Omega 59 (2016) 40–46

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Constrained subspace classifier for high dimensional datasets

Orestis P. Panagopoulos [a], Vijay Pappu [b], Petros Xanthopoulos [a,*], Panos M. Pardalos [b]

[a] Department of Industrial Engineering and Management Systems, University of Central Florida, 4000 Central Florida blvd., Orlando 32816, FL, United States
[b] Department of Industrial Engineering, University of Florida, 401 Weil Hall, Gainesville 32608, FL, United States

ARTICLE INFO

Article history:
Received 31 August 2014
Accepted 11 May 2015
Available online 10 June 2015

Keywords:
Constrained subspace classifier
High dimensional datasets
Principal angles
Local subspace classifier

ABSTRACT

Datasets with significantly larger number of features, compared to samples, pose a serious challenge in supervised learning. Such datasets arise in various areas including business analytics. In this paper, a new binary classification method called *constrained subspace classifier (CSC)* is proposed for such high dimensional datasets. CSC improves on an earlier proposed classification method called *local subspace classifier (LSC)* by accounting for the relative angle between subspaces while approximating the classes with individual subspaces. CSC is formulated as an optimization problem and can be solved by an efficient alternating optimization technique. Classification performance is tested in publicly available datasets. The improvement in classification accuracy over LSC shows the importance of considering the relative angle between the subspaces while approximating the classes. Additionally, CSC appears to be a robust classifier, compared to traditional two step methods that perform feature selection and classification in two distinct steps.

1. Introduction

High dimensional datasets are currently prevalent in many business applications. The methodical collection of every facet of the data has lead to a significant increase in its dimensionality. Examples include but are not limited to financial services [43], e-commerce [12] and marketing [32]. Other examples of datasets with a high number of features are shown in Table 1.

Classification tasks on high dimensional datasets pose significant challenges to the standard statistical methods and render many existing classification techniques impractical [22]. The generalization ability of many classification algorithms is compromised due to *curse of dimensionality* arising from high number of features of the input space [26]. Earlier studies have revealed the geometrical distortion that arises in high dimensional data spaces, where the ratio of distances between the farthest and nearest neighbors to a given target is almost equal to 1 for a wide variety of data distributions and distance functions [4]. Moreover, several statistical methods require knowing class covariances *a priori*. In the case that class covariances are unavailable, such estimates from sample data would be unreliable due to small sample sizes. One common approach to address the aforementioned challenges involves reducing the dimensionality of the dataset either by using feature extraction [29] and/or feature selection prior to classification [34,8].

Feature selection is usually performed in different ways through filter, wrapper, and embedded methods. Filter methods access features during a separate process prior to classification. Variables are given a score according to a filtering function and are ordered accordingly. Features with the lowest scores are discarded while the rest are used from the classifier. Hypothesis testing and statistic tests such as *t*-test have also been used as filtering procedures [17]. Wrapper methods on the other hand use the classifier structure itself to evaluate the importance of features based on the idea that the classifier can provide a better estimate of accuracy than a separate independent process [6]. The main drawback of wrapper methods is that increased computational power is often required since the classification process has to be repeated for each feature set considered. Metaheuristics used for feature selection can also be classified as wrapper methods [40,30,47]. Embedded methods perform feature selection in a way so that the classification algorithm is executed while variables are evaluated and selected. Examples include the weighting of features in support vector machines [18], where the authors developed the SVM method of recursive feature elimination for feature selection, and the use of random forests for feature evaluation [21]. In the later, feature elimination occurs for the attributes with the lowest raw importance score.

Feature extraction techniques transform the input data into a set of *meta*-features that extract the relevant information from the input data for classification. One popular technique called *principal component analysis (PCA)* finds a set of linearly uncorrelated variables called *principal components* from a set of observations of possibly correlated variables [23,36]. PCA removes redundancy by transforming the data from a higher dimensional space into an orthogonal lower dimensional space. This transformation is

* Corresponding author.
E-mail addresses: orepana@gmail.com (O.P. Panagopoulos), psnvijay.iitm@gmail.com (V. Pappu), petrosx@ucf.edu (P. Xanthopoulos), pardalos@ufl.edu (P.M. Pardalos).

http://dx.doi.org/10.1016/j.omega.2015.05.009
0305-0483/

Table 1
Examples of high dimensional datasets.

Dataset	Reference
Customer relationship management data	[39]
Covariation information of stocks	[7]
Text datasets for classification	[20]
Data collected from surveys	[2]
Netflix dataset	[3]
MRI data	[24]
Mass spectroscopy data	[14]

performed in a way that the first principal component captures as much variation in the data as possible, and each succeeding component accounts for a decreasing amount of variance [42]. The number of retained principal components is usually less than or equal to the number of original variables and are determined using several criteria like the eigenvalue-one criterion, scree test and proportion of variance accounted for.

The aforementioned dimensionality reduction techniques decrease the complexity of the classification model and attempt to improve the classification performance [34]. The choice of the dimensionality reduction technique depends on the nature (e.g. level of correlation, presence of outliers) of the data that is used for classification.

Local subspace classifier (LSC) utilizes PCA to perform classification. During the training phase, a lower dimensional subspace is found for each class that approximates the data [27]. In the testing phase, a new data point is classified by calculating the distance of the point to each subspace and choosing the class with minimal distance. Although LSC is simple and relatively easy to implement, it has its own limitations. LSC finds the subspaces for each class *separately* without the *knowledge* of the presence of the other class. While each subspace approximates the data well, however these projections may not be *ideal* from a classification perspective. In this paper, we construct another classifier called *constrained subspace classifier* (CSC) which expands LSC by including the relative orientation of the subspaces of two classes in an integrated optimization model. LSC formulation is modified to include the relative angle between the subspaces and is solved efficiently using alternating optimization techniques. The performance of CSC on publicly available datasets is evaluated and compared with LSC and other classifiers.

The remainder of the paper is organized as follows. Section 2 gives an introduction to LSC and Section 3 introduces the CSC. In Section 4 we demonstrate a first comparison on a toy dataset whereas in Section 5 we present the computational experiment on six real datasets along with their discussion as well as we provide the comparative computational results for CSC against support vector machine (SVM), PCA/SVM and Naive Bayes classifier. Lastly, in Section 6 we discus potential future extensions of this algorithm.

