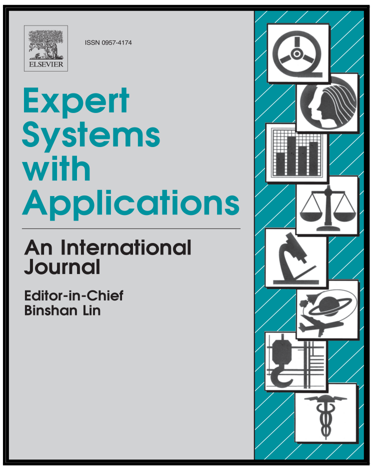

Embedded Feature Selection Accounting for Unknown Data Heterogeneity

Meng Lu

PII: S0957-4174(18)30726-7
DOI: https://doi.org/10.1016/j.eswa.2018.11.006
Reference: ESWA 12305

To appear in: *Expert Systems With Applications*

Received date: 11 July 2018
Revised date: 26 September 2018
Accepted date: 5 November 2018

Please cite this article as: Meng Lu, Embedded Feature Selection Accounting for Unknown Data Heterogeneity, *Expert Systems With Applications* (2018), doi: https://doi.org/10.1016/j.eswa.2018.11.006

Highlights

- Data heterogeneity leads to spurious classification and feature selection results.
- Our embedded feature selection method can account for unknown data heterogeneity.
- Sparse optimal scoring on the adjusted data is proposed for multi-class classification.
- Effective proximal gradient update rules are developed to find optimal solutions.
- Our method outperforms the state-of-the-arts on synthetic data and three benchmark image datasets.

Embedded Feature Selection Accounting for Unknown Data Heterogeneity

Meng Lu[a,b,*]

[a]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77840, USA.
[b]Department of Information Management and Management Science, Tianjin University, Tianjin 300072, China.

Abstract

Data heterogeneity is one of the big challenges in modern data analysis caused by the effects of unknown/unwanted factors introduced during data collection procedures. It will cause spurious estimation of variable effects when traditional methods are applied for feature selection which simply assume that data samples are independently and identically distributed. Although some existing statistical models can evaluate more accurately the significance of each variable by estimating and including unknown factors as covariates, they are categorized as filter methods suffering from variable redundancy and lack of predictability. Therefore, we propose an embedded feature selection method from a sparse learning perspective capable of adjusting unknown heterogeneity. Its performance is investigated by evaluating the classification performance using the selected features in multi-class classification problems. Benefitting from the effective adjustment of unknown heterogeneity and model selection strategy, the experimental results on synthetic data and three real-world benchmark data sets have shown that our method can achieve consistent superiority over several conventional embedded methods and existing statistical models.

Keywords: feature selection, data heterogeneity, embedded method, sparse optimal scoring.

[*]Corresponding author
Email address: lvmeng0502@gmail.com (Meng Lu)

1. Introduction

In real-world applications, heterogenous data is becoming more prevalent (Fan et al., 2014; Li et al., 2016) with the generation of big data with large number of samples or measured features. Typically heterogenous data refers to data coming from disparate data sources in machine learning area. However, data heterogeneity has a broader definition in statistics area which refers to the patterns of variation due to any unmodeled factor including group factor and other unwanted factors that could be known or unknown. For instance, people sometimes collect data from multiple sources to generate big data. In this case, the source where a sample comes from is a known group factor that leads to data heterogeneity. Fig. 1 shows an intuitive example demonstrating the impact of data heterogeneity. There are 300 samples lying in a 2-D feature space shown by Fig. 1(a) in which the first 100 samples belongs to Class 1; the next 100 samples belongs to Class 2; and the others Class 3. Each class of samples are generated from a distinct Gaussian distribution with a different mean. Fig. 1(b) shows the distribution of data heterogeneity, where the first half samples and the other half are supposed coming from two different sources and generated via two Gaussian signals with different means. Fig. 1(c) shows the distribution of all the samples under the impact of data heterogeneity from Fig. 1(b). From Fig. 1, we have the following observations: In (a), the samples are clearly separated into three classes based on the features; Moreover, either one of the features can not clearly separate them. In (c), all the samples fall into two clusters that correspond to the two sources instead of the true three classes; The pattern of data variation changed; The samples can be well separated using only the first feature. It demonstrates that in this example the data heterogeneity blurs the true effects of features and leads to spurious classification and feature selection results if standard methods are utilized. This consequently requires the development of new sophisticated methods to take good care of the data heterogeneity for various types of data analyses such as classification and feature selection.

The above example shows the data heterogeneity caused by a known group

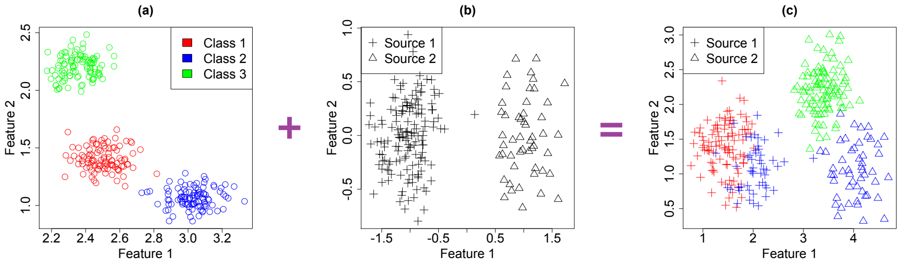

Figure 1: An example showing the impact of heterogeneity. (a) shows the data samples falling into 3 classes in a 2-D space; (b) shows the heterogeneity introduced by two different sources; (c) shows the new data distribution under additional effect from the heterogeneity, which however exhibits a spurious 2-class structure that can be identified using just Feature 1.

factor. However, in most cases, data heterogeneity is usually caused by factors that are unaware or unknown introduced during the data generation step. In bioinformatics area, the microarray data frequently suffer from the unknown heterogeneity that may be either biological or technical in nature. For example, some unaware conditions in the laboratory or chips such as temperature and the amount of ozone in the air are key environmental factors that can affect gene expression values (Fare et al., 2003; Boedigheimer et al., 2008). Speech data can be influenced by the unaware factors such as accent of the speaker as well as the laboratories that the tests were performed in. Image data can also vary subject to unaware lighting conditions. These unknown data heterogeneity poses even more difficulties in developing powerful strategies for analyses of heterogeneous data. In this article, we will focus on tackling the unknown data heterogeneity and develop a powerful embedded feature selection method for heterogeneous data that can be applied for simultaneous feature selection and multi-class classification.

Over the past few decades, many feature selection methods are proposed and have proven to be effective in handling high-dimensional data. Feature selection methods fall into three categories: filter methods, wrapper methods and embedded methods according to their search strategies. Filter methods can achieve high computational efficiency without running any learning algorithms, but may select a set of features that are not optimal as input of a target learning algorithm for subsequent classification. Although wrapper methods attempt to select the optimal features guided by the performance of learning algorithms, the computational cost is very high due to the exponential search space. Embedded methods provide a trade-off solution between filter methods and wrapper methods by embedding feature selection into the model learning. They return both the learned model and selected features simultaneously and are often employed for classification. The most widely used embedded methods are the regularization models such as Lasso (Tibshirani, 1996), sparse linear discriminant analysis (Clemmensen et al., 2011; Wu et al., 2015) and regularized support vector machine (Weston et al., 2000; Zhu et al., 2004; Wang et al., 2006). Many recent sparse learning methods are proposed in form of $\ell_{2,1}$-norm regularized regression models for multi-class classification (Xiang et al., 2012; Du & Shen, 2015; Han et al., 2015). The form of matrix norm has also been extended to $\ell_{2,p}(p \in (0, 1])$ (Wang et al., 2014; Tao et al., 2016) and $\ell_{r,p}(r > 1)$ norms for robust feature selection. There also exists several sparse kernel-based learning methods to improve classification accuracy and feature sparsity. For instance, JCFO (Krishnapuram et al., 2004) seeks sparse kernel basis functions and features by introducing Gaussian priors to their scaling parameters in a Bayesian model for feature selection. Recently, RSFM (Mohsenzadeh et al., 2013) imposes Gaussian priors to both feature weights and sample weights to simultaneously select relevant samples and relevant features. To reduce its high computational complexity, IRSFM (Mohsenzadeh et al., 2016) employs a constructive procedure for model learning which is computationally efficient for data sets with large number of samples. However, most of the above conventional feature selection methods are designed for generic data that are assumed independently and identically distributed (i.i.d.). Heterogenous data violates this simple assumption of data distribution, which calls for sophisticated feature selection methods to take care of the heterogeneity. Leek & Store (2007) have proposed a surrogate variable analysis (SVA) method to adjust unobserved heterogeneity in gene expression analysis. This statistical method attempts to represent the gene expression heterogeneity by the estimated surrogate variables and then use them as covariates while analyzing the association between genes and disease phenotypes. It can be considered as a filter method if the top ranked genes are selected based on their significance scores of association, which unfortunately suffers from the common issues of filter methods. To overcome the well-known issues of variable redundancy and lack of predictability in this filter method, we propose an effective embedded feature selection method from a sparse learning perspective capable of adjusting the unknown data heterogeneity.

