dcor}(\boldsymbol{Y}_1,\mathbf{X}^k)$.

(1) *Model Training:* For fixed k and λ, solve $(\tilde{\mathbf{w}}^{k,\lambda},\tilde{\mathbf{b}}^{k,\lambda})=\underset{\tilde{\mathbf{w}},\tilde{\mathbf{b}}}{\mathrm{argmin}}Q_1^{k,\lambda}(\tilde{\mathbf{w}},\tilde{\mathbf{b}}|\boldsymbol{Y}_1,\mathbf{X}^k)$.

(2) *Solution Reconstruction:* Calculate $\mathbf{w}^{k,\lambda} = \tilde{\mathbf{w}}^{k,\lambda}[\mathbf{P}^k]^T$, $\mathbf{b}^{k,\lambda} = \tilde{\mathbf{b}}^{k,\lambda}$.

(3) *Prediction:* The tuning error $\xi(k,\lambda)$ is the classification error on predicting $\mathbf{Y}_2$ using $\mathbf{U}$ by the model from Step 2.

(4) *Iteration:* Repeat Steps 1 – 3 for all $k \in \{k_{opt}, k_{opt}+1, k_{opt}+2\}$ and $\lambda \in \{\lambda_1, \ldots, \lambda_n\}$.

(5) *Output:* The final solution $(\mathbf{w}^*, \mathbf{b}^*)$ is $(\mathbf{w}^{k,\lambda}, \mathbf{b}^{k,\lambda})$ from the model with the lowest $\xi(k,\lambda)$. For the choice of λ, we use a warm-start strategy, such that for the sequence of model fittings from λ_1 to λ_n ($\lambda_1 > \lambda_2 > \ldots > \lambda_n$), the solution based on λ_i is provided as the starting point for λ_{i+1}. This approach helps improve the convergence of the algorithm.

3.3 Refitting algorithm for sparse CLM

Variable selection is important for the analysis of high dimensional data. The PCA algorithm we proposed above does not have variable selection capability due to the choice of the ℓ_2 penalty and the solution reconstruction step. As $\mathbf{P}^k$ is full rank, $\mathbf{w} = \mathbf{w}[\mathbf{P}^k]^T$ is not sparse in general, even if $\mathbf{w}$ is. Consequently, we cannot achieve sparsity on $\mathbf{w}$ by simply replacing the ℓ_2 penalty with the lasso penalty in $\tilde{Q}_1^{k,\lambda}$ in (3.2).

To achieve variable selection, we propose a refitting procedure. We first identify the informative variables from the output of the PCA algorithm with all variables as predictors. Then, we refit the PCA algorithm using the informative variables only. The key step is to identify the informative variables. For this purpose, we fit three penalized logistic regression (PLR) models with the elastic net penalty [7, 27, 36] separately for f_1, f_2 and f_3. The details are given in Algorithm 2:

Refitting algorithm for the CLM method

(0) *Initialization:* Denote the training data set as $\{\mathbf{Y}, \mathbf{X}\}$ and the solution from Algorithm 1 as $\widehat{f_1}, \widehat{f_2}, \widehat{f_3}$. Let PLR($\mathbf{Y}$, $\mathbf{X}$) be the solution of the PLR fitting, and let the initial value of the active variable index set be $A = \{1, 2, \ldots, p\}$.

(1) *Approximate* f_1: Let $Z_i = \mathrm{sign}(\widehat{f_1}(\mathbf{x}_i^A))$. Then obtain $(\mathbf{w}_1^s, b_1^s) = \mathrm{PLR}(\boldsymbol{Z}, \mathbf{X}^A)$, where "s" represents the sparse solution.

(2) *Approximate* f_2 *and* f_3: The samples in $\{\mathbf{Y}, \mathbf{X}^A\}$ are divided into Set 1 and Set 2 by the sign of $f_1^s(\mathbf{X})$. Using Set 1 data, we have $(\mathbf{w}_2^s, b_2^s) = \mathrm{PLR}(\boldsymbol{Y}_{\mathrm{Set1}}, \mathbf{X}_{\mathrm{Set1}}^A)$, and similarly $(\mathbf{w}_3^s, b_3^s) = \mathrm{PLR}(\boldsymbol{Y}_{\mathrm{Set2}}, \mathbf{X}_{\mathrm{Set2}}^A)$.

(3) *Select active variables:* Let $A = I_1 \cup I_2 \cup I_3$, where $I_1 = \{j : w_{1j}^s \neq 0\}$, $I_2 = \{j : w_{2j}^s \neq 0\}$, $I_3 = \{j : w_{3j}^s \neq 0\}$.

(4) *Refit:* Refit Algorithm 1 using X^A as predictors.

(5) *Iteration:* Repeat Steps 1 – 4 until the active variable index set A stabilizes.

(6) *Output:* The final solution is obtained from the most recent Step 4. Based on our experience, the algorithm converges within several steps of refitting. In our simulation studies and application settings, we observe that the refitting procedure

performs well overall. Depending on the computational budget, one may also couple our algorithm with the stability selection procedure proposed by [22] to select the active variables.

4 Simulation Studies

We now investigate the performance of the proposed methods on two synthetic examples. We simulate both low- and high-dimensional situations. The training and testing data are generated from the same distributions with sample sizes 200 and 20000, respectively. For each example, there are several scenarios with different Bayes errors (the error rate of the optimal Bayes rule) and different numbers of noise variables. In both examples, the data contain four clusters of equal sizes. To reflect the latent subclass structure, two clusters belong to the "+" class and the others are for the "−" class. We compare the proposed methods with the linear SVM, the quadratic and Gaussian kernel SVMs, Random Forests, HME and l_1–penalized logistic regression. In addition, we also include an enhanced tree-based method GUIDE [20] for comparison. We select tuning parameters for all methods via five-fold cross validation.