2. Local subspace classifier

Consider a binary classification problem. Let the matrices $\mathcal{X}_1 \in \mathbb{R}^{p\times m}$ and $\mathcal{X}_2 \in \mathbb{R}^{p\times l}$ be given, whose columns represent the training examples of two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. The number of samples in $\mathcal{C}_1$ and $\mathcal{C}_2$ are given by m and n respectively. The number of features is given by p. Local subspace classifier attempts to find two subspaces separately, one for each class that *best* approximates the data. Let $\boldsymbol{U}_1 = [\boldsymbol{u}_1^{(1)}, \boldsymbol{u}_2^{(1)}, \ldots, \boldsymbol{u}_k^{(1)}]_{p\times k}$ and $\boldsymbol{U}_2 = [\boldsymbol{u}_1^{(2)}, \boldsymbol{u}_2^{(2)}, \ldots, \boldsymbol{u}_k^{(2)}]_{p\times k}$ represent orthonormal bases of two k-dimensional linear subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ that approximate classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. We assume the dimensionality of subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ to be same and equal to k without loss of generality. $\mathcal{S}_1$ and $\mathcal{S}_2$ attempt to capture *maximal* variance in classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively by optimizing the following optimization problems:

$$\begin{aligned} \underset{\boldsymbol{U}_1 \in \mathbb{R}^{p\times k}}{\text{maximize}} \quad & \operatorname{tr}(\boldsymbol{U}_1^T \mathcal{X}_1 \mathcal{X}_1^T \boldsymbol{U}_1) \\ \text{subject to} \quad & \boldsymbol{U}_1^T \boldsymbol{U}_1 = \boldsymbol{I}_k \end{aligned} \tag{1}$$

where $\boldsymbol{I}_k$ is the identity matrix of size k.

The solution to the optimization problem (1) is given by eigenvectors corresponding to the k largest eigenvalues of matrix $\mathcal{X}_1\mathcal{X}_1^T$ [15]. Similarly, the following optimization problem is solved to obtain the orthonormal basis $\boldsymbol{U}_2$ representing $\mathcal{S}_2$:

$$\begin{aligned} \underset{\boldsymbol{U}_2 \in \mathbb{R}^{p\times k}}{\text{maximize}} \quad & \operatorname{tr}(\boldsymbol{U}_2^T \mathcal{X}_2 \mathcal{X}_2^T \boldsymbol{U}_2) \\ \text{subject to} \quad & \boldsymbol{U}_2^T \boldsymbol{U}_2 = \boldsymbol{I}_k \end{aligned} \tag{2}$$

The orthonormal basis $\boldsymbol{U}_2$ is obtained by choosing eigenvectors corresponding to the k largest eigenvalues of matrix $\mathcal{X}_2\mathcal{X}_2^T$. A new point $\boldsymbol{x}$ is classified by computing its distance from subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$:

$$\operatorname{dist}(\boldsymbol{x}, \mathcal{S}_i) = \operatorname{tr}(\boldsymbol{U}_i^T \boldsymbol{x}\boldsymbol{x}^T \boldsymbol{U}_i) \tag{3}$$

and the class of $\boldsymbol{x}$ is determined as

$$\operatorname{class}(\boldsymbol{x}) = \arg \min_{i \in \{1,2\}} \{\operatorname{dist}(\boldsymbol{x}, \mathcal{S}_i)\} \tag{4}$$

Though the subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ approximate the classes well, these projections may not be *ideal* for classification tasks as each of them are obtained *without* the knowledge of another class/subspace. Hence, from a classification performance perspective, these subspaces may not be the *best* projections for the classes. In order to account for the presence of another subspace, we consider the relative orientation of the subspaces.

3. Constrained subspace classifier

Constrained subspace classifier finds two subspaces *simultaneously*, one for each class, such that each subspace accounts for maximal variance in the data in the *presence* of the other class/subspace. Thus, CSC allows for a *tradeoff* between approximating the classes well and the relative orientation among the subspaces. The relative orientation between subspaces is generally defined as principal angles [19]. We briefly review principal angles between subspaces below, which are further utilized to modify the formulation of LSC to include the relative orientation among the subspaces.

Definition 1. Let $\boldsymbol{U}_1 \in \mathbb{R}^{p\times k}$ and $\boldsymbol{U}_2 \in \mathbb{R}^{p\times k}$ be two orthonormal matrices spanning subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$. The principal angles $0 \le \theta_1 \le \theta_2 \le \theta_3 \le \cdots \le \theta_k \le \pi/2$ between subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$, are defined recursively by

$$\begin{aligned} \cos\theta_i = \max_{\boldsymbol{x}_m \in \mathcal{S}_1} \max_{\boldsymbol{y}_n \in \mathcal{S}_2} \quad & \boldsymbol{x}_m^T \boldsymbol{y}_n \\ \text{subject to} \quad & \boldsymbol{x}_m^\top \boldsymbol{x}_n = 1, \quad \boldsymbol{y}_m^\top \boldsymbol{y}_n = 1 \quad \text{for } m = n \\ & \boldsymbol{x}_m^\top \boldsymbol{x}_n = 0, \quad \boldsymbol{y}_m^\top \boldsymbol{y}_n = 0 \quad \text{for } m \ne n \\ & \forall m, n = 1, 2, \ldots, k. \end{aligned} \tag{5}$$

where $\boldsymbol{x}_m$ and $\boldsymbol{y}_n$ are the column vectors of $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ respectively. Intuitively, the first principal angle θ_1 is the smallest angle between all pairs of unit vectors in the first and second subspaces. The rest of the principal angles are similarly defined.

Theorem 1. *Let $U_1 \in \mathbb{R}^{p\times k}$ and $U_2 \in \mathbb{R}^{p\times k}$ be rectangular matrices whose column vectors span the subspaces $S_1 \in \mathbb{R}^k$ and $S_2 \in \mathbb{R}^k$ respectively. Let $M = U_1^\top U_2 \in \mathbb{R}^{k\times k}$, using singular value decomposition we can express M by*

$$M = YCZ^\top \tag{6}$$

where $Y^\top Y = I_k$, $Z^\top Z = I_k$ and $C = diag(\sigma_1, \sigma_2, \ldots, \sigma_k)$.

If we assume that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$ *then the principal angles are given by* $\cos\theta_k = \sigma_k(M)\ \forall i = 1, 2, \ldots, k$.