Our work has three main contributions: (1) To the best of our knowledge, this is the first embedded feature selection method that takes unknown data heterogeneity into account by explicitly capturing the unknown heterogeneous factors; (2) We derive the corresponding proximal gradient descent algorithm to solve a sparse optimal scoring model on an adjusted data set, which promises good convergency rate and effective updates in each step; (3) The experimental results on three image benchmark data sets have shown the superiority of our selected features in multi-class classification to those features selected from conventional methods ignoring data heterogeneity, especially when a small number of features are selected. The rest of this paper is organized as follows. Section 2 describes the problem and briefly reviews the related work including SVA and optimal scoring; Section 3 describes our feature selection strategy capable of adjusting data heterogeneity. It also introduces an effective proximal gradient algorithm to find the optimal solutions as well as the class-prediction rules for new samples; Section 4 illustrates the classification and feature selection performance of our method on the synthetic data simulated under different extent of data heterogeneity; Section 5 illustrates the classification performance using the selected features from our method comparing to several state-of-the-art methods via the experiments on three benchmark data sets. The sensitivity of the number of heterogeneous factors is also studied for our method; Section 6 concludes the paper.

2. Background

2.1. Problem Statement

Let matrix X denote a data set of n samples and p features and the i-th row of X corresponds to a sample denoted by $\boldsymbol{x}_i$. Each sample is influenced by a response variable of interest and some other unknown signals. Assuming the response variable have C possible choices of classes, we denote the corresponding responses for all the samples by an indicator matrix Y containing only '0' and '1' values in which the element y_{ic} with value '1' denotes that the i-th sample belongs to the c-th class and vice versa. Since the factor of interest in our study is the response variable, those unknown signals will cause unwanted heterogeneity and need to be adjusted.

Problem Definition: *Given the input data X influenced by the response indicator Y and some other unknown factors, find a subset S consisting of t features and a function δ: $\boldsymbol{x}' \rightarrow \boldsymbol{y}'$ such that for a new sample $\boldsymbol{x}'$ represented by the t features, it uniquely assigns a class membership $\boldsymbol{y}'$ to $\boldsymbol{x}'$ and aims to achieve the lowest classification error using the feature set S.*

2.2. Related Work

To solve the target problem, two crucial subproblems need to be addressed: how to capture the unknown data heterogeneity and how to adjust it in a multi-class classifier that is capable of selecting features. Surrogate variable analysis (SVA) proposed by Leek & Store (2007) is an effective statistical method that can estimate the unknown heterogenous factors and adjust their effects in an association study for feature selection. Although this statistical analysis method suffers from the disadvantages of filter method, it provides an effective way to extract the unknown heterogenous factors from the data. As to the second subproblem, we will start by introducing the optimal scoring method (Hastie et al., 1994) that is identical to linear discriminant analysis (LDA) in regression context which enables embedded feature selection for multi-class classification.

2.2.1. Surrogate Variable Analysis—a filter method

The overall goal of SVA is to provide a more accurate and reproducible parsing of the effects of interesting variables and unwanted heterogeneity in an association analysis when data heterogeneity is present (Leek & Store, 2007). Basically it can be described by the following steps in the context of our problem.

Step 1. Remove the effects from the variables of interest (i.e. the response variable).

In many standard statistical analyses, samples are typically assumed collected with random noises. Correspondingly, X is intuitively modeled as:

$$X = Y\Gamma + \Upsilon, \tag{1}$$

where $\Upsilon \in \mathcal{R}^{n\times p}$ is the *i.i.d.* noise term whose element $\epsilon_{ij} \sim N(0, \sigma^2)$. $\Gamma \in \mathcal{R}^{C\times p}$ determines the influences of various responses on all the features. However, in the big data age, there may exist some other unknown factors causing unwanted heterogeneity of the data variation as mentioned previously. This unknown heterogeneity describes patterns of variation due to unmodeled factors that contribute to the variation of measured features in X but are not explicitly included in the intuitive model (1). Assume there are l unknown factors denoted by $U = \{\boldsymbol{u}_m : 1 \leq m \leq l\}$. Consequently, model (1) is corrected as:

$$X = Y\Gamma + U\Psi + \Upsilon \tag{2}$$

for heterogenous data exhibiting heterogeneity caused by unknown factors, where $\Psi \in \mathcal{R}^{l\times p}$ reflects the influences of unknown factors on all the features.

Denote the column space of Y by $\Re_Y$. Then the residual operator of Y that projects onto the orthogonal complement of $\Re_Y$ is denoted by R_Y, i.e., $I - Y(Y^TY)^{-1}Y^T$. Multiply both sides of model (2) by R_Y to obtain

$$\begin{aligned} R_YX &= R_YY\Gamma + R_YU\Psi + R_Y\Upsilon \\ &= R_YU\Psi + R_Y\Upsilon, \end{aligned} \tag{3}$$

which removes the effects from the response variable to facilitate the next study of unknown factors.

Step 2. Obtain signatures of unknown heterogeneity.

The estimation of $\boldsymbol{u}_m (1 \leq m \leq l)$ from equation (3) is a challenging statistical problem. This step estimates signatures $\boldsymbol{h}_m (1 \leq m \leq l)$ instead which represent the residual heterogeneity $R_Y U \Psi$. Any factor analysis method can be applied on $R_Y X$ to produce $\boldsymbol{h}_m$. Singular value decomposition (SVD) is considered here to remove arbitrary. The orthogonal basis of singular vectors are regarded as signatures driven by the unknown factors.

Step 3. Construct unknown factors.

For each signature $\boldsymbol{h}_m$,

1) collect a set of features of X most associated with it;

2) perform SVD on the set and return the eigenvectors $\boldsymbol{e}_j (1 \leq j \leq n)$;

3) let $j^* = \texttt{argmax}_{1 \leq j \leq n} cor(\boldsymbol{e}_j, \boldsymbol{h}_m)$ and set $\hat{\boldsymbol{u}}_m = \boldsymbol{e}_{j^*}$.

The estimated unknown factors $\{\hat{\boldsymbol{u}}_m : (1 \leq m \leq l)\}$ are also regarded as surrogate variables.

Step 4. Association analysis using estimated unknown factors as covariates.

Include all the estimated unknown factors (i.e. surrogate variables) as covariates in the subsequent regression model with a given feature as dependent variable and the response as independent variable. Then, a more accurate statistical significance of each feature can be estimated by adjusting the data heterogeneity.

2.2.2. Optimal Scoring—a flexible multi-class classifier

Optimal scoring is a regression problem equivalent to linear discriminative analysis (LDA). The point of optimal scoring is to turn categorical class variables into quantitative ones by assigning scores to class labels such that the relations between features and classes can be estimated by solving a linear regression problem with constraints. Given a C-vector of scores $\boldsymbol{\theta}$ corresponding to the C classes, $Y\boldsymbol{\theta}$ calculates a vector of response scores for the samples which one may regress on the predictor matrix X. The optimal scoring problem is formulated to estimate such a sequence of $\boldsymbol{\theta}$: $\Theta = \{\boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_k\}$ and the corresponding sequence of regression coefficients $B = \{\boldsymbol{\beta}_1 \cdots \boldsymbol{\beta}_k\}$ as following:

$$\begin{aligned} \min_{\Theta,B} \frac{1}{n} \parallel Y\Theta - XB \parallel_F^2 \\ s.t. \quad \Theta^T Y^T Y \Theta = I, \end{aligned}$$

where $\parallel . \parallel_F$ denotes the Frobenius norm. The above orthogonality constraint ensures that optimal scoring is equivalent to LDA. The sequence of $\boldsymbol{\beta}_k$ are known to be identical to the sequence of LDA discriminant vectors up to scalars (Mardia et al., 1979). Owing to this elegant regression framework, optimal scoring can be easily extended to regularized versions such as penalized optimal scoring (Hastie et al., 1995) to improve the classification performance especially for high-dimensional data.

To solve this optimization problem, Hastie et al. (1994) proposed the following algorithm:

(1) Choose an initial score matrix Θ_0 satisfying $\Theta_0^T Y^T Y \Theta_0 = I$.

(2) Fit a multiple regression model of $Y\Theta$ on X, yielding fitted values $\hat{B} = (X^T X)^{-1} X^T Y \Theta_0$.

(3) Obtain eigenvector matrix Φ of $\Theta_0^T Y^T X (X^T X)^{-1} X^T Y \Theta_0$; the optimal scores are $\hat{\Theta} = \Theta_0 \Phi$.

(4) Update $\hat{B}$ by $\hat{B}\Phi$.

Define D as a diagonal matrix with the k-th diagonal term as:

$$D_{kk} = \{\frac{1}{\alpha_k^2 (1 - \alpha_k^2)}\}^{\frac{1}{2}},$$

where α_k is the k-th largest eigenvalue calculated in step (3). The decision rule for a new sample $\boldsymbol{x'}$ is to assign it to class c that minimizes:

$$\parallel D\hat{B}^T(\boldsymbol{x'} - \boldsymbol{\mu}^c) \parallel^2,$$

where $\boldsymbol{\mu}^c$ denotes the centroid of the c-th class.

