Computational Statistics and Data Analysis 55 (2011) 789–801

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

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

# Supervised multidimensional scaling for visualization, classification, and bipartite ranking

Daniela M. Witten [a,*], Robert Tibshirani [b]

[a] Department of Statistics, Stanford University, 390 Serra Mall, Stanford CA 94305, USA

[b] Departments of Statistics and Health Research & Policy, Stanford University, 390 Serra Mall, Stanford CA 94305, USA

ARTICLE INFO

*Article history:*
Received 3 November 2009
Received in revised form 1 July 2010
Accepted 1 July 2010
Available online 18 July 2010

*Keywords:*
Classification
Multidimensional scaling
Unidimensional scaling
Unsupervised learning
Majorization
Ranking

ABSTRACT

*Least squares multidimensional scaling* (MDS) is a classical method for representing a $n \times n$ dissimilarity matrix $\mathbf{D}$. One seeks a set of configuration points $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^S$ such that $\mathbf{D}$ is well approximated by the Euclidean distances between the configuration points: $D_{ij} \approx \|\mathbf{z}_i - \mathbf{z}_j\|_2$. Suppose that in addition to $\mathbf{D}$, a vector of associated binary class labels $\mathbf{y} \in \{1, 2\}^n$ corresponding to the $n$ observations is available. We propose an extension to MDS that incorporates this outcome vector. Our proposal, *supervised multidimensional scaling* (SMDS), seeks a set of configuration points $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^S$ such that $D_{ij} \approx \|\mathbf{z}_i - \mathbf{z}_j\|_2$, and such that $z_{is} > z_{js}$ for $s = 1, \ldots, S$ tends to occur when $y_i > y_j$. This results in a new way to visualize the observations. In addition, we show that SMDS leads to a method for the classification of test observations, which can also be interpreted as a solution to the bipartite ranking problem. This method is explored in a simulation study, as well as on a prostate cancer gene expression data set and on a handwritten digits data set.



## 1. Introduction

Consider a symmetric $n \times n$ matrix $\mathbf{D}$ that contains the pairwise dissimilarities of a set of $n$ observations. That is, $D_{ij}$ represents the dissimilarity between observations $i$ and $j$. For instance, $\mathbf{D}$ could be derived from a set of $n$ observation vectors measured on $p$ features, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$; then, $D_{ij}$ might be the Euclidean distance between observations $\mathbf{x}_i$ and $\mathbf{x}_j$. Alternatively, no underlying observation vectors might be available.

Given such an object $\mathbf{D}$, one might wish to visualize the $n$ observations that it represents. *Multidimensional scaling* (MDS) refers to a set of methods for this task. A number of variations of MDS exist; these include Sammon mapping, classical scaling, and Shephard–Kruskal nonmetric scaling (Shepard, 1962; Kruskal, 1964; Green, 1989; Cox and Cox, 1994; Borg and Groenen, 2005; Buja et al., 2008; Hastie et al., 2009). In this paper, we will consider *least squares MDS*, in which one seeks $n$ $S$-dimensional configuration points $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^S$ that minimize the *MDS criterion* or *stress function*

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(D_{ij} - \|\mathbf{z}_i - \mathbf{z}_j\|_2)^2. \tag{1}$$

If $S = 2$, then the configuration points lie in $\mathbb{R}^2$ and can be plotted, leading to a convenient tool for the visualization of $\mathbf{D}$.

Now suppose that a vector of outcome measurements is also available: $\mathbf{y} \in \mathbb{R}^n$, where $y_i$ is the outcome associated with observation $i$. A special case of the outcome vector $\mathbf{y}$ occurs when $y_i \in \{1, 2\}$; then, $\mathbf{y}$ can be interpreted as a vector of binary class labels. One might like to visualize the $n \times n$ dissimilarity matrix $\mathbf{D}$ in a way that incorporates information about the outcome. In this paper, we propose a supervised version of MDS, which we refer to as *supervised multidimensional scaling*

* Corresponding author. Tel.: +1 6502486323.
*E-mail addresses:* dwitten@stanford.edu (D.M. Witten), tibs@stat.stanford.edu (R. Tibshirani).