Example 1 (Twisted Case)—The four clusters are sampled from four bi-variate normal distributions with corresponding means (μ, μ) and $(-\mu, -\mu)$ for the "+" class, $(\mu, -\mu)$ and $(-\mu, \mu)$ for the "−" class, and the identity covariance matrix. In addition to the two informative variables, we generate random noise variables from $N(0, 0.5^2)$. We present the scenarios with $\mu = 2.24$ or 1.2, where larger μ makes the classification task easier with smaller Bayes error. We also compare the performance of different methods on the scenarios with the same μ but different numbers of noise variables. Note that although the Bayes error only depends on μ in our example, the classification problem becomes more challenging when more noise variables are added. The scatter plot for the no noise variables scenario ($\mu = 2.24$) with the CLM solution boundary is shown in Figure 2(a).

Example 2 (Parallel Case)—The four clusters are sampled from four bi-variate normal distributions whose means are $(\mu, 0)$, $(-\mu, 0)$ for the "+" class, and $(0, 0)$, $(2\mu, 0)$ for the "−" class. The covariance matrix is $\mathbf{\Sigma} = 0.6\mathbf{I}_{2\times 2} + 0.4\mathbf{J}_{2\times 2}$, where $\mathbf{I}$ is the identity matrix, and $\mathbf{J}$ is a matrix with all elements equal to 1. We set $\mu = 3.90$ or 2, and the additional random noise variables follow i.i.d $N(0, 0.5^2)$. As in Example 1, a similar scatter plot ($\mu = 3.90$, no noise variables) is shown in Figure 2(b), where the four clusters are parallel to each other. The testing errors of the CLM and other methods in both examples are reported in Table 1.

The results of Examples 1 and 2 show that our methods outperform the competitors in most scenarios. Both examples have latent subclasses, and our method is well suited for these problems. The results show that our method is very effective in detecting the true latent structure.

In the twisted case, linear SVM and PLR methods fail to detect the pattern. Although the quadratic kernel method works, it still has larger testing errors than the CLM methods. In Example 1, we can divide the samples into four clusters by lines $X_1 = 0$ and $X_2 = 0$. Since these lines can be approximated by a quadratic function, the quadratic kernel can work well

in the twisted case. For the parallel case, we need three parallel lines to separate the four clusters, so methods such as the quadratic kernel SVM, the linear SVM, and PLR do not perform well. For both examples, the Gaussian kernel SVM and Random Forests work well under the low dimensional setting. When the dimension is high, Random Forests fail to choose the right covariate to split among all covariates. In addition, the Random Forests will only split variables with strong marginal effects, hence the performance of Random Forests is better in the parallel case than in the twisted case. The model structure of the Gaussian kernel SVM is flexible enough to separate the two classes for low dimensions. However, its performance decays rapidly in the high dimensional setting, possibly due to overfitting. In contrast, the results of the CLM method under the high dimensional setting are still comparable to those under the low dimensional setting. The CLM method uses linear classifiers, and therefore may alleviate the overfitting problem compared to kernel based methods.

When there are no latent subclasses or only one class has subclasses, our method is still comparable to that of the competitors (see Appendix A for details). Note that for the parallel example, X_2 carries almost all the information about Y. As a result, GUIDE performs well for the parallel case in general due to its ability to generate unbiased splits regardless of whether it searches for pairwise interactions or not. In the twisted example, the marginal effect of X_1 and X_2 is weak, while the interaction effect is strong. However, when the dimension is high, due to the high computational demand, the implementation of GUIDE does not perform interaction tests for determining which covariates to split. Hence, GUIDE does not perform very well in the twisted case with $p = 1000$. In both examples, HME performs worse than CLM when the dimension is high or the classification task is challenging, as the CLM can better identify the latent variable (see Appendix B for details). Overall, the performance of CLM is the best among all methods.

Note that little information is lost when applying the PCA strategy. In particular, the structure of the four clusters is preserved after its application: the projected data pattern in the space spanned by X_1 and X_2 in Figures 2(a) is similar to the projected data pattern in the space spanned by the first two principal components in Figures 2(c). Similar phenomenon exist in 2(b) and 2(d).

5 Applications

In this section, we apply our methods to the analysis of Alzheimer's disease and ovarian carcinoma data. The first data set involves imaging data, while the second involves gene expression data. We are interested in the ability of our method to detect meaningful latent heterogenous groups while simultaneously delivering competitive classification accuracy. The performance of different methods is evaluated by cross validation (CV) errors of 100 random divisions of the data. We use 75% of data for training and 25% for testing. We also keep the original class proportions within both the training and testing sets. In the training steps, tuning parameters are selected by 5-fold CV. We compare the same methods evaluated in Section 4, and report the average testing errors in Table 2.

5.1 Alzheimer's disease

We first apply our methods to the Alzheimer's Disease Neuroimaging Initiative (ADNI) [24], a longitudinal study designed for the Alzheimer's disease detection and tracking. There are 226 normal controls and 393 MCI patients in total. The primary goal of our analysis is to discriminate between MCI samples and normal control samples using imaging data collected at their first visit. As mentioned previously, such a discrimination tool could help physicians know when to begin intervention. Each sample is characterized by features extracted from structural MR imaging (MRI) which measures brain atrophy (a known AD related factor). The image pre-processing and feature extraction follow the procedure described in [31] and [34]. Basically, each processed image is divided into 93 regions-of-interest (ROI) and then the volume of grey matter tissue in each ROI region is computed as a feature [34]. The results of classifying MCI versus normal controls using the 93 MRI features are shown in the left panel of Table 2. We can see that the sparse CLM with the LUM loss provides the smallest average testing error for discriminating AD and normal control among all methods.