Proof. See Bjorck and Golub [5].□

The cosines of the principal angles are also sometimes known as *canonical correlations*.

3.1. Seeking a distance metric

We consider the metric that defines the relative orientation between S_1 and S_2 spanned by U_1 and U_2 respectively to be the projection F-norm [13] defined by

$$d_{pF}(\boldsymbol{U}_1, \boldsymbol{U}_2) = \frac{1}{\sqrt{2}} \| \boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top \|_F \tag{7}$$

The projection F-norm is obtained by embedding the Grassmann manifold in the set of n-by-n projection matrices of rank p. The choice of the metric preserves convexity. It can be represented in terms of the sines of principal angles as follows.

The right hand side norm can be expressed as

$$\| \boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top \|_F^2 = \mathrm{tr}((\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top)^\top (\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top)) \tag{8}$$

$$\| \boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top \|_F^2 = \| \boldsymbol{U}_1 \|_F^2 + \| \boldsymbol{U}_2 \|_F^2 - 2\| \boldsymbol{U}_2^\top \boldsymbol{U}_1 \|_F^2 \tag{9}$$

Using Theorem 1, (9) becomes

$$= \sum_{i=1}^{k} \lambda_i + \sum_{i=1}^{k} \lambda_i - 2\sum_{i=1}^{k} \cos^2\theta_i \tag{10}$$

$$= k + k - 2\sum_{i=1}^{k} \cos^2\theta_i \tag{11}$$

$$= 2\sum_{i=1}^{k} \sin^2\theta_i \tag{12}$$

where λ_i are the eigenvalues of $U_j\ \forall i = 1, 2, \ldots, k$ and $j = \{1, 2\}$.

Hence the projection F-norm becomes

$$d_{pF}(\boldsymbol{U}_1, \boldsymbol{U}_2) = \frac{1}{\sqrt{2}} \| \boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top \|_F = \sqrt{\sum_{i=1}^{k} \sin^2\theta_i}. \tag{13}$$

3.2. Formulating CSC

The projection metric is utilized to incorporate the relative orientation between subspaces in LSC. The formulation of LSC is modified as shown below to obtain the *constrained subspace classifier (CSC)*:

$$\underset{\boldsymbol{U}_1, \boldsymbol{U}_2 \in \mathbb{R}^{p\times k}}{\text{maximize}} \quad \mathrm{tr}(\boldsymbol{U}_1^T \mathcal{X}_1 \mathcal{X}_1^T \boldsymbol{U}_1) + \mathrm{tr}(\boldsymbol{U}_2^T \mathcal{X}_2 \mathcal{X}_2^T \boldsymbol{U}_2) - C \| \boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top \|_F^2$$

$$\text{subject to} \quad \boldsymbol{U}_1^T\boldsymbol{U}_1 = \boldsymbol{I}_k \tag{14a}$$

$$\boldsymbol{U}_2^T\boldsymbol{U}_2 = \boldsymbol{I}_k \tag{14b}$$

where the parameter C controls the tradeoff between the relative orientation of the subspaces and the approximation of the data.

From calculations in Section 3.1:

$$\| \boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top \|_F^2 = 2k - 2\,\mathrm{tr}(\boldsymbol{U}_1^\top \boldsymbol{U}_2\boldsymbol{U}_2^\top \boldsymbol{U}_1) \tag{15}$$

Hence the optimization problem becomes

$$\underset{\boldsymbol{U}_1, \boldsymbol{U}_2 \in \mathbb{R}^{p\times k}}{\text{maximize}} \quad \mathrm{tr}(\boldsymbol{U}_1^T \mathcal{X}_1 \mathcal{X}_1^T \boldsymbol{U}_1) + \mathrm{tr}(\boldsymbol{U}_2^T \mathcal{X}_2 \mathcal{X}_2^T \boldsymbol{U}_2) + C\,\mathrm{tr}(\boldsymbol{U}_1^T \boldsymbol{U}_2\boldsymbol{U}_2^T \boldsymbol{U}_1)$$

$$\text{subject to} \quad \boldsymbol{U}_1^T\boldsymbol{U}_1 = \boldsymbol{I}_k \tag{16a}$$

$$\boldsymbol{U}_2^T\boldsymbol{U}_2 = \boldsymbol{I}_k \tag{16b}$$

It is important to note here that when $C = 0$, CSC reduces to LSC. Additionally, for larger positive values of C, the relative orientation between subspaces reduces, while for larger negative values of C, the relative orientation increases.

3.3. Algorithm

Here we introduce an alternating optimization algorithm to solve (16a). For a fixed $\boldsymbol{U}_2$, (16a) reduces to:

$$\underset{\boldsymbol{U}_1 \in \mathbb{R}^{p\times k}}{\text{maximize}} \quad \mathrm{tr}(\boldsymbol{U}_1^T(\mathcal{X}_1\mathcal{X}_1^T + C\boldsymbol{U}_2\boldsymbol{U}_2^T)\boldsymbol{U}_1)$$

$$\text{subject to} \quad \boldsymbol{U}_1^T\boldsymbol{U}_1 = \boldsymbol{I}_k \tag{17}$$

The solution to (17) is obtained by choosing the eigenvectors corresponding to the k largest eigenvalues of symmetric matrix $\mathcal{X}_1\mathcal{X}_1^T + C\boldsymbol{U}_2\boldsymbol{U}_2^T$.

Similarly, for a fixed $\boldsymbol{U}_1$, (16) reduces to

$$\underset{\boldsymbol{U}_2 \in \mathbb{R}^{p\times k}}{\text{maximize}} \quad \mathrm{tr}(\boldsymbol{U}_2^T(\mathcal{X}_2\mathcal{X}_2^T + C\boldsymbol{U}_1\boldsymbol{U}_1^T)\boldsymbol{U}_2)$$

$$\text{subject to} \quad \boldsymbol{U}_2^T\boldsymbol{U}_2 = \boldsymbol{I}_k \tag{18}$$

where the solution to (18) is again obtained by choosing the eigenvectors corresponding to the k largest eigenvalues of symmetric matrix $\mathcal{X}_2\mathcal{X}_2^T + C\boldsymbol{U}_1\boldsymbol{U}_1^T$. We define the following three termination rules:

- Maximum limit Z on the number of iterations,
- Relative change in $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ at iteration m and $m+1$,

$$\mathrm{tol}_{\boldsymbol{U}_1}^m = \frac{\| \boldsymbol{U}_1^{(m+1)} - \boldsymbol{U}_1^{(m)} \|_F}{\sqrt{q}}, \quad \mathrm{tol}_{\boldsymbol{U}_2}^m = \frac{\| \boldsymbol{U}_2^{(m+1)} - \boldsymbol{U}_2^{(m)} \|_F}{\sqrt{q}} \tag{19}$$

  where $q = pk$.
- Relative change in objective function value of (16) at iteration m and $m+1$,

$$\mathrm{tol}_f^m = \frac{F^{(m+1)} - F^{(m)}}{|F^{(m)}| + 1} \tag{20}$$

For proof of convergence see Theorem 2.