0167-9473/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2010.07.001

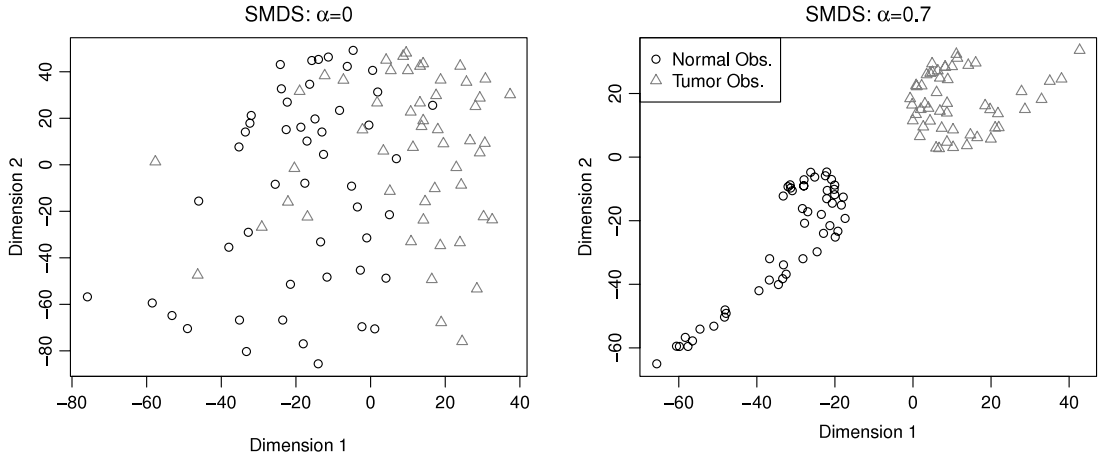


**Fig. 1.** A plot of the configuration points obtained on the prostate data set, using MDS (left; MDS is equivalent to SMDS with $\alpha = 0$) and SMDS with $\alpha = 0.7$ (right). In the MDS plot, there is little separation between the tumor and normal samples.

(SMDS), that does just this. We supervise the construction of the configuration points $\mathbf{z}_i$ and $\mathbf{z}_j$ based on the value of the outcome measurements $y_i$ and $y_j$. If $y_i \approx y_j$ then we construct configuration points $\mathbf{z}_i$ and $\mathbf{z}_j$ that are not too far apart; otherwise, we construct configuration points such that $\|\mathbf{z}_i - \mathbf{z}_j\|_2$ is large.

As an example, we apply SMDS to a data set of gene expression values measured on microarrays, presented in Singh et al. (2002). Each of the 102 observations represents a prostate sample; 52 observations are tumor samples and 50 are non-tumor, or normal. The expression levels of 6033 genes are available. MDS does not result in clear separation between the two classes, whereas SMDS does. This is not surprising, since SMDS is supervised. The configuration points obtained using SMDS can potentially provide insight into the data that is not available from the MDS configuration (Fig. 1).

The rest of this paper is organized as follows. We present the SMDS criterion in Section 2. In that section, we further demonstrate SMDS's potential as a tool for data visualization. In Section 3 we show that SMDS can be used to classify unlabeled test observations when the outcome vector $\mathbf{y}$ is binary. In Section 4, we discuss the relationship of SMDS with the bipartite ranking problem. We extend SMDS to the case of $K > 2$ classes in Section 5. Section 6 contains the Discussion.

## 2. The supervised multidimensional scaling proposal

### 2.1. The criterion for supervised multidimensional scaling

As mentioned in Section 1, the MDS stress function (1) seeks configuration points $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^S$ such that $D_{ij} \approx \|\mathbf{z}_i - \mathbf{z}_j\|_2$. MDS is unsupervised, in that it does not make use of an outcome vector $\mathbf{y} \in \mathbb{R}^n$ that might be available.

Our proposal directly extends (1) to make use of such an outcome. We want configuration points such that $D_{ij} \approx \|\mathbf{z}_i - \mathbf{z}_j\|_2$, *and* such that $z_{is} > z_{js}$ will tend to occur when $y_i > y_j$, for all $s = 1, \ldots, S$. We define *supervised multidimensional scaling* (SMDS) as the solution to the problem

$$\underset{\mathbf{z}_1,\ldots,\mathbf{z}_n \in \mathbb{R}^S}{\text{minimize}} \left\{ \frac{1}{2}(1-\alpha) \sum_{i=1}^{n}\sum_{j=1}^{n} (D_{ij} - \|\mathbf{z}_i - \mathbf{z}_j\|_2)^2 + \alpha \sum_{i,j:y_j>y_i} (y_j - y_i) \sum_{s=1}^{S} \left( \frac{D_{ij}}{\sqrt{S}} - (z_{js} - z_{is}) \right)^2 \right\}, \tag{2}$$

where $\alpha \in [0, 1]$ is a tuning parameter. When $\alpha = 0$, the SMDS criterion (2) reduces to the MDS stress function (1). As $\alpha$ increases, the criterion (2) becomes increasingly supervised. From the form of the criterion (2), we observe that when $y_j > y_i$ and $\alpha$ is large, $z_{js} > z_{is}$ will tend to occur for all $s = 1, \ldots, S$. This will result in observations with larger values of the outcome having larger values in their configuration points, and observations with smaller outcome values having smaller configuration points. Note that the $\sqrt{S}$ in the second term of (2) is required in order for the first and second terms to be on the same scale, since the second term contains a summation over $S$ elements.