In addition to accurate diagnosis of AD/MCI, another important task in the AD research is to predict whether or not MCI patients will convert to AD. Clinically, AD and MCI are defined according to certain severity measures of dementia symptoms at the baseline based on the Clinical Dementia Rating (CDR) scale [23]. In particular, CDR = 0 is considered normal, CDR = 0.5 is considered MCI, and CDR = 1, 2, 3 are considered AD with different levels of severity. If an MCI patient's condition worsens during the follow-up (i.e., they receive a CDR higher than 1), then he/she is considered a converter (to AD). Otherwise, they are considered nonconverters. For the data set we analyze, there are 167 converters and 226 nonconverters. We found that if we predict MCI patients with $\widehat{f_1}(\mathbf{x})<0$ as converters, and the other MCI patients as nonconverters, then the average prediction accuracy among 100 replications is 0.799 (sd=0.14). This prediction result is greater than the accuracy of directly predicting MCI subclasses using the same baseline MRI data. In particular, the linear classifier constructed from l_1-penalized logistic regression has a ten fold cross validation error value of 0.69. This result indicates that MRI data can characterize the heterogeneity among MCI patients, which provides new insights for prognosis of MCI. Therefore, besides successfully classifying MCI versus normal control patients, the proposed CLM also provides a good prognostic tool for MCI patients in terms of separating converters from nonconverters. Our method discovers meaningful latent subclasses of MCI without using follow-up clinical information.

5.2 Ovarian Carcinoma

The second application is to an ovarian carcinoma data set containing 11, 864 genes from the TCGA data portal [25]. There are four cancer subtypes: immunoreactive (sample size is 107), differentiated (135), proliferative (138) and mesenchymal (109). We focus on classifying the proliferative samples ("+" class) versus the non-proliferative samples ("−" class). We select 3520 genes with the largest median absolute deviance (MAD) value for further classification analysis. From the testing errors shown in the right part of Table 2, we can see that the CLM with the LUM loss performs the best, followed by the CLM with the logistic loss. Both methods perform better than the other competitors.

To better understand the results, we can visualize the results of the CLM method by projecting the samples onto the space spanned by $\widehat{f_1}, \widehat{f_2}$ if $\widehat{f_1}(\mathbf{x})<0$, and onto the space spanned by $\widehat{f_1}, \widehat{f_3}$ otherwise. Figure 3 suggests that some subgroups may exist within the proliferative samples (Pro(A) and Pro(B)). Additionally, the "−" class samples are grouped into two parts: one consists of some immunoreactive samples (Imm(B)) and all differentiated samples, the other consists of the remaining immunoreactive samples (Imm(A)) and almost all mesenchymal samples.

To confirm statistical stability of the subgroups found in proliferative and immunoreactive samples, the Sigclust method proposed by [17] is applied for testing whether or not the difference between the two subgroups is significant. The subgroups are determined by the average of the sign of $\widehat{f_1}$ in the 100 simulations given by the CLM method with the LUM loss. The p-value for the subclasses within the immunoreactive subtype is very significant, i.e., < 0.001, while the p-value for the subclasses within the proliferative samples is not.

The detected subgroups can also be visualized from the heatmap of gene expression of a subset of genes (size = 153) in Figure 4. These genes were selected more than 15 times as "active variables" by the sparse CLM with the LUM loss among the 100 random splits. The genes are displayed in rows, and samples are shown in columns where the red line separates samples by the sign of $\widehat{f_1}$ (positive on the left). The plot shows that there exists a clear distinction between Imm(A) and Imm(B), as well as between Pro(A) and Pro(B) in the gene expression level, which suggests our latent subclass findings are not random. Additionally, the plot indicates that the sign of $\widehat{f_1}$ is driven by 20 genes. We apply a gene functional enrichment analysis using DAVID [9] on these 20 genes, and we find that most of them (17 with adjusted p-value $\leq 3 \times 10^{-4}$) are related to glycoprotein and secreted protein. These findings suggest that further biological investigation may be worth pursuing. Note that the selected active genes consist of 5 percent of the genes in the training data set, so our methods not only decrease the classification error and detect new subclass structure, they also facilitate variable selection.

6 Discussion

In this article, we propose Composite Large Margin classifiers to address the classification problem with latent subclasses by splitting the data and classifying the subsets using separate linear classifiers. Our approach inherits the nice interpretability of the standard linear approach while remaining flexible enough to handle complex data structures. In addition, our classifier is simpler than more complex methods such as kernel techniques and tree based methods. Consequently, it has less tendency for overfitting. The CLM method not only detects latent subclasses, but also enables visualization of high dimensional data using low dimensional plots.

To achieve feature selection, we also propose a refitting algorithm for CLM. One future direction is to explore variable selection consistency. Another direction is to make use of other penalties such as the group-lasso penalty [32] in CLM besides the l_2 penalty for selecting groups of variables.

Although our focus is on the CLM with the LUM and logistic losses, the basic idea can be implemented with other linear classifiers as well. Currently, our method is designed for binary classification with up to two latent subclasses in each class, so only one splitting function is needed. We can extend the CLM method to permit multiple splits to handle data with potentially multiple latent subclasses. In particular, we can extend the method by finding multiple splitting functions to divide the covariate space and fit a separate linear classifier for each region. We can also extend the CLM method for multicategory classification with heterogeneous data by modifying the loss function for the two (sub)classifiers. For example, we can use MLUM loss [33] or multiclass ψ-learning loss [18] for multicategory classification.