Optimal scoring is an effective and flexible tool for multi-class classification due to its equivalence to LDA and flexible regression framework that enables model regularization and feature selection.

3. Methodology

Due to the feature redundancy and lack of predictability issue of filter method SVA, we aim to develop an embedded feature selection method that can accomplish both feature selection and classification in a regularized regression framework based on the adjusted data. We believe that the variation of the input data comes from two sources: the response variable $\boldsymbol{y}$ and other unknown factors U which leads to the unwanted heterogeneity that deteriorates the performance of classification and feature selection. Correspondingly, the input data X is modeled as equation (2):

$$X = Y\Gamma + U\Psi + \Upsilon.$$

Our strategy contains two main procedures: 1) remove the variation from the source of unknown factors U and build an adjusted data set X_a whose variation is only determined by the response Y; and 2) select features with a sparse learning model of multi-class classification based on X_a .

3.1. Removing unknown data heterogeneity

In the first procedure, we need to first estimate the unknown factors U from the data model (2) which however is generally impossible to directly accomplish due to the unidentifiability problem explained as below. Let A denote any invertible $l \times l$ matrix. Then

$$(UA)(A^{-1}\Psi) = U\Psi,$$

so neither U or Ψ are identifiable. Alternatively, a feasible intermediate step is introduced to first estimate the corresponding signatures for the unknown factors $\boldsymbol{u}_m (1 \leq m \leq l)$ to represent the signals that are independent of Y. These signatures are easy to be estimated. One should notice that the signatures are neither necessarily physically meaningful nor exactly U since U is not necessarily independent of Y. To allow for physical meanings and potential overlap with the response variable Y, each $\boldsymbol{u}_m$ is extracted from a set of original data variables most correlated with its corresponding signature. The detailed procedure of estimating U is implemented using the same first three steps of SVA illustrated in Section 2.2. Consequently, the true associations between features and the response variable are then assumed hidden in the adjusted data X_a obtained as $X - U\Psi$ by removing the unknown heterogeneity.

As the unknown heterogeneity is usually introduced during the data collection or experiment design phase which directly affect the data variation pattern, there is no direct causal relationship between the unknown factors and response variables. Therefore, the adjustment for unknown heterogeneity can be performed prior to the subsequent analyses that involve response variables. This separate adjustment procedure allows for the alleviation of computational burden in the down-stream analyses and the flexibility of study of feature selection and classification.

3.2. Sparse optimal scoring based on the adjusted data set

To select a set of features that can achieve the best classification performance, we develop an embedded feature selection method on the adjusted data set to estimate the true effects of features in separating the classes. As we mentioned in the related work, optimal scoring is an attractive multi-class classification approach due to its equivalent performance to LDA and flexible regression framework. The extension of optimal scoring to a regularized version for sparse learning enables embedded feature selection and is more natural than the extension of LDA to sparse LDA. We perform a sparse optimal scoring method on the adjusted data set, which turns the response indicator Y into a sequence of quantitive uni-variate outcomes by assigning a sequence of scores stored in Θ to each corresponding class and then performs regularized regression of the sequence of quantitive outcomes on X_a. We name our method as sparse optimal scoring with adjustment (SOSA). Correspondingly, the embedded feature selection by SOSA is formulated as:

$$\begin{aligned} &\min_{B,\Theta} \frac{1}{n}||X_a B - Y\Theta||_F^2 + \lambda\Omega(B) \\ &\qquad\qquad s.t. \quad \Theta^T Y^T Y \Theta = I, \end{aligned} \tag{4}$$

in which $\Omega(.)$ is a regularization function of a matrix that sums up its row norms, i.e, $\Omega(B) = \sum_{j=1}^{p} ||\boldsymbol{b}_j||_2$ where $\boldsymbol{b}_j$ is a row vector denoting the j-th row of the $p \times k$ projection matrix B. With an appropriate choice of λ, this model allows for feature selection by shrinking some $\boldsymbol{b}_j$ exactly to a zero vector which suggests that the j-th feature is not selected. Our SOSA has the similar formulation as the sparse optimal scoring model in (Leng, 2008) but with a different design matrix containing adjusted features exempt from the unknown heterogeneity. In our model, the fact that our design matrix X_a lacks orthonormality further poses difficulty to solve (4).

Existing algorithms for solving sparse optimal scoring problem regard it as a group lasso problem by considering each row of B as a group and reshaping B to a long vector containing p groups. However, those strategies solving group lasso problem such as least angle regression selection (LARS) (Yuan & Lin, 2006) or extension of shooting algorithm based on Karush−Kuhn−Tucker (KKT) conditions require the design matrix to be orthonormal. The general situation in adopting these strategies to solve group lasso/sparse optimal scoring problem is to orthonormalize the data first and then solve the problem in terms of the new data, which however leads to a different problem inequivalent to the original one resulting in irrelevant solutions that are not able to reverse back. Although an alternative strategy is proposed by Simon & Tibshirani (2012) to extend group lasso to general data by penalizing the fit of group for model selection, unfortunately this alternative strategy does not work for the sparse optimal scoring problem. We propose an algorithm using proximal gradient to solve SOSA for general design matrix X_a that is not necessarily orthonormal.

To solve problem (4), we iteratively update B and Θ until convergence.

(1) update B**.** Given Θ, we update B by solving the following subproblem:

$$\min_{B} \frac{1}{n} ||X_a B - Y\Theta||_F^2 + \lambda \Omega(B). \tag{5}$$

We propose a new algorithm to solve problem (5) by proximal operator Moreau (1962) through an equivalent problem transformation.

Lemma 1. *Consider a vector* $\boldsymbol{w} = \begin{pmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_k \end{pmatrix}_{pk\times 1}$ *where* $\boldsymbol{v}_s (1 \leq s \leq k)$ *is a p-length vector. Let* $\boldsymbol{\theta}_s$ *denote the s-th column of* Θ *and* ψ *be the* $l_{2,1}$*-norm mapping:* $\boldsymbol{w} \mapsto \sum_{g\in\mathcal{G}} \parallel \boldsymbol{w}_g \parallel_2$ *in which* $\mathcal{G}$ *is a partition of* $\{1, \ldots, pk\}$ *defined by a set* $\Big\{ \{j+(s-1)p \,|\, 1 \leq s \leq k\} \,\Big|\, 1 \leq j \leq p \Big\}$ *and* $\boldsymbol{w}_g$ *denotes a k-length vector storing the elements of* $\boldsymbol{w}$ *indexed by g in* $\mathcal{G}$*. Consider a* $l_{2,1}$*-norm regularized problem*

$$\min_{\boldsymbol{w}} \frac{1}{n} \sum_{s=1}^{k} ||X_a \boldsymbol{v}_s - Y\boldsymbol{\theta}_s||_2^2 + \lambda\psi(\boldsymbol{w}), \tag{6}$$

then its solutions $\hat{\boldsymbol{w}}$ *have these connections with* (5)*'s solutions* $\hat{B}$*:*

$$\hat{\boldsymbol{w}} = \begin{pmatrix} \tilde{\boldsymbol{b}}_1 \\ \vdots \\ \tilde{\boldsymbol{b}}_k \end{pmatrix}_{pk\times 1}, \qquad \hat{\boldsymbol{w}}_g = \hat{\boldsymbol{b}}_j,$$

where $\tilde{\boldsymbol{b}}_s (1 \leq s \leq k)$ is the s-th column vector of $\hat{B}$ and $\hat{\boldsymbol{b}}_j$ is the j-th row vector of $\hat{B}$.

Proof. The relationship holds since (5) and (6) are equivalent which can be easily proved by linear algebra. □

By re-expressing (5), we arrive at its equivalent problem (6) with a nice formulation that can be solved by proximal gradient (Bach et al., 2011) which is tailored to solve convex optimization problem of the following general form:

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) + \lambda P(\boldsymbol{w}), \tag{7}$$

where $f(.)$ is a convex differentiable function and P is typically a non-smooth and non-Euclidean norm that induces sparsity. Consider a quadratic approximation of function $f(\boldsymbol{w})$ which turns (7) into

$$\begin{aligned} & \min_{\boldsymbol{w}} f(\boldsymbol{w}^t) + \nabla f(\boldsymbol{w}^t)^T(\boldsymbol{w} - \boldsymbol{w}^t) + \frac{L}{2} \parallel \boldsymbol{w} - \boldsymbol{w}^t \parallel_2^2 + \lambda P(\boldsymbol{w}) \\ = \;& \min_{\boldsymbol{w}} \frac{1}{2} \parallel \boldsymbol{w} - (\boldsymbol{w}^t - \frac{1}{L}\nabla f(\boldsymbol{w}^t)) \parallel_2^2 + \frac{\lambda}{L} P(\boldsymbol{w}), \end{aligned}$$

where $L > 0$ is an upper bound on the Lipschitz constant of ∇f. Then, the general proximal gradient update rule (Moreau, 1962) is:

$$\boldsymbol{w}^{t+1} = prox_{\frac{\lambda}{L}P}(\boldsymbol{u}), \quad where \quad \boldsymbol{u} = \boldsymbol{w}^t - \frac{1}{L}\nabla f(\boldsymbol{w}^t). \tag{8}$$