In this paper, we consider the case where the outcome vector $\mathbf{y}$ represents two class labels. Specifically, suppose that $y_i \in \{1, 2\}$, and that observation $i$ belongs to class $y_i$. In this case, (2) can be re-written as

$$\underset{\mathbf{z}_1,\ldots,\mathbf{z}_n \in \mathbb{R}^S}{\text{minimize}} \left\{ \frac{1}{2}(1-\alpha) \sum_{i=1}^{n}\sum_{j=1}^{n} (D_{ij} - \|\mathbf{z}_i - \mathbf{z}_j\|_2)^2 + \alpha \sum_{i:y_i=1} \sum_{j:y_j=2} \sum_{s=1}^{S} \left( \frac{D_{ij}}{\sqrt{S}} - (z_{js} - z_{is}) \right)^2 \right\}. \tag{3}$$

It is clear that as $\alpha \in [0, 1]$ increases, so will the separation between the configuration points of the observations corresponding to the two classes. In particular, observations in class 2 will tend to have larger $z_{is}$ than observations in class 1, for all $s = 1, \ldots, S$. Criterion (3) was used to create the right-hand panel of Fig. 1.

We now briefly consider the problem of solving (3). When $\alpha < 1$, this criterion is non-convex because the MDS stress function (1) is non-convex. The topic of optimization of the MDS stress function has been studied extensively, and is discussed for instance in Borg and Groenen (2005). An iterative majorization approach to solve (3) is discussed in the Appendix.

When data visualization is the goal, we recommend performing SMDS for a range of values of $\alpha$ in order to see if the resulting plots yield insight into the data.

Note that the criterion (3) encourages each dimension of $\mathbf{z}_1, \ldots, \mathbf{z}_n$ to have the property that $z_{js} > z_{is}$ if $y_j = 2$ and $y_i = 1$. One could instead propose a criterion in which only one dimension is supervised. For instance,

$$\underset{\mathbf{z}_1,\ldots,\mathbf{z}_n \in \mathbb{R}^S}{\text{minimize}} \left\{ \frac{1}{2}(1-\alpha) \sum_{i=1}^{n}\sum_{j=1}^{n}(D_{ij} - \|\mathbf{z}_i - \mathbf{z}_j\|_2)^2 + \alpha \sum_{i:y_i=1}\sum_{j:y_j=2}(D_{ij} - (z_{j1} - z_{i1}))^2 \right\} \tag{4}$$

is one possibility. However, this criterion would lead to clear separation between the two classes in the first dimension, but not in later dimensions. In fact, the two classes may be even less separated in dimensions $2, \ldots, S$ than in MDS in order to correct for the excessive separation between the two classes in the first dimension. Therefore, this criterion would not achieve the stated goal of constructing configuration points that approximate the dissimilarities between the observations while also separating the two classes.

### 2.2. Increased stress due to supervision

The SMDS criterion (3) is composed of two terms, the MDS *stress term*

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(D_{ij} - \|\mathbf{z}_i - \mathbf{z}_j\|_2)^2 \tag{5}$$

and a *supervised term*

$$\sum_{i:y_i=1}\sum_{j:y_j=2}\sum_{s=1}^{S}\left(\frac{D_{ij}}{\sqrt{S}} - (z_{js} - z_{is})\right)^2. \tag{6}$$

As $\alpha$ increases, the stress of the resulting solution is non-decreasing. One can interpret the increase in the stress term for a given value of $\alpha$, relative to the MDS stress that occurs when $\alpha = 0$, as the amount that the approximation to the dissimilarity matrix is distorted due to supervision.

### 2.3. Examples of supervised multidimensional scaling

We study the differences between MDS and SMDS, as well as the effect of the tuning parameter $\alpha$, under two simple models involving two equally-sized classes. There are $n$ observations and $p$ features, as follows.