Acknowledgments

The work were partially funded by NSF grant DMS-1407241, NIH/NCI grant R01 CA-149569, and NCI Grant P01 CA142538. Data used in Section 5.1 of this manuscript were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: Abbott; Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Amorfix Life Sciences Ltd.; AstraZeneca; Bayer HealthCare; BioClinica, Inc.; Biogen Idec Inc.; Bristol-Myers Squibb Company; Eisai Inc.; Elan Pharmaceuticals Inc.; Eli Lilly and Company; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; GE Healthcare; Innogenetics, N.V.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Medpace, Inc.; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Servier; Synarc Inc.; and Takeda Pharmaceutical Company. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Disease Cooperative Study at the University of California, San Diego. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of California, Los Angeles. This research was also supported by NIH grants P30 AG010129, K01 AG030514, and the Dana Foundation.

References

1. Breiman L. Random forests. Machine learning. 2001; 45(1):5–32.
2. Breiman, L.; Friedman, JH.; Olshen, RA.; Stone, CJ. Classification and Regression Trees. Vol. 19. CRC Press; 1984.
3. Carlstein, EG.; Müller, HG.; Siegmund, D. Change-Point Problems. Vol. 23. Institute of Mathematical Statistics; 1994.
4. Fisher RA. The use of multiple measurements in taxonomic problems. Annals of Eugenics. 1936; 7(2):179–188.
5. Fraley, C.; Raftery, AE. Model-Based Clustering, Discriminant Analysis, and Density Estimation. 2002.
6. Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. Annals of Statistics. 2000; 28(2):337–407.
7. Friedman J, Hastie T, Tibshirani R. Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal Of Statistical Software. 2010; 33(1):1–22. [PubMed: 20808728]
8. Hastie, T.; Tibshirani, R.; Friedman, J. The Elements of Statistical Learning. Vol. 1. Springer; New York: 2009.
9. Huang DW, Sherman BT, Lempicki RA. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nature protocols. 2009; 4(1):44–57. [PubMed: 19131956]
10. Huang H, Liu Y, Marron JS. Bidirectional Discrimination with Application to Data Visualization. Biometrika. 2012; 99(4):851–864. [PubMed: 23843672]
11. Jordan MI, Jacobs RA. Hierarchical Mixtures of Experts and the EM Algorithm. Neural Computation. 1994; 6(2):181–214.

12. Kang, C. PhD thesis. Univerty of North Carolina; Chapel Hill: 2011. New Statistical Learning Methods for Chemical Toxicity Data Analysis.
13. Kim H, Loh WY. Classification Trees With Unbiased Multiway Splits. Journal of the American Statistical Association. 2001; 96(454):589–604.
14. Leek JT, Scharpf RB, Bravo HC, Simcha D, Langmead B, Johnson WE, Geman D, Baggerly K, Irizarry RA. Tackling the widespread and critical impact of batch effects in high-throughput data. Nature Reviews Genetics. 2010; 11(10):733–739.
15. Li R, Zhong W, Zhu L. Feature Screening via Distance Correlation Learning. Journal of the American Statistical Association. 2012; 107(499):1129–1139. [PubMed: 25249709]
16. Liu X, Parker J, Fan C, Perou CM, Marron JS. Visualization of cross-platform microarray normalization. Batch Effects and Noise in Microarray Experiments: Sources and Solutions. 2009:167–181.
17. Liu Y, Hayes DN, Nobel A, Marron JS. Statistical Significance of Clustering for High-Dimension, Low-Sample Size Data. Journal of the American Statistical Association. 2008; 103(483):1281–1293.
18. Liu Y, Shen X. Multicategory -learning. Journal of the American Statistical Association. 2006; 101(474):500–509.
19. Liu Y, Zhang HH, Wu Y. Hard or Soft Classification? Large-Margin Unified Machines. Journal of the American Statistical Association. 2011; 106(493):166–177. [PubMed: 22162896]
20. Loh WY. Improving the precision of classification trees. The Annals of Applied Statistics. 2010; 3(4):1710–1737.
21. Marron JS, Todd MJ, Ahn J. Distance Weighted Discrimination. Journal of the American Statistical Association. 2007; 102(480):1267–1271.
22. Meinshausen N, Bühlmann P. Stability selection. Journal of the Royal Statistical Society - Series B: Statistical Methodology. 2010; 72(4):417–473.
23. Misra C, Fan Y, Davatzikos C. Baseline and longitudinal patterns of brain atrophy in MCI patients, and their use in prediction of short-term conversion to AD: results from ADNI. NeuroImage. 2009; 44(4):1415–1422. [PubMed: 19027862]
24. Mueller SG, Weiner MW, Thal LJ, Petersen RC, Jack CR, Jagust W, Trojanowski JQ, Toga AW, Beckett L. Ways toward an early diagnosis in Alzheimer's disease: the Alzheimer's Disease Neuroimaging Initiative (ADNI). Alzheimer's & dementia. 2005; 1(1):55–66.
25. Cancer Genome Atlas Research Network et al. Integrated genomic analyses of ovarian carcinoma. Nature. 2011; 474(7353):609–615. [PubMed: 21720365]
26. Nocedal, J.; Wright, SJ. Numerical Optimization. Vol. 43. Springer; New York, NY: 1999.
27. Park MY, Hastie T. L1-regularization path algorithm for generalized linear models. Journal of the Royal Statistical Society - Series B: Statistical Methodology. 2007; 69(4):659–677.
28. Székely GJ, Rizzo ML, Bakirov NK. Measuring and testing dependence by correlation of distances. Annals of Statistics. 2008; 35(6):2769–2794.
29. Vapnik, VN. The Nature of Statistical Learning Theory. Vol. 8. Springer; New York, NY: 1995.
30. Wahba G. Soft and hard classification by reproducing kernel Hilbert space methods. Proceedings of the National Academy of Sciences of the United States of America. 2002; 99(26):16524–16530. [PubMed: 12477931]
31. Wang, Y.; Nie, J.; Yap, PT.; Shi, F.; Guo, L.; Shen, D. Robust deformable-surface-based skull-stripping for large-scale studies. Vol. 6893. Springer; Berlin/Heidelberg, Toronto, Canada: 2011.
32. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society - Series B: Statistical Methodology. 2006; 68(1):49–67.
33. Zhang C, Liu Y. Multicategory large-margin unified machines. The Journal of Machine Learning Research. 2013; 14(1):1349–1386. [PubMed: 24415909]
34. Zhang D, Wang Y, Zhou L, Yuan H, Shen D. Multimodal classification of Alzheimer's disease and mild cognitive impairment. NeuroImage. 2011; 55(3):856–67. [PubMed: 21236349]
35. Zhang T. Statistical behavior and consistency of classification methods based on convex risk minimization. Annals of Statistics. 2004; 32(1):56–85.