Theorem 1. *The proximal gradient update rules for solving problem* (5) *are*

$$\boldsymbol{b}_j^{t+1} = (1 - \frac{\frac{\lambda}{2d}}{\| q(\boldsymbol{b}_j^t) \|_2})_+ q(\boldsymbol{b}_j^t), \quad 1 \le j \le p,$$

where d is the largest eigenvalue of X_a; and

$$q(\boldsymbol{b}_j^t) \quad = \quad \boldsymbol{b}_j^t - \frac{1}{nd}\Big(B^{tT} X_a^T - \Theta^T Y^T\Big)(X_a)_{:j}.$$

Proof. We start by finding the proximal gradient update rules for its equivalent problem (6), in which we have

$$\begin{aligned} f(\boldsymbol{w}) &= \frac{1}{n}\sum_{s=1}^{k} ||X_a \boldsymbol{v}_s - Y\boldsymbol{\theta}_s||_2^2, \\ P &= \psi. \end{aligned}$$

The first derivative of $f(\boldsymbol{w})$ is computed as

$$\nabla f(\boldsymbol{w}) = \frac{2}{n}\begin{pmatrix} X_a^T X_a \boldsymbol{v}_1 - X_a^T Y \boldsymbol{\theta}_1 \\ \vdots \\ X_a^T X_a \boldsymbol{v}_k - X_a^T Y \boldsymbol{\theta}_k \end{pmatrix}_{pk \times 1}$$

and the second derivative of $f(\boldsymbol{w})$ is

$$H = \frac{2}{n}\begin{pmatrix} [X_a^T X_a]_{p \times p} & & \\ & \ddots & \\ & & [X_a^T X_a]_{p \times p} \end{pmatrix}_{pk \times pk} .$$

L is then set to the (smallest) Lipschitz constant of ∇f:

$$L = 2e_{max}(H) = 2e_{max}(X_a) = 2d,$$

where $e_{max}(.)$ denotes the largest eigenvalue of a matrix and d represents the largest eigenvalue of X_a.

The proximal mapping for the $l_{2,1}$-norm ψ with penalty λ is computed as

$$[prox_{\lambda\psi}(\boldsymbol{u})]_g = (1 - \frac{\lambda}{\parallel \boldsymbol{u}_g \parallel_2})_+ \boldsymbol{u}_g, \quad g \in \mathcal{G},$$

according to Combettes & Wajs (2006) and Bach et al. (2011). By substituting $\nabla f(\boldsymbol{w})$, L and the proximal mapping function into the general proximal gradient update rules (8), we have the proximal gradient update rules for problem (6) as:

$$\boldsymbol{w}_g^{t+1} = [prox_{\frac{\lambda}{L}\psi}(\boldsymbol{u})]_g = (1 - \frac{\frac{\lambda}{2d}}{\parallel \boldsymbol{u}_g \parallel_2})_+ \boldsymbol{u}_g, \quad g \in \mathcal{G},$$

where

$$\begin{aligned} \boldsymbol{u}_g &= \boldsymbol{w}_g^t - \frac{1}{L}[\nabla f(\boldsymbol{w}^t)]_g \\ &= \boldsymbol{w}_g^t - \frac{1}{nd}\begin{pmatrix} (X_a^T)_{j:}X_a\boldsymbol{v}_1^t - (X_a^T)_{j:}Y\boldsymbol{\theta}_1 \\ \vdots \\ (X_a^T)_{j:}X_a\boldsymbol{v}_k^t - (X_a^T)_{j:}Y\boldsymbol{\theta}_k \end{pmatrix}_{k\times 1} \\ &= \boldsymbol{w}_g^t - \frac{1}{nd}\Big((X_a^T)_{j:}X_a[\boldsymbol{v}_1^t, \ldots, \boldsymbol{v}_k^t] - (X_a^T)_{j:}Y\Theta\Big)^T \\ &= \boldsymbol{w}_g^t - \frac{1}{nd}\Big([\boldsymbol{v}_1^t, \ldots, \boldsymbol{v}_k^t]^T X_a^T - \Theta^T Y^T\Big)(X_a)_{:j}. \end{aligned}$$

Then, the proximal gradient update rules for problem (5) can be easily obtained using the connections between $\hat{\boldsymbol{w}}_g$ and $\hat{\boldsymbol{b}}_j$; $\hat{\boldsymbol{v}}_s$ and $\tilde{\boldsymbol{b}}_s$ stated by **Lemma 1**. □

(2) update Θ. In each $(t+1)$-th iteration, given fixed B^t, we update Θ^{t+1} by solving the following subproblem:

$$\begin{aligned} &\min_{\Theta} \frac{1}{n}||X_aB^t - Y\Theta||_F^2 \\ &s.t. \quad \Theta^T Y^T Y \Theta = I. \end{aligned}$$

By introducing a new variable Θ' that equals $(Y^TY)^{\frac{1}{2}}\Theta$, we transform this subproblem to the following equivalent optimization problem with respect to

Θ':

$$\begin{aligned}
& \min_{\Theta'} \frac{1}{n}||X_a B^t - Y(Y^TY)^{-\frac{1}{2}}\Theta'||_F^2 \\
= \quad & \min_{\Theta'} \frac{1}{n}||(Y^TY)^{-\frac{1}{2}}Y^TX_aB^t - \Theta'||_F^2 \\
& s.t. \quad \Theta'^T\Theta' = I.
\end{aligned}$$

Θ' can be updated according to **Theorem 2** proposed by Zou et al. (2006) given below:

Theorem 2. *Reduced-Rank Procrustes Rotation. Given two matrices $M_{N\times D}$ and $N_{N\times L}$, consider the constrained minimization problem*

$$\min_A ||M - NA^T||^2 \quad s.t. \quad A^TA = I_{L\times L}.$$

Suppose that the Singular Value Decomposition (SVD) of M^TN is in the form of UDV^T, then $\hat{A} = UV^T$.

Denote Q as $(Y^TY)^{-\frac{1}{2}}Y^TX_aB^t$. Compute SVD of $Q = R\Lambda V^T$. According to **Theorem 2**, Θ' is updated by RV^T. Then, we have

$$\Theta^{t+1} = (Y^TY)^{-\frac{1}{2}}RV^T.$$

We alternatively update B and Θ until the change of objective function value of (4) is less than a predefined small threshold. In summary, the full procedure for feature selection by SOSA is given in details as **Algorithm 1**. In this algorithm, Stage I takes $O(n^2p + npl)$ computational operations to estimate and remove the heterogeneity. In Stage II, computing M and N has the computational complexity $O(npg)$. The eigen decomposition of X_a has the complexity $O(n^2p)$. The update of Θ and B has complexity $O(npk)$ in each iteration. In total, the computational complexity of the whole procedure is $O(n^2p + rnpk)$ if it takes r iterations to converge.

3.3. Predicting the response for a new sample

Analogous to optimal scoring, our SOSA has the similar decision rule for multi-class classification which however is applied in the adjusted feature space.

Algorithm 1 The algorithm for solving SOSA.

```
**Input:** $X \in R^{n\times p}$; $l$: the desired number of heterogeneous factors; $k$: the desired dimension of projection space; $\lambda$: the tuning parameter for regularization; and the initial estimates of $B$ and $\Theta$.

// *Stage I: remove unknown data heterogeneity.*
1. Calculate $R_Y$ as $I - Y(Y^TY)^{-1}Y^T$ and set the left $l$ eigenvectors of $R_YX$ as $\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_l\}$.
  for $m = 1 : l$,
      collect a set of features of $X$ most associated with $\boldsymbol{h}_m$;
      perform SVD on the set and return the eigenvectors $\boldsymbol{e}_j (1 \leq j \leq n)$;
      let $j^* = \texttt{argmax}_{1\leq j\leq n} cor(\boldsymbol{e}_j, \boldsymbol{h}_m)$ and set $\hat{\boldsymbol{u}}_m = \boldsymbol{e}_{j^*}$.
  end
2. Let $\hat{U} = \{\hat{\boldsymbol{u}}_1, ..., \hat{\boldsymbol{u}}_l\}$. Calculate $\hat{\Psi}$ as $(\hat{U}^T R_Y \hat{U})^{-1}\hat{U}^T R_Y X$ and set $X_a = X - \hat{U}\hat{\Psi}$.

// *Stage II: embedded feature selection on the adjusted data.*
3. Calculate $M = (Y^TY)^{-\frac{1}{2}}Y^TX_a$ and $N = X_a^TY$. Do eigen decomposition of $X_a$ and set the largest eigen value as $d$.
4. Denote $Q$ as $MB$. Calculate the SVD of $Q = R\Lambda V^T$ and update $\Theta$ by $(Y^TY)^{-\frac{1}{2}}RV^T$.
5. Calculate $P = B^TX^T$.
  for $j = 1, \ldots, p$,
      $\boldsymbol{t} = [B^T]_{:j} - \frac{1}{nd}(PX_{:j} - \Theta^T[N^T]_{:j})$
      update $B_{j:}$ by $(1 - \frac{\lambda}{2d\|\boldsymbol{t}\|_2})_+\boldsymbol{t}$.
   end
6. Repeat 4-5 until convergence.
7. Return $\hat{U}$, $\hat{\Psi}$, $\hat{\Theta}$ and $\hat{B}$.
```