The algorithm for CSC can be summarized as follows:

Algorithm 1. CSC ($\mathcal{X}_1, \mathcal{X}_2, k, C$).

1. Initialize $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ such that $\boldsymbol{U}_1^T\boldsymbol{U}_1 = \boldsymbol{I}_k$, $\boldsymbol{U}_2^T\boldsymbol{U}_2 = \boldsymbol{I}_k$.
2. Find eigenvectors corresponding to the k largest eigenvalues of symmetric matrix $\mathcal{X}_1\mathcal{X}_1^T + C\boldsymbol{U}_2\boldsymbol{U}_2^T$.
3. Find eigenvectors corresponding to the k largest eigenvalues of symmetric matrix $\mathcal{X}_2\mathcal{X}_2^T + C\boldsymbol{U}_1\boldsymbol{U}_1^T$.
4. Alternate between 2 and 3 until one of the termination rules is satisfied.

Theorem 2. *Algorithm 1 converges.*

Proof. Let $\mathcal{S}_l$ be a subspace of the space $\mathcal{S}_L$, where L is the dimensionality of the original data points and l is the reduced dimensionality of those points when projected onto the subspace $\mathcal{S}_l$. There is a choice of γ many such subspaces where

$$\gamma = \frac{L!}{l!(L-l)!} \tag{21}$$

with each subspace choice having a basis whose elements are the vector columns of $\boldsymbol{U}^i = [\boldsymbol{u}_1\ \boldsymbol{u}_2\ \ldots\ \boldsymbol{u}_l] \in \mathbb{R}^{L\times l}$ where $u_k \in \mathbb{R}^L$ with $k = 1, 2, 3, \ldots, l$ and $i = 1, 2, 3, \ldots, \gamma$.

Each choice of $\boldsymbol{U}^i$ corresponds to a covariant matrix of the projected data points that has a trace given by $T^i \equiv \mathrm{tr}(\boldsymbol{U}^{iT} XX^T \boldsymbol{U}^i)$. Since there is a finite number of subspaces, we also have a finite

number of bases $\boldsymbol{U}^i$ and therefore a finite number of values for the T^i.

Define the set of all values of $\{T^1, T^2, T^3, \ldots, T^\gamma\}$ and also define the corresponding set of $\{\boldsymbol{U}^1, \boldsymbol{U}^2, \boldsymbol{U}^3, \ldots, \boldsymbol{U}^\gamma\}$. Similarly for the second class of data points and the corresponding set of subspaces define the set of values of the traces $\{S^1, S^2, S^3, \ldots, S^\gamma\}$ and also define the corresponding set of subspace basis $\{\boldsymbol{V}^1, \boldsymbol{V}^2, \boldsymbol{V}^3, \ldots, \boldsymbol{V}^\gamma\}$ where $S^j \equiv \mathrm{tr}(\boldsymbol{V}^{jT} YY^T \boldsymbol{V}^j)$.

Let n and $n+1$ be two consecutive iterations. Then the objective function at each iteration r is given by

$$F_r = T^{i_r} + S^{j_r} + M^{i_r j_r} \quad \text{where}$$
$$T^{i_r} \equiv \mathrm{tr}(\boldsymbol{U}^{i_r^T} XX^T \boldsymbol{U}^{i_r})$$
$$S^{j_r} \equiv \mathrm{tr}(\boldsymbol{V}^{j_r^T} YY^T \boldsymbol{V}^{j_r})$$
$$M^{i_r j_r} \equiv \mathrm{tr}(\boldsymbol{U}^{i_r^T} \boldsymbol{V}^{j_r} \boldsymbol{V}^{j_r^T} \boldsymbol{U}^{i_r}) \tag{22}$$

For $r=n$ we fix $j_r=j_n$ (that is we fix S_n^j) and find i_n that maximizes

$$F_n = T^{i_n} + S^{j_n} + M^{i_n j_n} \tag{23}$$

Effectively, we solve

$$\arg \max_{i_n \in \{1,\ldots,\gamma\}} \{T^{i_n} + M^{i_n j_n} \mid j_n = constant\} \tag{24}$$

For $r = n+1$ we fix $i_r = i_{n+1} = i_n$ (that is we fix $T^{i_{n+1}} = T^{i_n}$) and find j_{n+1} that maximizes

$$F_{n+1} = T^{i_{n+1}} + S^{j_{n+1}} + M^{i_{n+1} j_{n+1}}$$
$$= T^{i_n} + S^{j_{n+1}} + M^{i_n j_{n+1}} \tag{25}$$

Effectively, we solve

$$\arg \max_{j_{n+1} \in \{1,\ldots,\gamma\}} \{S^{j_{n+1}} + M^{i_n j_{n+1}} \mid i_n = constant\} \tag{26}$$

Therefore,

$$F_{n+1} - F_n = (T^{i_n} + S^{j_{n+1}} + M^{i_n j_{n+1}}) - (T^{i_n} + S^{j_n} + M^{i_n j_n})$$
$$= (S^{j_{n+1}} - S^{j_n}) + (M^{i_n j_{n+1}} - M^{i_n j_n}) \tag{27}$$

Since $j_n \in \{1, \ldots, \gamma\}$ then $S^{j_{n+1}} + M^{i_n j_{n+1}}$ and $S^{j_n} + M^{i_n j_n}$ are terms of the same sequence.

Therefore from (26) we have that

$$S^{j_{n+1}} + M^{i_n j_{n+1}} > S^{j_n} + M^{i_n j_n} \tag{28}$$

From (27) and (28) we get that $F_{n+1} - F_n > 0$. Hence $F_{n+1} > F_n$.

Therefore, the sequence $\{F_n\}$ is increasing.