36. Zou H, Hastie T. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society - Series B: Statistical Methodology. 2005; 67(2):301–320.

Appendix A

We investigate the performance of the proposed methods on two additional simulation examples to access the impact of the existence of latent subclasses on the proposed methods.

Example 3

The two equal size clusters are sampled from two bi-variate normal distributions with corresponding means (1, 1) for the "+" class, (−1, 1) for the "−" class, and the identity covariance matrix. In addition to the two informative variables, we generate random noise variables from $N(0, 0.5^2)$. We report the results with $p = 1000$. The training and testing sample sizes are 200 and 20000, respectively. In principle, one linear function is sufficient for classifying the data. In this case, there does not exist latent subclasses, and we show that the CLM methods still gives competitive performance in Table 3.

Example 4

Three equal sized clusters are sampled from three bivariate normal distributions with means (1, 1) and (−1, −1) for the "+" class, and (1, −1) for the "−" class. The training and testing sample sizes are 150 and 20000 respectively. Other settings for this example are the same as that of Example 3. The numerical results of different methods are presented in the right side of Table 3. In this example, only the "+" class has latent subclasses. Similar to the results in Section 4, we can see that the sparse CLM with the LUM loss and the logistic loss deliver the lowest classification errors.

Appendix B

In this section, we compare the performance of CLM and HME in terms of identifying the latent structure. As both methods identify a linear function of X as surrogate for the latent class variable that can be used to separate the data into two parts, we can calculate the angle between the oracle linear function and the linear functions estimated from CLM and HME. Consider the twisted case with $p = 1000$ and $\mu = 1.2$ as an example (results given in Table 4). In this setting, the oracle linear function is either $X_1 = 0$ or $X_2 = 0$. In each simulation, if the angle between $X_1 = 0$ and the estimated line is small, then we use $X_1 = 0$ as the truth, otherwise we use $X_2 = 0$. We also calculate the classification error for the latent class variable. We can see that the separating line $\widehat{f_1(x)}$ from the CLM methods have a smaller angle to the oracle line than that from HME. In addition, the CLM methods have a smaller classification error in classifying the latent class. Both results imply that CLM can better identify the latent structure.