One critical problem we are facing is how to derive the adjusted features for a given new sample $\boldsymbol{x}'$ such that we are able to employ the decision rule in the adjusted feature space to predict the class label. Given the learnt coefficients $\hat{\Psi}$ and $\hat{\Gamma}$, we can derive the corresponding heterogeneous factor $\boldsymbol{u}'$ for $\boldsymbol{x}'$ based on the following equation:

$$\boldsymbol{x}' = \hat{\Gamma}^T \boldsymbol{y}' + \hat{\Psi}^T \boldsymbol{u}' + \boldsymbol{\epsilon}, \tag{9}$$

where the heterogeneous factor $\boldsymbol{u}'$ and the C-length vector $\boldsymbol{y}'$ are unknown. Compute $R_{\hat{\Gamma}^T} = I - \hat{\Gamma}^T(\hat{\Gamma}\hat{\Gamma}^T)^{-1}\hat{\Gamma}$. Multiply both sides of (9) by $R_{\hat{\Gamma}^T}$ to obtain

$$R_{\hat{\Gamma}^T}\boldsymbol{x}' = R_{\hat{\Gamma}^T}\hat{\Psi}^T\boldsymbol{u}' + R_{\hat{\Gamma}^T}\boldsymbol{\epsilon}.$$

Thus, $\boldsymbol{u}'$ is estimated as

$$\hat{\boldsymbol{u}}' = (\hat{\Psi}R_{\hat{\Gamma}^T}\hat{\Psi}^T)^{-1}\hat{\Psi}R_{\hat{\Gamma}^T}\boldsymbol{x}'$$

and then we have

$$\boldsymbol{x}_a{}' = \boldsymbol{x}' - \hat{\Psi}^T\hat{\boldsymbol{u}}'.$$

Obtain the largest k eigen values of $\hat{\Theta}^T Y^T X_a (X_a^T X_a)^{-1} X_a^T Y \hat{\Theta}$. Define D as a diagonal matrix with the k-th diagonal term:

$$D_{kk} = \{\frac{1}{\alpha_k^2(1-\alpha_k^2)}\}^{\frac{1}{2}},$$

where α_k is the k-th largest eigenvalue calculated. The decision rule for a new sample $\boldsymbol{x}'$ is to assign it to class c that minimizes:

$$\parallel D\hat{B}^T(\boldsymbol{x}_a{}' - \boldsymbol{\mu}^c) \parallel^2,$$

where $\boldsymbol{\mu}^c = \sum_{\boldsymbol{y}_i=c}(X_a)_{i:}/n_c$ denotes the centroid of the c-th class.

4. Simulation Study

In this section, we investigate the performance of our algorithm on synthetic data and focus on the study of the impact of data heterogeneity on classification and feature selection performance. The synthetic data is simulated based on our previously introduced model (2):

$$X = Y\Gamma + U\Psi + \Upsilon.$$

We simulated $n = 100$ samples containing $p = 5,000$ features. The samples are assumed having a potential class structure represented by a $n \times C$ matrix Y which however is affected by l additional heterogeneous factors stored in a $n \times l$ matrix U. We set C to 10 and labeled every 10 samples from 1 to 10 for simplicity. The first 100 features are assumed to be able to discriminate the C classes while the remaining ones are redundant features. Since the c-th row of Γ stores the influences of class c on all the features, we sampled the first 100 elements of c-th row of Γ from a normal distribution with zero mean and the standard deviation as s_c which is sampled uniformly from the range of $0.01 - 0.1$. The other elements in Γ is sampled from a normal distribution with zero mean and the standard deviation as 0.005 to discriminate the true features and redundant features. For the m-th row of Ψ that stores the effects of the m-th heterogeneous factor on all the features, we sampled its elements from a normal distribution with mean as μ and the standard deviation as s_m which is sampled uniformly from the range of $0.01 - 0.1$. By varying μ and the number of heterogenous factors l, we can control the strength of data heterogeneity. We sampled U from a multivariate normal distribution $N(\mathbf{0}, I_{l\times l})$. Then, the columns of U are orthogonalized to assure that the heterogeneous factors are not correlated. Υ is the random noise which is sampled from a multivariate normal distribution $N(\mathbf{0}, 0.01 * I_{l\times l})$. Finally, X is obtained based on model (2).

We run our SOSA on several settings of the synthetic data, comparing with two embedded feature selection methods without adjusting the unknown data heterogeneity and a filter method accounting for the data heterogeneity:

1) L1-SVM (Fan et al., 2008) — an embeded feature selection method by imposing ℓ_1 regularization on the coefficients of the popular standard SVM for multi-class classification;

2) SOS (sparse optimal scoring) — an embeded feature selection method

built by using the original data X instead of the adjusted data X_a as the input for model (4). Consequently, SOS employs the same classification model as SOSA, but the only difference is that SOS lacks the adjustment of data heterogeneity. This will ensure a fair comparison of the performance in adjusting data heterogeneity;

-3) SVA (surrogate variable analysis) (Leek & Store, 2007) — a representative filter method capable of taking care of unknown data heterogeneity.

In addition, we also considered the classification performance using all the features as baseline. To guarantee the fairness of comparison of feature selection efficacy, we examined the resulted classification performance based on an uniform classifier: 1-nearest neighbor for all the methods using their selected features respectively. In this simulation study, we randomly sampled 10% samples as the testing samples and the others are regarded as training samples. The classification performance is evaluated by the average classification testing error over 100 simulations. The feature selection performance is evaluated by the hit ratio which is calculated as the percentage of the 100 true features that are correctly selected. Table 1 reports the hit ratio and classification error rate respectively for all the methods under 3 different choices of μ and 5 different choices of l. As we can see, with increasing number of heterogeneous factors or heterogeneity effects, the error rate of all the methods will increase except some cases of SOSA. Using all features for classification suffer badly from the data heterogeneity, especially when l and μ are moderate or large. Although SVA can adjust these effect, its classification and feature selection performance are still not satisfying due to the disadvantages of filtering method. Both SOS and L1-SVM can achieve lower error rate and higher hit ratio benefit from their sparse learning models. In some settings, L1-SVM can perform better in both criteria than SOS probably because of its kernel-based learning framework can somehow help alleviate the data heterogeneity. Our SOSA performs the best out of all the settings in both criteria due to the appropriate adjustment of data heterogeneity and sparse learning approach. We also observed that with increasing number of heterogeneous factors or heterogeneity effects, the hit ratio

		μ=0.1		μ=0.3		μ=0.5	
l	Method	Hit(%)	Error(%)	Hit(%)	Error(%)	Hit(%)	Error(%)
1	baseline	-(-)	4.00(0.95)	-(-)	21.0(1.72)	-(-)	36.3(2.36)
	SVA	27(0.93)	5.00(1.16)	26.9(0.91)	9.00(1.25)	26.9(0.91)	13.7(1.51)
	SOSA	**99.5(0.16)**	**0.00(0.00)**	**99.5(0.16)**	**0.00(0.00)**	**99.5(0.17)**	**0.00(0.00)**
	SOS	90.2(0.94)	0.00(0.00)	97.8(0.32)	0.67(0.36)	96.2(0.34)	3.67(0.87)
	L1-SVM	60.5(1.24)	0.00(0.00)	70.2(1.53)	1.33(0.49)	79.1(1.64)	3.67(0.95)
5	baseline	-(-)	58.6(2.12)	-(-)	64.3(1.73)	-(-)	70.0(1.89)
	SVA	22.1(1.17)	22.0(2.58)	22.1(1.21)	31.3(2.89)	23.0(1.23)	39.0(2.79)
	SOSA	**99.2(0.14)**	**0.00(0.00)**	**99.1(0.14)**	**0.00(0.00)**	**99.4(0.14)**	**0.00(0.00)**
	SOS	98.2(0.19)	1.00(0.43)	92.7(0.64)	6.33(1.08)	78.4(1.31)	16.0(1.51)
	L1-SVM	69.7(1.06)	1.67(0.65)	83.0(0.87)	5.33(1.16)	85.5(0.55)	15.0(1.47)
10	baseline	-(-)	77.0(2.20)	-(-)	80.0(1.62)	-(-)	81.3(1.65)
	SVA	24.0(0.89)	31.3(2.92)	25.7(0.90)	38.0(2.47)	29.1(0.98)	43.7(2.33)
	SOSA	**98.1(0.27)**	**0.00(0.00)**	**98.8(0.22)**	**0.00(0.00)**	**99.3(0.14)**	**3.00(2.09)**
	SOS	98.3(0.17)	2.00(0.57)	85.5(0.73)	14.3(1.52)	60.7(1.39)	29.3(2.38)
	L1-SVM	71.7(1.39)	3.33(0.77)	81.9(1.05)	13.7(1.59)	82.1(0.73)	21.3(2.09)
15	baseline	-(-)	77.0(2.14)	-(-)	77.0(2.20)	-(-)	78.3(1.86)
	SVA	22.2(0.93)	37.0(2.97)	26.0(1.01)	39.0(2.55)	31.8(1.01)	43.7(2.21)
	SOSA	**97.7(0.35)**	**0.00(0.00)**	**99.5(0.12)**	**0.33(0.34)**	**98.4(0.30)**	**2.67(1.43)**
	SOS	97.1(0.34)	1.67(0.54)	77.1(1.21)	16.7(1.83)	47.1(1.58)	34.7(2.16)
	L1-SVM	75.4(1.04)	3.33(0.93)	82.8(0.68)	15.0(1.77)	82.4(0.49)	25.0(1.88)