- For the *constant* model (so called because there is a constant mean for the observations in each class), the $i$th observation is generated as follows:

$$\mathbf{x}_i \sim \begin{cases} N(-\mathbf{0.4}, \mathbf{I}_p) & \text{if observation } i \text{ is in class 1} \\ N(\mathbf{0.4}, \mathbf{I}_p) & \text{if observation } i \text{ is in class 2.} \end{cases} \tag{7}$$

  Here, $\mathbf{I}_p$ is the $p \times p$ identity matrix, and $\mathbf{0.4}$ is a $p$-vector containing 0.4's.
- For the *two-sided* model (so called because the observations in class 2 fall into two distinct groups), the $i$th observation is generated as follows:

$$\mathbf{x}_i \sim \begin{cases} N(\mathbf{0}, \mathbf{I}_p) & \text{if observation } i \text{ is in class 1} \\ N(\mathbf{1}, \mathbf{I}_p) \text{ or } N(-\mathbf{1}, \mathbf{I}_p) & \text{with equal probability if observation } i \text{ is in class 2.} \end{cases} \tag{8}$$

  Here, $\mathbf{0}$ and $\mathbf{1}$ are $p$-vectors containing 0's and 1's, respectively.

Each simulated data set consists of $n = 100$ observations and $p = 10$ features. Fig. 2 shows the 2-dimensional configuration points obtained from performing SMDS with three values of $\alpha$: $\alpha = 0$, $\alpha = 0.4$, and $\alpha = 0.8$. As $\alpha$ increases, so too does the separation between the configuration points corresponding to the observations in the two classes. The stress term and the supervised term (5) and (6) are also reported. As $\alpha$ increases, the stress term increases and the supervised term decreases.

## 3. Supervised multidimensional scaling for visualization and classification of test observations

### 3.1. Extension of the SMDS criterion to test observations

Consider the setting of previous sections, in which there are $n$ observations for which a $n \times n$ matrix of pairwise dissimilarities and a vector $\mathbf{y} \in \{1, 2\}^n$ of class labels is available. We will refer to these as the *training* observations.