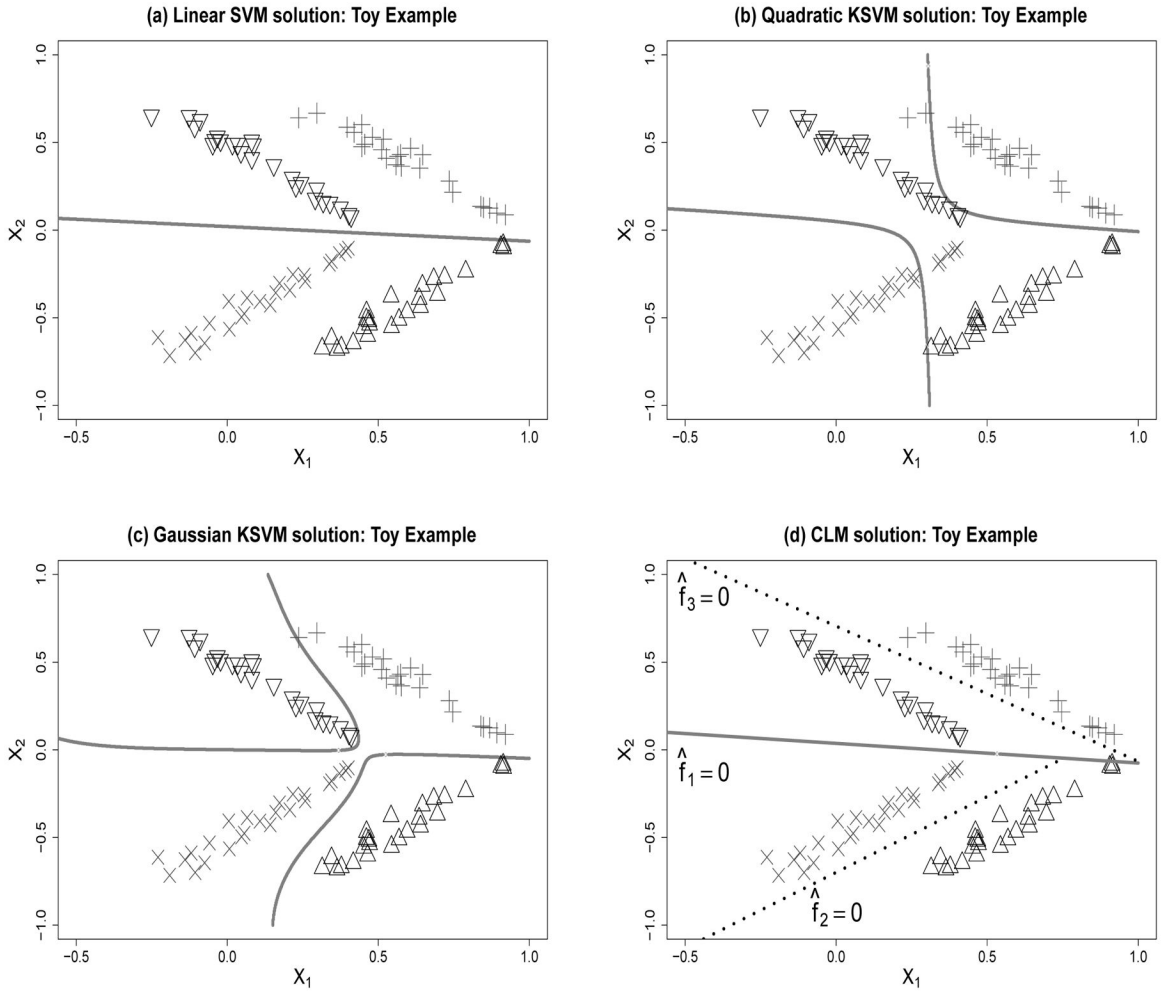

Figure 1.

Illustration of a two dimensional toy example. Gray (+ and ×) represents the positive class and black (▽ and △) represents the negative class. In panels (a), (b) and (c), the decision boundaries are drawn with wide gray lines. In panel (d) for the CLM method, the wide grey line splits the data into two parts and in each part the dashed line is the separating hyperplane for the corresponding classifier.

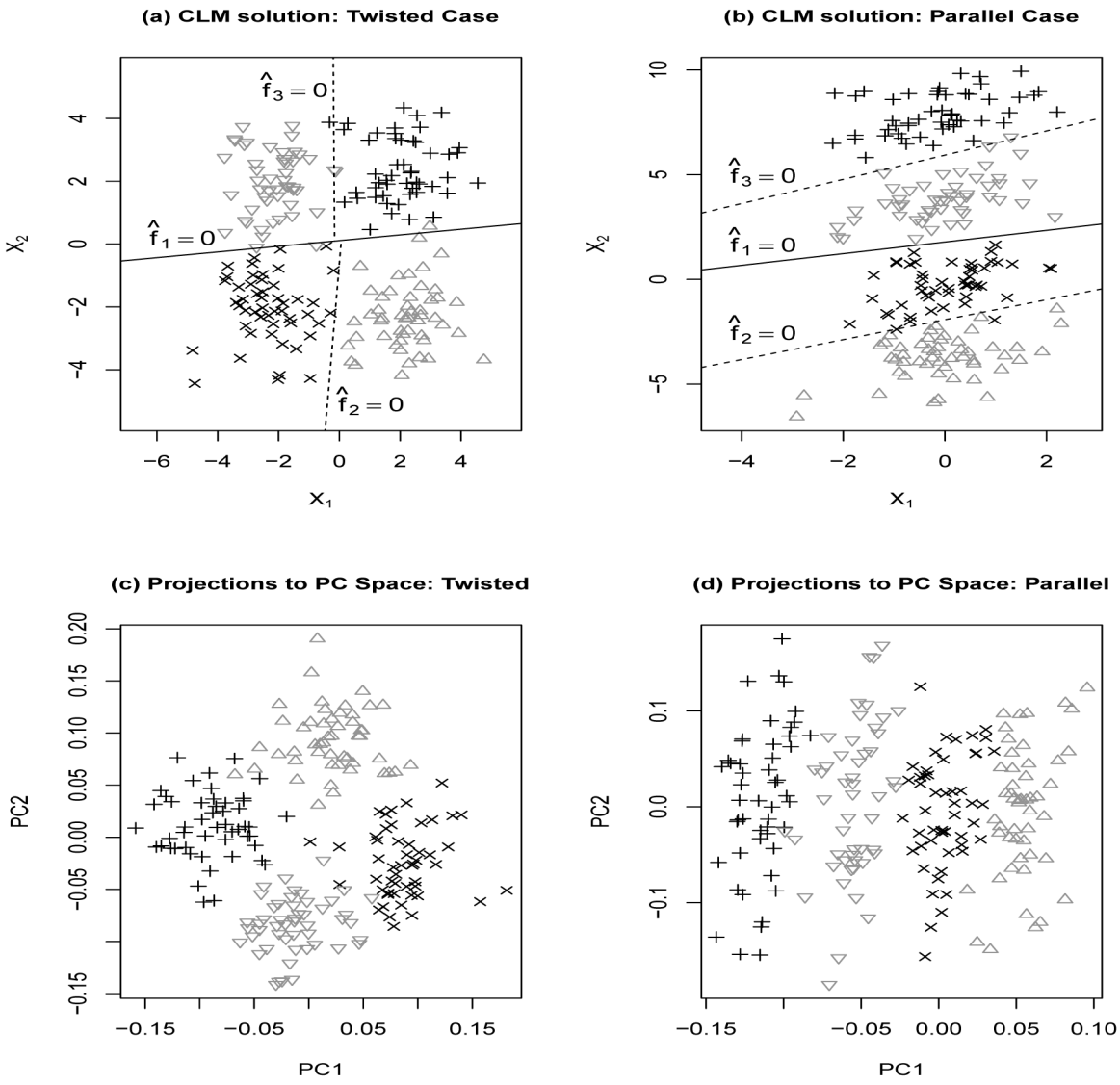

Figure 2.

Plots for CLM methods in twisted and parallel cases. Black color (+ and ×) represents the positive class and grey color (▽ and △) represents the negative class. Different symbols in the same class indicates the latent subclasses. In both panels (a) and (b), the boundary of $\widehat{f_1}$ is shown as a solid line and the boundaries of $\widehat{f_2}$ and $\widehat{f_3}$ are given as dashed lines. In panels (c) and (d), we show projections onto the space spanned by the first 2 principal components for the twisted and parallel cases. We apply PCA on the data which contains the informative variables X_1 and X_2 as well as an additional 998 noise variables. The four-clusters structure in the original space (as observed in panels (a) and (b)) is preserved in the PC space.