Since $\{\boldsymbol{U}^i\}$ and $\{\boldsymbol{V}^i\}$ are finite sets then $\{T^i\}$, $\{S^j\}$ and $\{M^{ij}\}$ are also finite sets. Therefore, $\{T^i\}$, $\{S^j\}$ and $\{M^{ij}\}$ have maxima. Since each element of $\{F_n\}$ is a linear combination of elements from $\{T^i\}$, $\{S^j\}$ and $\{M^{ij}\}$ then $\{F_n\}$ also has a maximum. That means $\{F_n\}$ is bounded from above.

We have proven that $\{F_n\}$ is an increasing sequence of real numbers and also bounded from above. Therefore, it converges.□

Table 2
Average classification accuracies and relative angle between subspaces generated from LSC and CSC in two examples.

Datasets	$\mathcal{N}_1$		$\mathcal{N}_2$		LSC		CSC	
	$\boldsymbol{\mu}_1$	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_2$	$\boldsymbol{\Sigma}_2$	Acc (%)	Angle (θ)	Acc (%)	Angle (θ)
Example 1	$\begin{bmatrix}9\\10\end{bmatrix}$	$\begin{bmatrix}4 & 1.1\\1.1 & 4\end{bmatrix}$	$\begin{bmatrix}2\\5\end{bmatrix}$	$\begin{bmatrix}4 & 0\\0 & 3\end{bmatrix}$	74	0.54	**92**	0.99
Example 2	$\begin{bmatrix}3\\5\end{bmatrix}$	$\begin{bmatrix}4 & -2\\-2 & 6\end{bmatrix}$	$\begin{bmatrix}10\\10\end{bmatrix}$	$\begin{bmatrix}5 & 2\\2 & 5\end{bmatrix}$	87	0.92	**97**	0.16

4. Illustrating examples

We consider two examples here showing the effect of changing the relative angle between subspaces generated by LSC. The datasets are generated from two bivariate normal distributions $\mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ representing classes $\mathcal{C}_1$ and $\mathcal{C}_2$. Each class consists of 100 randomly generated points from $\mathcal{N}_1$ and $\mathcal{N}_2$ respectively. The parameters of $\mathcal{N}_1$ and $\mathcal{N}_2$ for the two classes are shown in Table 2. The LSC and CSC are trained on the data with $k=1$. The values of Z, tol_f^m, $tol_{U_1}^m$ and $tol_{U_2}^m$ are chosen to be 2000, 1e−6, 1e−6 and 1e−6 respectively. The value of C is set to -10^3 for Example 1 and 10^3 for Example 2. The classification accuracies are obtained via leave-one-out cross validation (LOOCV) [25]. The subspaces obtained for each of the training folds in example 1 and example 2 are shown in Figs. 1 and 2 respectively. The average classification accuracies and the average relative angle θ $(0 \le \theta \le \pi/2)$ between the subspaces for LSC and CSC are reported in Table 2. In example 1, increasing the relative angle between the subspaces clearly improves the classification accuracy by ≈ 24%. However in Example 2, decreasing the relative angle between the subspaces shows better classification performance and outperforms LSC by ≈ 11%. These examples show that the relative orientation of the subspaces should also be considered in addition to capturing the *maximal* variance in data.

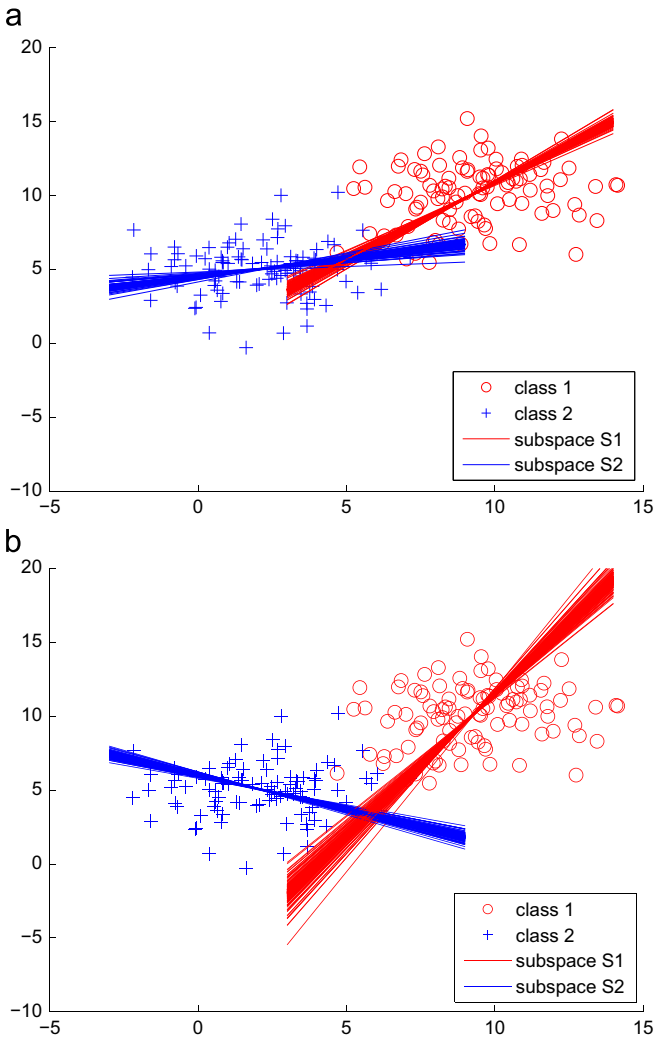

Fig. 1. Data points generated by $\mathcal{N}_1$ and $\mathcal{N}_2$ in example 1 and the subspaces generated by LSC and CSC in each of the training folds.