Table 1: Comparison of the performance of SVA, SOS, SOSA and L1-SVM on synthetic data with 5,000 features under different numbers of heterogeneous factors and different levels of heterogeneity effect. The average (standard deviation) of hit ratio, classification error rate, running time over 100 simulations are presented for these methods as well as the baseline using all the features.

for SVA and L1-SVM somehow increases, which probably is influenced by the bad feature selection strategy or inappropriate adjustment of heterogeneity. In summary, our SOSA can achieve significant improvement in the classification and feature selection performance to other methods by appropriately adjusting the data heterogeneity in a sparse learning model. Moreover, this superiority is consistent among various cases corresponding to different extent of data heterogeneity.

5. Experiments on Real-world Data

5.1. Data sets

We conducted our experiments on three benchmark data sets whose important statistics are summarized by Table 2.

The first one is ORL face database (Samaria & Harter, 1994) which consists of a total of 400 face images. There are ten different grey images of each of 40 distinct subjects. For some subjects, the images were captured at different times under varying conditions such as the lighting, facial expressions (open or closed eyes, smiling or not smiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. We use the normalized (in scale and orientation) images processed by He et al. (2005) such that the two eyes were aligned at the same position. They cropped the facial areas into the final images for matching. The size of each cropped image is 32×32 pixels. Correspondingly, each face image can be represented by a 1,024-dimensional vector.

The second one is COIL20 image library (Nene et al., 1996) from Columbia which contains 20 objects. There are 72 grey images for each object taking 5 degrees apart as the object is rotated on a turn table. The size of each image is 32×32 pixels. Correspondingly, each object image can be represented by a 1,024-dimensional vector.

The third one is the UCI DrivFace data which contains facial images sequences of subjects while driving in real scenarios (KaterineDiaz-Chito et al., 2016). It is composed of 606 samples of 80×80 pixels each, acquired over different days from 4 drivers (2 women and 2 men) with several facial features like beard or glasses. Each driver can have three possible gaze direction: looking-right, frontal and looking-left. Correspondingly, each facial image can be represented by a 6,400-dimensional vector. Note that this data set has different class sample size in a range from 90 to 179. It is unbalanced while the other two data sets are balanced with equal number of samples for each class.

Data set	No. of samples	No. of features	No. of classes	No. of samples per class
ORL	400	1,024	40	10
COIL20	1,440	1,024	20	72
DrivFace	606	6,400	4	90-179

Table 2: Statistics of three benchmark data sets.

5.2. Experimental setting

In order to evaluate the multi-class feature selection and classification performance more thoroughly, we studied several settings corresponding to different number of data samples and different number of classes. Specifically, for each data set, we selected all the samples of C random classes out of the total classes and then evaluate the C-class feature selection and classification performance on the corresponding data subset. Knowing that the entire ORL, COIL20 and DrivFace data consist of 40, 20 and 4 respective classes, we set $C = (10, 20, 30, 40)$ for ORL; $C = (5, 10, 15, 20)$ for COIL20; and $C = (2, 4)$ for DrivFace. This thus leads to the corresponding data subsets of $(100, 200, 300, 400)$ samples for ORL; $(360, 720, 1080, 1440)$ samples for COIL20; and $(257, 606)$ samples for DrivFace.

We show the effectiveness of our proposed SOSA by evaluating its performance in feature selection and classification, comparing with L1-SVM, SOS and SVA introduced in the simulation study. In addition, we also considered the classification performance using all the features as baseline. To guarantee the fairness of comparison of feature selection efficacy, we examined the resulted classification performance based on an uniform classifier: 1-nearest neighbor for all the methods using their selected features respectively. Their classification performance were evaluated by the criterion of classification testing error rate based on cross-validation.

For each setting of each data set, we employed L1-SVM, SVA, SOS and our SOSA to select t features and then evaluated the classification performance for each method based on their selected t features using 10-fold cross validation.

In detail, for SVA, we ranked all the features and selected the top t features according to their p-values calculated using F-test which compares the standard model (1) and the corrected model (2) using heterogeneous factors as covariates. To realize L1-SVM, we implemented ℓ_1-regularized hinge-loss support vector multi-class classification by applying the LiblineaR package from Fan et al. (2008). Since each feature has C sparse coefficients specifying its contribution to the C respective classes, we define for each feature a score as the maximum absolute value of its coefficients in all the C classes. Then, the top t features are selected by L1-SVM based on the score ranking. For SOS and SOSA, we selected the t features whose coefficients have non-zero ℓ_2-norms while the others have exactly zero ℓ_2-norms. This can be achieved by controlling the penalization tuning parameter λ for the $\ell_{2,1}$-norm regularization to force the coefficients of other features in the C classes are all zeros. We search λ in the range of 0.001 to 0.1. As for the cross validation, we trained 90 percent of a given setting of the data to select the best t features in each fold and then calculated the classification testing error rate by predicting the labels of the remaining 10 percent of the samples using 1-nearest neighbor based on the respective t features selected by each method. Note that the training samples were chosen from each class by 90 percent. In our experiments, t is set to 15 different numbers: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 120, 140, 160, 180, 200. We set the number of heterogeneous factors as 3, 3 and 5 for ORL, COIL20 and DrivFace respectively. The selection of this l will be discussed in Section 5.4. The dimension of projection space (k) was set to $C-1$ for data with C classes for simplicity.

5.3. Classification results using the selected features

Fig. 2∼ 4 show the plots of average classification testing error rate over 10 folds versus the number of selected features(t) on ORL, COIL20 and Drivface respectively. In each figure, (a), (b), (c) and (d) show the respective plot for different settings of C-class classification with different number of samples. As we can see, our proposed SOSA consistently outperforms all its competitors for all the settings of all the data sets. It is interesting to note that our SOSA

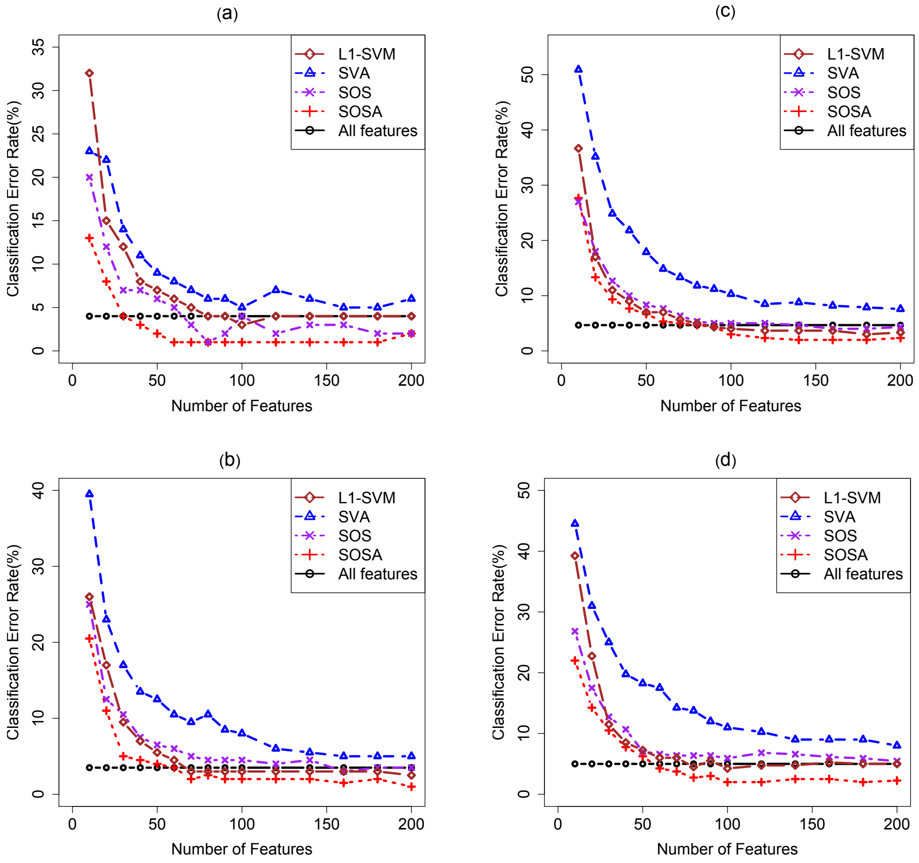

Figure 2: Comparison of the classification performance on ORL using different number of features selected from L1-SVM, SVA, SOS and SOSA. (a) 10-class classification with 100 samples; (b) 20-class classification with 200 samples; (c) 30-class classification with 300 samples; (d) 40-class classification with 400 samples. The black lines show the results using all the 1024 features.