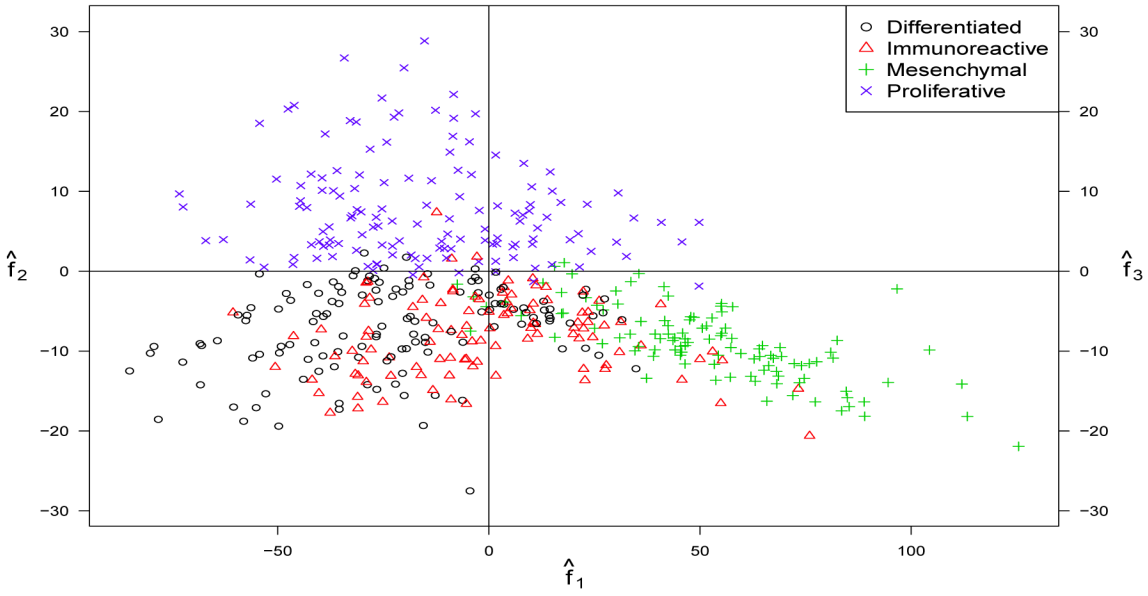

Figure 3.

Visualization of latent subclasses in the ovarian cancer dataset. The x-axis is the $\widehat{f_1}$ value, the y-axis displays the $\widehat{f_2}$ value of the points for which $\widehat{f_1}$ is less than 0, otherwise it displays the $\widehat{f_3}$ value. The plot indicates that subclasses exist within both the proliferative and immunoreactive types of ovarian cancer.

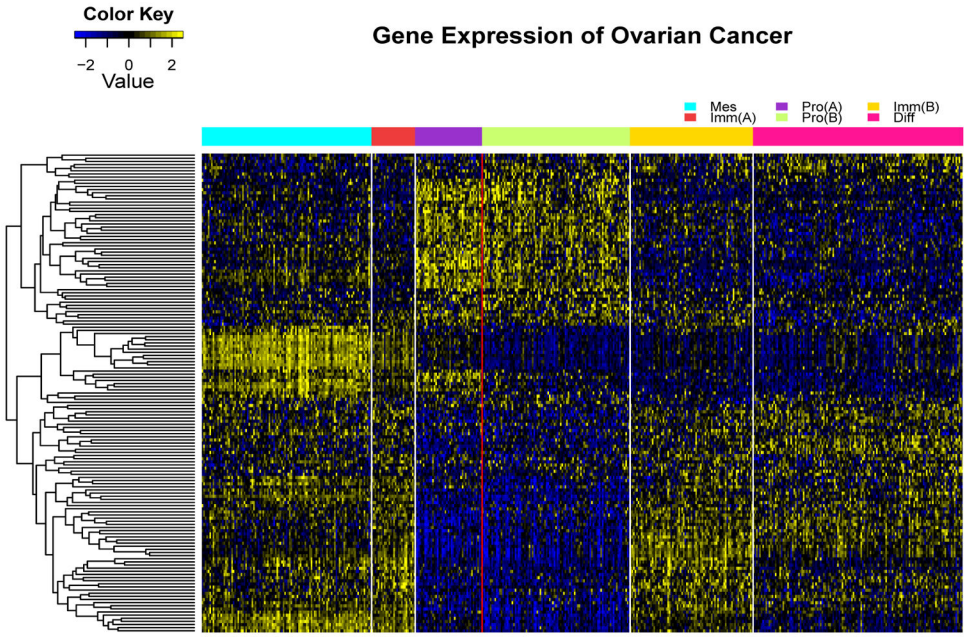

Figure 4.

Heatmap of ovarian cancer data using 153 active genes selected by the sparse CLM with the LUM loss. Samples are displayed in columns by subtypes. Genes are ordered by hierarchical clustering. Nearly all samples on the left of the red line have an average f_1 greater than 0, and the remaining samples have an average f_1 less than 0. We can see a clear distinction between Imm(A) and Imm(B), and a mild difference between Pro(A) and Pro(B), which suggests that subclasses exist in the immunoreactive and differentiated subtypes.