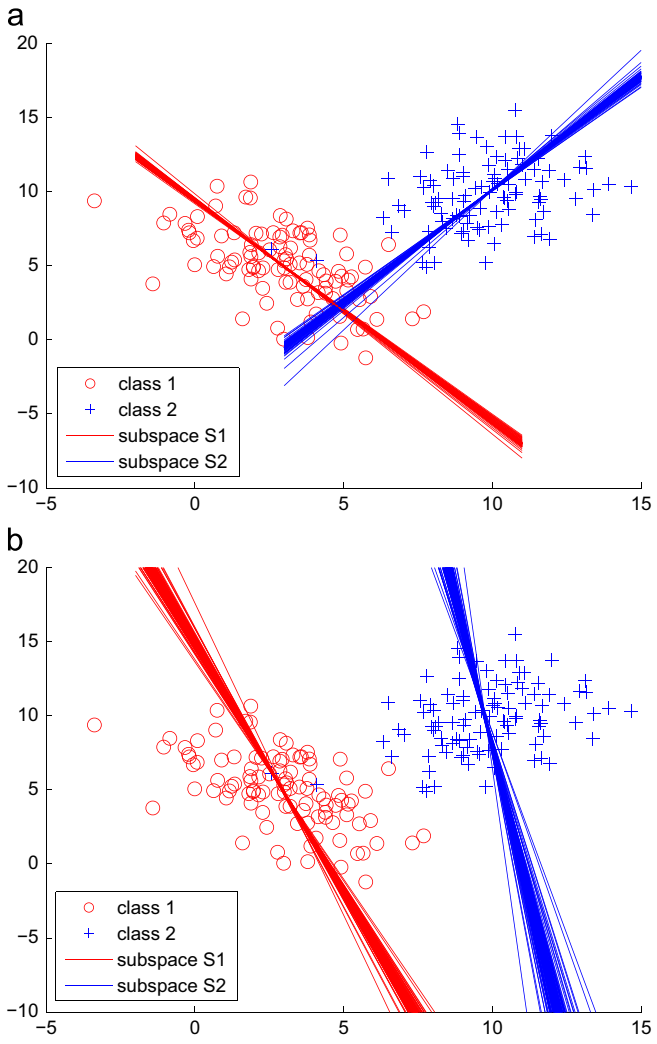

Fig. 2. Data points generated by $\mathcal{N}_1$ and $\mathcal{N}_2$ in example 2 and the subspaces generated by LSC and CSC in each of the training folds.

5. Numerical experiments

The performance of CSC is evaluated on six publicly available datasets and they are summarized in Table 3. Four of them (DLBCL, Breast, Colon, DBWorld) are high dimensional (#features≫#samples) and two of them (Mushroom, Spambase) have significantly more samples than features.

The performance of CSC is evaluated for different values of C, and compared to that of LSC. The values of Carechosen in such a way that the relative angle between the subspaces varies uniformly. The relative angle between the subspaces is evaluated in terms of the projection metric d_{pF}. The value of d_{pF} varies between 0 and k, where k is the dimensionality of the subspaces. The value of k is chosen as {1, 3, 10}. Experiments are performed with a 2.60 GHz Intel Core i5 CPU running OS X with 8.0 GB of main memory. The classification performance is evaluated using LOOCV technique.

The classification accuracies as a function of C for different values of k are shown in Fig. 3(a)–(f). C_0 represents the results of LSC since for $C=0$ the CSC reduces to LSC. C_{-1}, C_{-2} correspond to $C<0$ and C_{+1}, C_{+2} correspond to $C>0$. As mentioned earlier, positive values of C decrease the relative angle between the subspaces while negative values of C increase the relative angle. The values of Z, tol_f^m, $tol_{\boldsymbol{U}_1}^m$ and $tol_{\boldsymbol{U}_2}^m$ are chosen to be 2000, 1e−6, 1e−6 and 1e−6 respectively.

Table 3
Summary table of datasets used for experiments. The first four datasets, namely DLBCL, Breast, Colon and DBWorld are high dimensional since the number of features greatly outnumbers the number of samples. The last two (Mushroom and Spambase) are not high dimensional since the number of samples is greater than the number of features.

Dataset	Reference	# Samples	# Features
DLBCL	[37]	77	5469
Breast	[41]	77	4869
Colon	[1]	62	2000
DBWorld	[20]	64	4702
Mushroom	[35]	8124	126
Spambase	[28]	4601	57

For DLBCL and Colon datasets, classification accuracy is improved by reducing the relative angle between subspaces for $k=3$, $k=10$ and $k=1$, $k=3$ respectively. In the case of Breast dataset, increasing the relative angle for $k=1$ considerably improves the classification accuracy. For the DBWorld dataset the classification accuracy of CSC was almost identical to that of LSC.

With respect to the lower dimensional datasets, CSC performed at least as good as LSC. In the case of Spambase dataset, CSC was able to slightly increase the accuracy of classification for positive values of C. The penalty parameter C gives the flexibility to adjust.

5.1. Computational comparisons

We provide the comparative computational results for CSC against SVM, PCA/SVM and Naive Bayes classifier summarized in Table 4. PCA was used to reduce the dimensionality of the datasets prior to SVM classification. Through PCA, components that correspond to 80% of the total variance were used for classification. The contributed variance of the factors maintained exceed the 70% threshold [38] due to the relative small amount of samples compared to the number of features of the data. SVM was trained using a Radius Basis Function (RBF) kernel. The pair of parameter settings (k, C) used in CSC in each dataset was: DLBCL (3, 2E+10), Breast (1, −5E+03), Colon (3, 5E+09), DBWorld (1, 2E+03), Mushroom(10, −1E+03) and Spambase (10, 5E+03). Naive Bayes classifier shows the lowest overall accuracy. CSC demonstrates competitive behavior with respect to dataset dimensionality. The performance of SVM degrades in high dimensional datasets and the combined use of PCA/SVM does not perform well as the number of features decreases. However, CSC remains robust although it does not necessarily achieve the highest accuracy in every experiment.

6. Conclusion and future research

In this paper, a new classification algorithm, called constrained subspace classifier, was proposed and designed for high dimensional datasets. We have shown that the proposed algorithm outperforms LSC. In addition to approximating the classes well by individual subspaces, CSC also accounts for the relative angle between the subspaces by utilizing the projection metric. An efficient alternating optimization technique is also proposed. CSC has been evaluated on publicly available datasets and is compared to LSC. The improvement in classification accuracy shows the importance of considering the relative angle between subspaces while approximating the classes. Additionally, CSC seems to be effective when introduced for lower dimensional subspaces. The robust nature of CSC reveals that it can serve as a one step method for for preprocessing-free classification. To this end, CSC presents