performs surprisingly well on ORL and DrivFace, even better than using all the features in most cases by accounting for the data heterogeneity. For the 5-class subset of COIL20 data, SOSA can achieve comparative performance to the 530 baseline using only 20 out of 1,024 features. For several subsets(i.e., 10,15,20 classes) of COIL20 where their number of features is less than or comparative to

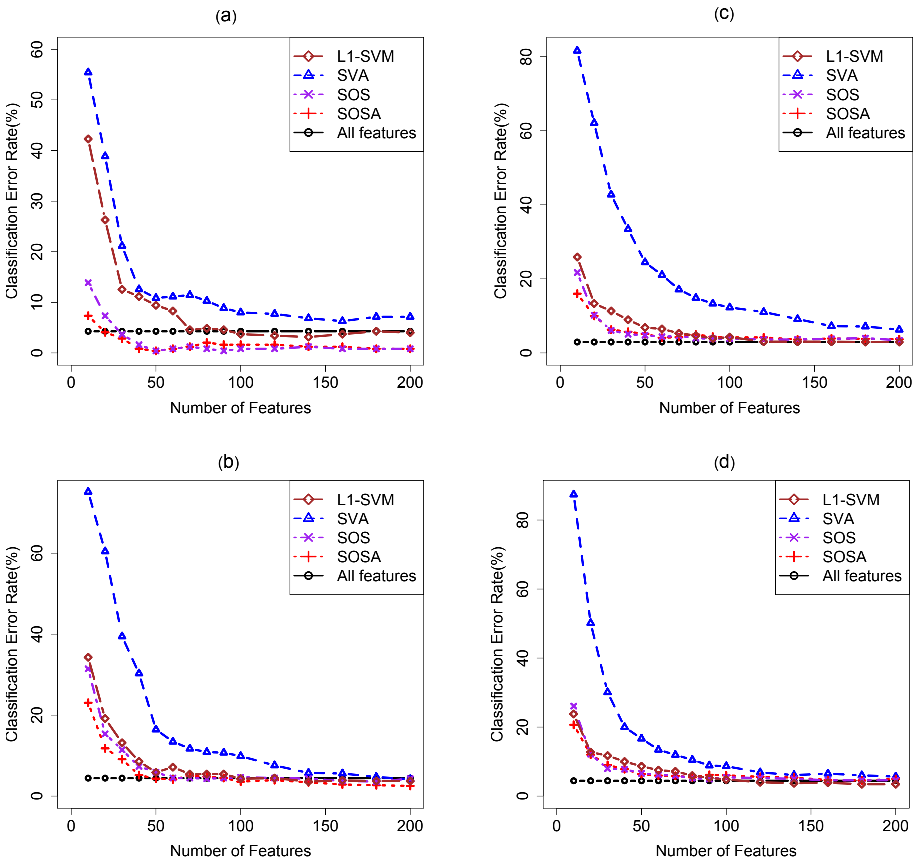

Figure 3: Comparison of the classification performance on COIL20 using different number of features selected from L1-SVM, SVA, SOS and SOSA. (a) 5-class classification with 360 samples; (b) 10-class classification with 720 samples; (c) 15-class classification with 1,080 samples; (d) 20-class classification with 1,440 samples. The black lines show the results using all the 1,024 features.

that of samples, it is not surprising that all the feature selection methods can not beat the baseline but they can achieve at least comparative performance using less than 50 or 200 features. The improvement of classification performance is usually larger when 10 $\sim$ 50 features were selected for the first two small data sets, which further implies its superior feature selection efficacy. SOSA and

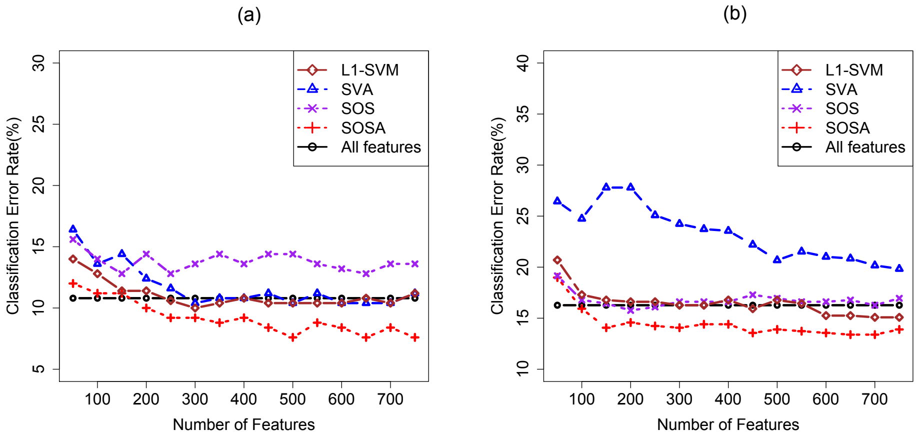

Figure 4: Comparison of the classification performance on DrivFace using different number of features selected from L1-SVM, SVA, SOS and SOSA. (a) 2-class classification with unbalanced 257 samples; (b) 4-class classification with unbalanced 606 samples. The black lines show the results using all the 6,400 features.

SOS approach to the best results using much less features than other methods, typically to the reasonable good results with around 50 features for the first two data sets and 200 features for DrivFace. For some settings of the data 540 sets, SOS performs better than L1-SVM probably due to the reason that the $\ell_{2,1}$-norm regularization can select features in a more suitable way than ℓ_1-norm regularization for multi-class classification problem. Although the ℓ_1-norm regularization in L1-SVM can lead to sparse coefficients, it can not guarantee that the coefficients of a feature in all the C classes are all zero. Instead, the 545 $\ell_{2,1}$-norm regularization is more natural to achieve this.

In Table 3∼ 5, we further report the average classification testing error rate over 10-fold cross validation using 10 features for all the methods on ORL and COIL20 data. For DrivFace data, we report the corresponding results using 50 features. In other word, the number of selected features used for classification 550 in these tables is about 10% of total features for each data set. The last column of each table records the average classification performance over all settings of

	10 Classes	20 Classes	30 Classes	40 Classes	Average
L1-SVM	32.0(5.93)	26.0(2.77)	36.7(2.57)	39.3(2.91)	33.6(3.55)
SVA	23.0(4.72)	39.5(4.11)	50.9(3.49)	44.5(1.89)	39.5(3.55)
SOS	20.0(3.33)	25.0(2.56)	27.0 (2.60)	26.8(2.87)	24.7(2.84)
SOSA	**13.0(3.14)**	**12.8(2.47)**	**27.7(2.94)**	**22.0(1.43)**	**18.9(2.49)**

Table 3: Average classification testing error rate (%) and its standard error from 10-fold cross validation by using **10** features on ORL data.

	5 Classes	10 Classes	15 Classes	20 Classes	Average
L1-SVM	42.3(6.18)	34.3(4.11)	25.9(4.62)	23.8(4.63)	31.6(4.88)
SVA	55.4(3.64)	75.1(2.32)	81.6(1.59)	87.4(3.41)	74.9(2.74)
SOS	13.9(2.52)	31.4(2.91)	21.7(2.37)	26.1(3.55)	23.3(2.84)
SOSA	**7.4(1.36)**	**23.0 (2.02)**	**16.0(1.59)**	**20.6 (2.71)**	**16.8 (1.92)**

Table 4: Average classification testing error rate (%) and its standard error from 10-fold cross validation by using **10** features on COIL20 data.

	2 Classes	4 Classes	Average
L1-SVM	14.0(3.59)	20.7(4.44)	17.4(4.01)
SVA	16.4(3.34)	26.4(4.65)	21.4(3.99)
SOS	15.6(4.48)	19.2(5.47)	17.4(4.98)
SOSA	**12.0(3.18)**	**19.0(4.27)**	**15.5(3.73)**

Table 5: Average classification testing error rate (%) and its standard error from 10-fold cross validation by using **50** features on DrivFace data.

the data. In summary, we see that SOSA can reduce the classification error rate by 10.9% to 23.5% in average compared to SOS owing to its appropriate adjustment of unknown data heterogeneity. Although SVA is capable of adjusting 555 the heterogeneity, it performs pretty worse than SOSA on all the data sets due to the feature redundancy issue as a filter method. Comparing with L1-SVM, SOSA achieves 10.9% to 46.8% improvements in average owing to its adjustment of unknown data heterogeneity and the more suitable $\ell_{2,1}$-norm regularization.