Table 1

Average testing errors in the simulation data with standard deviations in parentheses

Twisted Case						
Bayes Error (ε)	**μ = 2.24, ε = 0.025**			**μ = 1.2, ε = 0.204**		
Dimension	**p = 2**	**p = 100**	**p = 1000**	**p = 2**	**p = 100**	**p = 1000**
LSVM	.363 (.073)	.499 (.007)	.499 (.006)	.444 (.034)	.499 (.007)	.499 (.007)
KSVM(quadratic)	.035 (.006)	.040 (.008)	.072 (.020)	.221 (.010)	.270 (.012)	.390 (.024)
KSVM(Gaussian)	.032 (.004)	.120 (.011)	.454 (.004)	.232 (.010)	.409 (.012)	.496 (.002)
Random Forests	.033 (.004)	.491 (.004)	.500 (.001)	.231 (.012)	.495 (.003)	.500 (.001)
PLR	.500 (.001)	.500 (.001)	.500 (.002)	.500 (.002)	.500 (.001)	.500 (.002)
GUIDE	.031 (.003)	.039 (.020)	.496 (.004)	.220 (.008)	.308 (.021)	.498 (.002)
HME	.034 (.004)	.042 (.018)	.091 (.024)	.220 (.019)	.254 (.024)	.332 (.036)
CLM$_{log}$	.028 (.002)	.028 (.003)	.033 (.003)	.218 (.009)	.219 (.008)	.251 (.011)
CLM$_{lum}$	.028 (.002)	.029 (.003)	.033 (.003)	.215 (.006)	.220 (.008)	.249 (.009)
CLM$_{log}$ Sparse	.028 (.002)	.029 (.003)	.030 (.004)	.218 (.009)	.220 (.010)	.221 (.011)
CLM$_{lum}$ Sparse	.028 (.002)	.028 (.003)	.028 (.003)	.215 (.006)	.219 (.009)	.220 (.011)

Parallel Case						
Bayes Error(ε)	**μ = 3.9, ε = 0.024**			**μ = 2, ε = 0.206**		
Dimension	**p = 2**	**p = 100**	**p = 1000**	**p = 2**	**p = 100**	**p = 1000**
LSVM	.487 (.006)	.428 (.013)	.382 (.004)	.423 (.008)	.437 (.013)	.397 (.007)
KSVM(quadratic)	.377 (.048)	.435 (.013)	.342 (.016)	.378 (.018)	.421 (.008)	.395 (.013)
KSVM(Gaussian)	.035 (.004)	.258 (.010)	.385 (.003)	.286 (.029)	.401 (.008)	.380 (.003)
Random Forests	.040 (.006)	.289 (.065)	.333 (.082)	.259 (.041)	.372 (.034)	.415 (.051)
PLR	.332 (.012)	.352 (.025)	.405 (.040)	.378 (.027)	.401 (.032)	.405 (.032)
GUIDE	.042 (.003)	.047 (.006)	.052 (.006)	.238 (.009)	.281 (.027)	.304 (.030)
HME	.055 (.004)	.063 (.007)	.072 (.012)	.236 (.012)	.284 (.032)	.310 (.033)
CLM$_{log}$	.032 (.003)	.032 (.003)	.041 (.004)	.233 (.025)	.234 (.014)	.260 (.013)
CLM$_{lum}$	.031 (.003)	.032 (.004)	.041 (.004)	.225 (.014)	.227 (.011)	.257 (.010)

Parallel Case

Bayes Error(ε)	$\mu = 3.9, \varepsilon = 0.024$			$\mu = 2, \varepsilon = 0.206$		
Dimension	$p = 2$	$p = 100$	$p = 1000$	$p = 2$	$p = 100$	$p = 1000$
CLM_{log} Sparse	.032 (.003)	.034 (.005)	.037 (.008)	.233 (.025)	.241 (.019)	.260 (.024)
CLM_{lum} Sparse	.031 (.003)	.034 (.003)	.039 (.003)	.225 (.014)	.242 (.009)	.263 (.012)

Table 2

Testing errors on classifying Alzheimer's disease and ovarian cancer data with standard deviations in parentheses

	Alzheimer Disease	Ovarian Cancer
LSVM	.322 (.028)	.075 (.018)
KSVM (quadratic)	.384 (.032)	.148 (.021)
KSVM (Gaussian)	.333 (.033)	.065 (.017)
Random Forests	.323 (.027)	.080 (.022)
PLR	.323 (.030)	.078 (.019)
GUIDE	.323 (.029)	.083 (.021)
HME	.331 (.031)	.066 (.022)
CLM_{log}	.330 (.030)	.054 (.020)
CLM_{lum}	.314 (.030)	.042 (.014)
CLM_{log} Sparse	.315 (.025)	.071 (.021)
CLM_{lum} Sparse	.297 (.024)	.058 (.017)

Table 3

Average testing error rates for data in Example 3 and 4

Methods	**Example 1: p = 1000**	**Example 2: p = 1000**
LSVM	.245 (.011)	.302 (.011)
KSVM(quadratic)	.341 (.008)	.333 (.002)
KSVM(Gaussian)	.220 (.008)	.293 (.009)
Random Forests	.299 (.114)	.333 (.001)
PLR	.161 (.003)	.230 (.008)
GUIDE	.166 (.005)	.284 (.005)
HME	.197 (.008)	.264 (.025)
CLM_{log}	.191 (.007)	.250 (.018)
CLM_{lum}	.191 (.008)	.246 (.017)
CLM_{log} Sparse	.166 (.008)	.206 (.015)
CLM_{lum} Sparse	.166 (.006)	.207 (.015)

Table 4

Performance of detecting latent structure

Method	Sin(angle)	Classification Error
HME	.201 (.08)	.197 (.05)
CLM_{log}	.134 (.04)	.121 (.02)
CLM_{lum}	.122 (.04)	.112 (.02)
CLM_{log} Sparse	.098 (.03)	.093 (.01)
CLM_{lum} Sparse	.093 (.03)	.082 (.01)