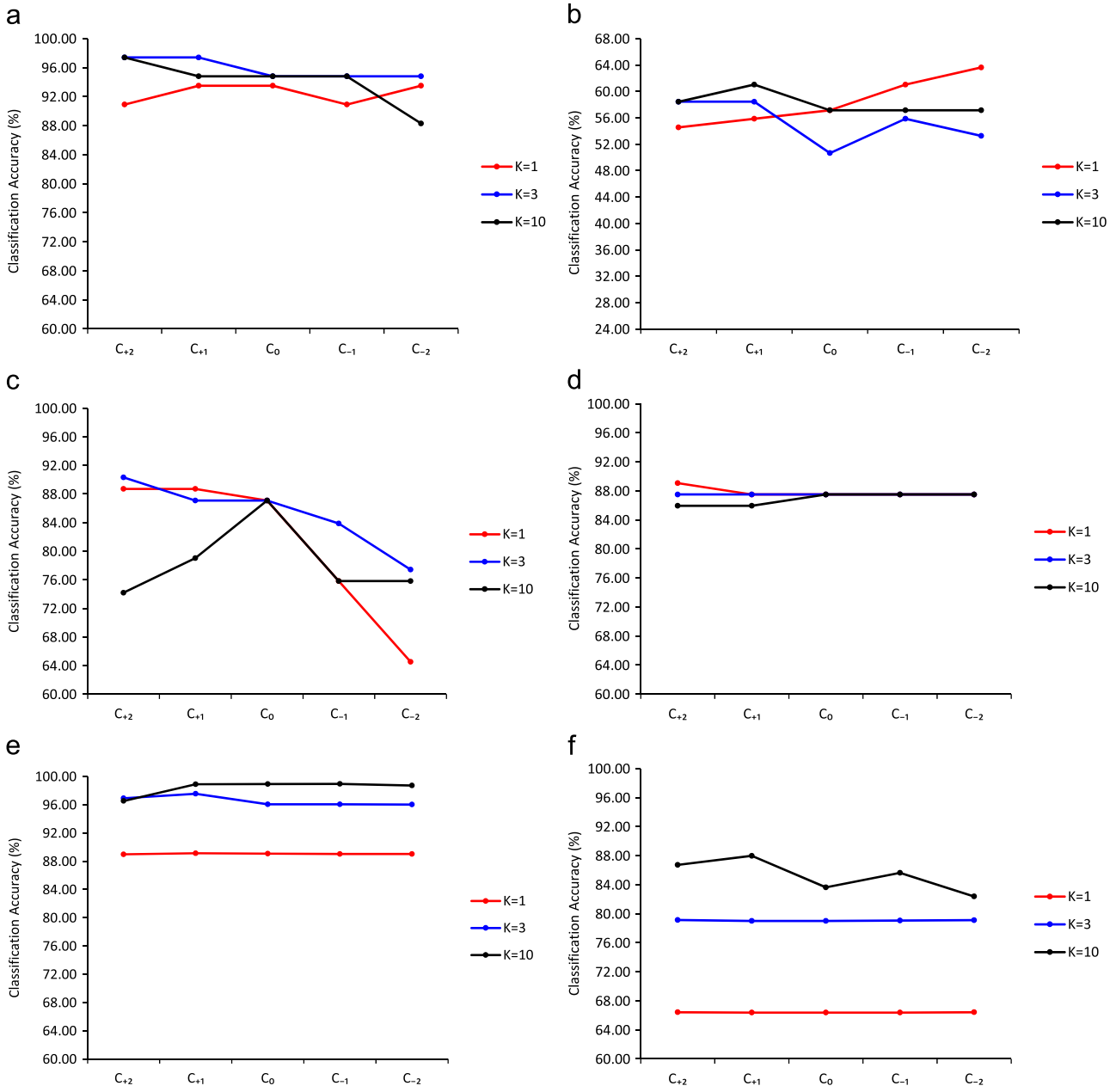

Fig. 3. Classification accuracy of CSC for (a) DLBCL, (b) Breast, (c) Colon, (d) DBWorld, (e) Mushroom and (f) Spambase datasets. C_0 represents the results of LSC. C_{-1}, C_{-2} correspond to $C<0$ and C_{+1}, C_{+2} correspond to $C>0$ used for CSC.

Table 4
Computational comparisons with corresponding classification accuracies Acc (%). Naive Bayes demonstrates the lowest overall accuracy. Performance of SVM degrades in high dimensional datasets. PCA/SVM does not perform well as the number of features decreases. CSC remains robust although it does not necessarily achieve the highest accuracy in every experiment. Parameter settings k, C of CSC also appear on the table.

Dataset	SVM	PCA/SVM	Naive Bayes	CSC	*k*	*C*
DLBCL	94.8	97.5	75	97.4	3	2E+10
Breast	68	68	62.5	63.6	1	−5E+03
Colon	75.9	92.1	71.4	90.3	3	5E+09
DBWorld	88	88	57.1	89	1	2E+03
Mushroom	100	100	88.1	98.9	10	−1E+03
Spambase	91	66	56.3	87.9	10	5E+03

an advantage over other popular models for high dimensional binary classification.

Potential future research directions include a *cost sensitive* version for *imbalanced classification* problems where the sample numbers of one class greatly outperform the samples of the other. Imbalanced classification problems are common in many business analytics areas [33] and in quality control [46]. In this setup one of the most popular cost sensitive algorithmic schemes is SVM; however, it is well known that it does not perform well for such large number of features. Therefore, alternative algorithms able to simultaneously handle high dimensional datasets and the problem of imbalanced classes are particularly useful for a number of applications.

Another extension is the development of the stream mining version that will incrementally retrain as new training data samples arrive in the form of a data stream. Incremental learning is useful in cases where the full retraining of a model is not desired. Such extensions have been proposed for generic SVM [9,11] and other classifiers [31,16,10,12].

Lastly, a robust optimization version of this algorithm needs to be proposed for handling datasets that are inexact or uncertain. In these cases the *robust counterpart* of the optimization problem needs to be defined and the solution corresponds to the worst case realization of the uncertain data [45,44].

Acknowledgements

Work of Panos M. Pardalos was conducted at National Research University Higher School of Economics and supported by RSF Grant 14-41-00039.