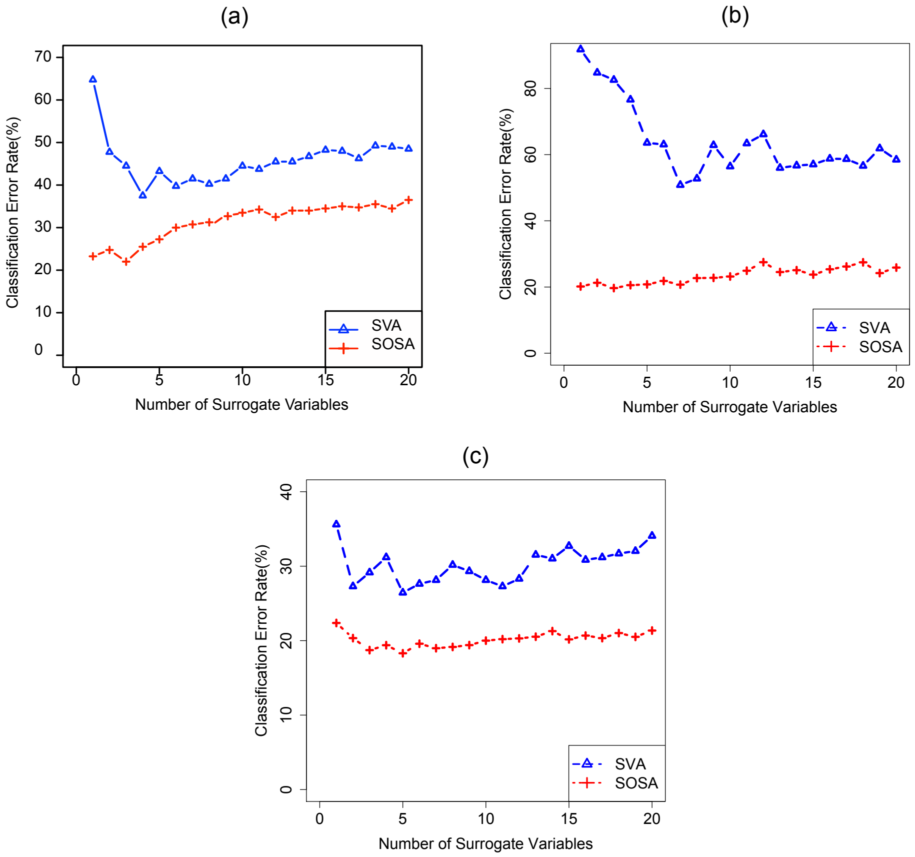

Figure 5: Plot of the classification performance versus the number of heterogeneous factors when using 10 features (50 features for DrivFace) selected from SVA and SOSA. (a) ORL; (b) COIL20; (c) DrivFace.

5.4. Parameter selection

To adjust the unknown data heterogeneity using our SOSA for feature selection, we need to set a parameter l– the number of heterogeneous factors to be estimated. We search in the range of 1 to 20 based on 10-fold cross validation. The best one leading to the lowest average classification testing error will be chosen. Fig. 5 shows the average classification testing error rate of SOSA from 10-fold cross validation versus the number of heterogeneous factors (l) using 10 selected features for ORL, COIL20 and 50 features for DrivFace. The classification is performed on their entire data sets respectively. As we observed, the classification testing error rate of SOSA for each data set decreases as the number of heterogeneous factors increases and then it increases when the number of heterogeneous factors exceeds a certain threshold. It is worth being aware that the classification performance of SOSA has such a relatively more flat variation trend with respect to l than that of SVA. Moreover, as an embedded feature selection method, SOSA always performed better than SVA on all the data sets when adjusting whichever the same number of heterogeneous factors. This further suggests that our sparse optimal scoring model and the effective algorithm can help select more discriminant features than the filter method. We also see that SOSA can adjust the unknown data heterogeneity to the maximum extent using 3 or 5 heterogeneous factors for ORL, COIL20 and DrivFace data because of the simple heterogeneity caused by such as lighting, facial expression, photographing angle and gaze direction. By observing that the best performance of SOSA on each data was achieved with a different number of heterogeneous factors, the choice of l is closely related with the specific data heterogeneity characteristics.

6. Conclusions

In this paper, we present a multi-class embedded feature selection method called as sparse optimal scoring with adjustment (SOSA), which is capable of addressing the data heterogeneity issue. We propose to perform feature selection on the adjusted data obtained by estimating and removing the unknown data heterogeneity from original data. Our feature selection is formulated as a sparse optimal scoring problem by imposing $\ell_{2,1}$-norm regularization on the coefficient matrix which hence can be solved effectively by proximal gradient algorithm. This allows our method can well handle the multi-class feature selection and classification simultaneously for heterogenous data. The experimental results on both synthetic data and three benchmark data sets have demonstrated that the features selected by our SOSA can consistently lead to better or comparative classification performance compared to those features selected by either traditional embedded methods or the filter method accounting for data heterogeneity. Moreover, the superiority of SOSA is especially more obvious when selecting less features. In the future work, we will consider to employ SOSA to high-dimensional modern data involving data heterogeneity and millions of features such as biomedical data.

Acknowledgement

This work is supported by National Natural Science Foundation of China (Grant No. 71702125).

References

Bach, F., Jenatton, R., Mairal, J., & Obozinski, G. (2011). *Convex optimization with sparsity-inducing norms*. MIT Press.

Boedigheimer, M. et al. (2008). Sources of variation in baseline gene expression levels from toxicogenomics study control animals across multiple laboratories. *BMC Genomics*, *9*, 285.

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

Combettes, P. L., & Wajs, V. R. (2006). Signal recovery by proximal forward-backward splitting. *SIAM Multiscale Modeling and Simulation*, *4*, 1168–1200.

Du, L., & Shen, Y. (2015). Unsupervised feature selection with adaptive structure learning. In *Proceedings of the Twentyoneth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 209–218).

Fan, J., Han, F., & Liu, H. (2014). Challenges of big data analysis. *National Science Review*, *1*, 293–314.

Fan, R., Chang, K., Hsieh, C., Wang, X., & Lin, C. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Fare, T. et al. (2003). Effects of atmospheric ozone on microarray data quality. *Analytical Chemistry*, *75*, 4672–4675.

Han, Y., Yang, Y., Yan, Y., Ma, Z., Sebe, N., & Zhou, X. (2015). Semi-supervised feature selection via spline regression for video semantic recognition. *IEEE Tran. on Neural Networks and Learning Systems*, *26*, 252–264.

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

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

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H. (2005). Face recognition using laplacian faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 328–340.

KaterineDiaz-Chito, AuraHernandez-Sabate, & M.Lopez, A. (2016). A reduced feature set for driver head pose estimation. *Applied Soft Computing*, *45*, 98–107.

Krishnapuram, B., Hartemink, A. J., Carin, L., & Figueiredo, M. A. (2004). A bayesian approach to joint feature selection and classifier design. *Statistica Sinica*, *26*, 1105–1111.

Leek, J. T., & Store, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, *3*, e161.

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

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2016). Feature selection: A data perspective. *arXiv preprint*, *arXiv:1601.07996*.

Mardia, K., Kent, J., & Bibby, J. (1979). *Multivariate Analysis*. New York: Academic Press.

Mohsenzadeh, Y., HamidSheikhzadeh, & SobhanNazarib (2016). Incremental relevance sample-feature machine: A fast marginal likelihood maximization approach for joint feature selection and classification. *Pattern Recognition*, *60*, 835–848.

Mohsenzadeh, Y., Sheikhzadeh, H., Reza, A. M., Bathaee, N., & Kalayeh, M. M. (2013). The relevance sample-feature machine: a sparse bayesian learning approach to joint feature-sample selection. *IEEE Transactions on Cybernetics*, *43*, 2241–2254.

Moreau, J. J. (1962). Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes Rendus de l'Académie des Sciences (Paris), Série A*, *255*, 2897–2899.

Nene, S. A., Nayar, S. K., & Murase, H. (1996). Columbia object image library (coil-20). In *Technical Report CUCS-005-96*.

Samaria, F., & Harter, A. (1994). Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*.

Simon, N., & Tibshirani, R. (2012). Standardization and the group lasso penalty. *Statistica Sinica*, *22*, 983–1001.

Tao, H., Hou, C., Nie, F., Jiao, Y., & Yi, D. (2016). Effective discriminative feature selection with nontrivial solution. *IEEE Transactions on Neural Networks and Learning Systems*, *27*, 796–808.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, *58*, 267–288.

Wang, L., Chen, S., & Wang, Y. (2014). A unified algorithm for mixed l2,p-minimizations and its application in feature selection. *Computational Optimization and Applications*, *58*, 409–421.

Wang, L., Zhu, J., & Zou, H. (2006). The doubly regularized support vector machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*, 589–616.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for svms. *Proceedings of the 13th International Conference on Neural Information Processing Systems*, (pp. 647–653).

Wu, Y., Wipf, D. P., & Yun, J. (2015). Understanding and evaluating sparse linear discriminant analysis. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* (pp. 1070–1078).

Xiang, S., Nie, F., Meng, G., Pan, C., & Zhang, C. (2012). Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 1738–1754.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Stat. Society, Series B*, *68*, 49–67.

Zhu, J., Rosset, S., Hastie, T., & Tibshirani, R. (2004). 1-norm support vector machines. In *Neural Information Processing Systems* (pp. 49–56).

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

Author contribution

Meng Lu is in charge of idea development, experiment design and implementation, result analysis and manuscript drafting.