References

[1] Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences 1999;96(12):6745–50.
[2] Belloni Alexandre VC, Hansen C. High-dimensional methods and inference on structural and treatment effects. The Journal of Economic Perspectives 2014;28:29–50.
[3] Bennett J, Lanning S. The netflix prize. In: Proceedings of the KDD cup and workshop; 2007. p. 6.
[4] Beyer K, Goldstein J, Ramakrishnan R, Shaft U. When is nearest neighbor meaningful? Database Theory ICDT 1999;99:217–35.
[5] Bjorck Å, Golub GH. Numerical methods for computing angles between linear subspaces. Mathematics of Computation 1973;27(123):579–94.
[6] Blum AL, Langley P. Selection of relevant features and examples in machine learning. Artificial Intelligence 1997;97(1):245–71.
[7] Campbell J, Lo AW, M. A. The econometrics of financial markets. Princeton, New Jersey, USA: Princeton University Press; 1997.
[8] Carrizosa E, Morales DR. Supervised classification and mathematical optimization. Computers and Operations Research 2013;40(1):150–65.
[9] Cauwenberghs G, Poggio T. Incremental and decremental support vector machine learning. In: Advances in neural information processing systems; 2001. p. 409–15.
[10] Cifarelli C, Guarracino MR, Seref O, Cuciniello S, Pardalos PM. Incremental classification with generalized eigenvalues. Journal of Classification 2007;24 (2):205–19.
[11] Diehl CP, Cauwenberghs G. SVM incremental learning, adaptation and optimization. In: Proceedings of the international joint conference on neural networks, vol. 4. IEEE; Portland, Oregon, USA, 2003. p. 2685–90.
[12] Dulá J, López F. DEA with streaming data. Omega 2013;41(1):41–7.
[13] Edelman Alan TAA, Smith ST. The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications 1998;20 (2):303–53.
[14] Fenn MB, Pappu V. Data mining for cancer biomarkers with raman spectroscopy. In: Data mining for biomarker discovery; 2012. p. 143–68.
[15] Golub GH, Van Loan CF. Matrix computations, vol. 3. JHU Press; Baltimore, Maryland, USA, 2012.
[16] Guarracino MR, Cuciniello S, Feminiano D. Incremental generalized eigenvalue classification on data streams. In: international workshop on data stream management and mining; 2009. p. 1–12.
[17] Guyon I, Elisseeff A. An introduction to variable and feature selection. The Journal of Machine Learning Research 2003;3:1157–82.
[18] Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Machine Learning 2002;46(1):389–422.
[19] Hamm J, Lee DD. Grassmann discriminant analysis: a unifying view on subspace-based learning. In: Proceedings of the 25th international conference on machine learning. ACM; Helsinki, Finland, 2008. p. 376–83.
[20] Joseph Hassell BA-M, Arpinar IB. Ontology-driven automatic entity disambiguation in unstructured text, vol. 4273. Berlin, Heidelberg: Springer; 2006.
[21] Jiang H, Deng Y, Chen H, Tao L, Sha Q, Chen J, et al. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. BMC Bioinformatics 2004;5(1):81.
[22] Johnstone IM, Titterington DM. Statistical challenges of high-dimensional data. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 2009;367(1906):4237–53.
[23] Jolliffe I. Principal component analysis. New York, USA: Springer; 2002.
[24] Kampa K, Mehta S, Chou C, Chaovalitwongse W, Grabowski T. Sparse optimization in feature selection: application in neuroimaging. Journal of Global Optimization 2014;59(2–3):439–57.
[25] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. International Joint Conference on Artificial Intelligence 1995;14(2):1137–45.
[26] Köppen M. The curse of dimensionality. In: 5th online world conference on soft computing in industrial applications (WSC5); 2000. p. 4–8.
[27] Laaksonen J. Local subspace classifier. In: Artificial neural networks, ICANN'97. Springer; Lausanne, Switzerland, 1997. p. 637–42.
[28] Lee SM, et al. Spam detection using feature selection and parameters optimization. Intelligent and Software Intensive Systems (CISIS) 2010:883–8.
[29] Liu H, Motoda H. Feature extraction, construction and selection: a data mining perspective. Norwell, Massachusetts, USA: Springer; 1998.
[30] López FG, Torres MG, Batista BM, Pérez JAM, Moreno-Vega JM. Solving feature subset selection problem by a parallel scatter search. European Journal of Operational Research 2006;169(2):477–89.
[31] Pang S, Ozawa S, Kasabov N. Incremental linear discriminant analysis for classification of data streams. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 2005;35(5):905–14.
[32] Pazzani MJ, Billsus D. Content-based recommendation systems. In: The adaptive web. Springer, Berlin Heidelberg, Germany, 2007. p. 325–41.
[33] Razzaghi T, Otero A, Xanthopoulos P. Imbalanced classification for business analytics. In: Encyclopedia of business analytics and optimization. IGI Global; 2014. p. 1145–54.
[34] Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics 2007;23(19):2507–17.
[35] Satsangi A, Zaiane OR. Contrasting the contrast sets: an alternative approach. In: 11th international database engineering and applications symposium; 2007. p. 114–9.
[36] Shanmugam R, Johnson C. At a crossroad of data envelopment and principal component analyses. Omega 2007;35(4):351–64.
[37] Shipp MA, Ross KN, Tamayo P, Weng AP, Kutok JL, Aguiar RC, et al. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nature Medicine 2002;8(1):68–74.
[38] Stevens JP. Applied multivariate statistics for the social sciences. New York, NY, USA: Routledge; 2012.
[39] Tseng T-LB, Huang C-C. Rough set-based approach to feature selection in customer relationship management. Omega 2007;35(4):365–83.
[40] Unler A, Murat A. A discrete particle swarm optimization method for feature selection in binary classification problems. European Journal of Operational Research 2010;206(3):528–39.
[41] van't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AA, Mao M, et al. Gene expression profiling predicts clinical outcome of breast cancer. Nature 2002;415(6871):530–6.
[42] Vidal R, et al. Generalized principal component analysis, vol. 1. Electronics Research Laboratory, College of Engineering, University of California; 2006.
[43] Wang FK, Du TCT. Using principal component analysis in process performance for multivariate data. Omega 2000;28:185–94.
[44] Xanthopoulos P, Guarracino MR, Pardalos PM. Robust generalized eigenvalue classifier with ellipsoidal uncertainty. Annals of Operations Research 2014;216 (1):327–42.
[45] Xanthopoulos P, Pardalos PM, Trafalis TB. Robust data mining. New York, NY, USA: Springer; 2010.
[46] Xanthopoulos P, Razzaghi T. A weighted support vector machine method for control chart pattern recognition. Computers and Industrial Engineering 2014;70:134–49.
[47] Yang J, Olafsson S. Optimization-based feature selection with adaptive instance sampling. Computers and Operations Research 2006;33 (11):3088–106.