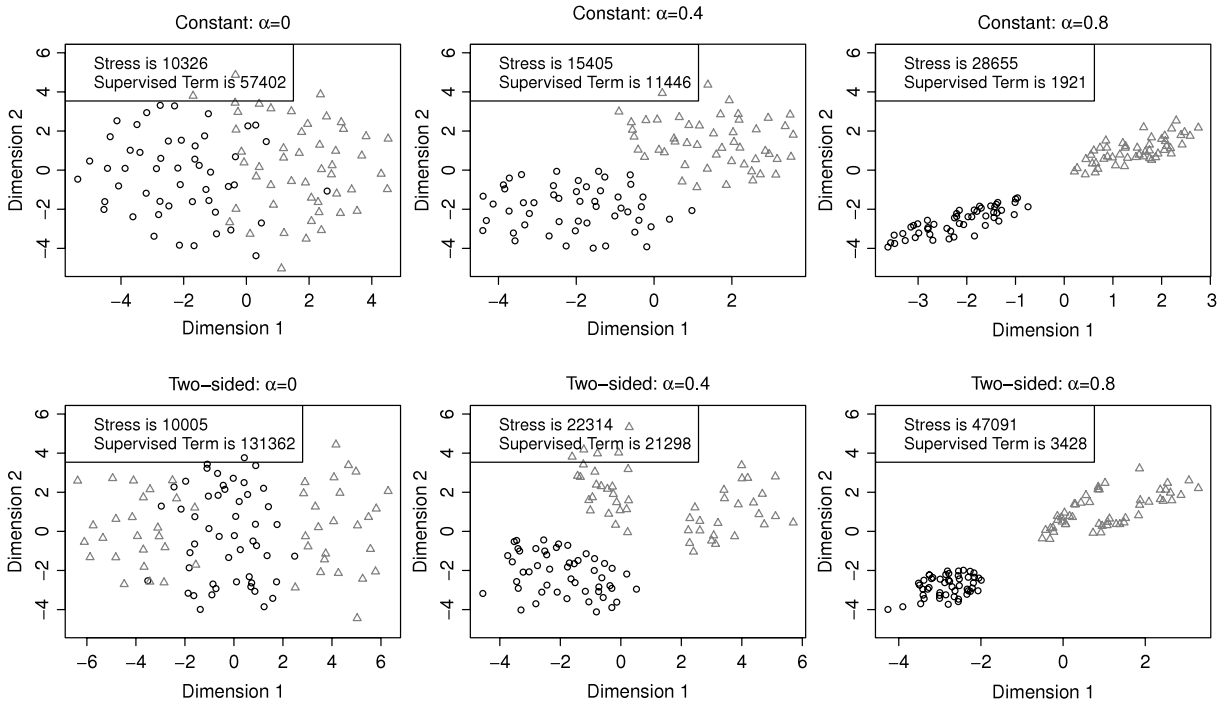


**Fig. 2.** The configuration points obtained using $\alpha = 0, 0.4, 0.8$ are shown for two simple simulations. When $\alpha = 0$, the SMDS configuration points are the same as for MDS. As $\alpha$ increases, so too does the extent of supervision, and therefore also the separation between the configuration points for the two classes. Class 2 is shown as triangles and class 1 is shown as circles.

In addition, suppose there is a *test* observation for which the class label $y_{n+1}$ is unknown, but for which the pairwise dissimilarity $D_{i,n+1}$ with observation $i$ is known, for $i = 1, \ldots, n$. We wish to:

1. Extend the SMDS criterion (3) in order to estimate the configuration point $\mathbf{z}_{n+1}$ of the test observation.
2. Predict the class label of the test observation, $y_{n+1}$.

As we will see, these two tasks are closely related. We will refer to the configuration points $\mathbf{z}_1, \ldots, \mathbf{z}_n$ obtained by performing SMDS on the training observations as the *training configuration points*. We extend the SMDS criterion (3):

$$\underset{\mathbf{z}_{n+1}\in\mathbb{R}^S, y_{n+1}\in\{1,2\}}{\text{minimize}} \left\{ \frac{1}{2}(1-\alpha)\sum_{i=1}^{n+1}\sum_{j=1}^{n+1}(D_{ij} - \|\mathbf{z}_i - \mathbf{z}_j\|_2)^2 + \alpha \sum_{i:y_i=1, 1\le i\le n+1} \sum_{j:y_j=2, 1\le j\le n+1} \sum_{s=1}^{S}\left(\frac{D_{ij}}{\sqrt{S}} - (z_{js} - z_{is})\right)^2 \right\}. \quad (9)$$

In other words, we extend (3) by including the test observation in the objective and holding the training configuration points fixed. We minimize (9) with respect to the test configuration point and the test class label, both of which are unknown.

Ignoring terms in (9) that do not involve $\mathbf{z}_{n+1}$ or $y_{n+1}$, we find that the values of $\mathbf{z}_{n+1}$ and $y_{n+1}$ that solve (9) also solve

$$\underset{\mathbf{z}_{n+1}\in\mathbb{R}^S, y_{n+1}\in\{1,2\}}{\text{minimize}} \left\{ (1-\alpha)\sum_{i=1}^{n}(D_{i,n+1} - \|\mathbf{z}_i - \mathbf{z}_{n+1}\|_2)^2 + \alpha 1_{\{y_{n+1}=1\}} \sum_{i:y_i=2, 1\le i\le n} \sum_{s=1}^{S}\left(\frac{D_{i,n+1}}{\sqrt{S}} - (z_{is} - z_{n+1,s})\right)^2 \right.$$
$$\left. + \ \alpha 1_{\{y_{n+1}=2\}} \sum_{i:y_i=1, 1\le i\le n} \sum_{s=1}^{S}\left(\frac{D_{i,n+1}}{\sqrt{S}} - (z_{n+1,s} - z_{is})\right)^2 \right\}. \quad (10)$$

We define some additional notation: Let $\mathbf{D}_{n+1} = [D_{1,n+1}, \ldots, D_{n,n+1}]^T$, and let

$$h_1(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) = (1-\alpha)\sum_{i=1}^{n}(D_{i,n+1} - \|\mathbf{z}_i - \mathbf{z}_{n+1}\|_2)^2 + \alpha \sum_{i:y_i=2, 1\le i\le n} \sum_{s=1}^{S}\left(\frac{D_{i,n+1}}{\sqrt{S}} - (z_{is} - z_{n+1,s})\right)^2, \quad (11)$$

$$h_2(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) = (1-\alpha)\sum_{i=1}^{n}(D_{i,n+1} - \|\mathbf{z}_i - \mathbf{z}_{n+1}\|_2)^2 + \alpha \sum_{i:y_i=1, 1\le i\le n} \sum_{s=1}^{S}\left(\frac{D_{i,n+1}}{\sqrt{S}} - (z_{n+1,s} - z_{is})\right)^2. \quad (12)$$

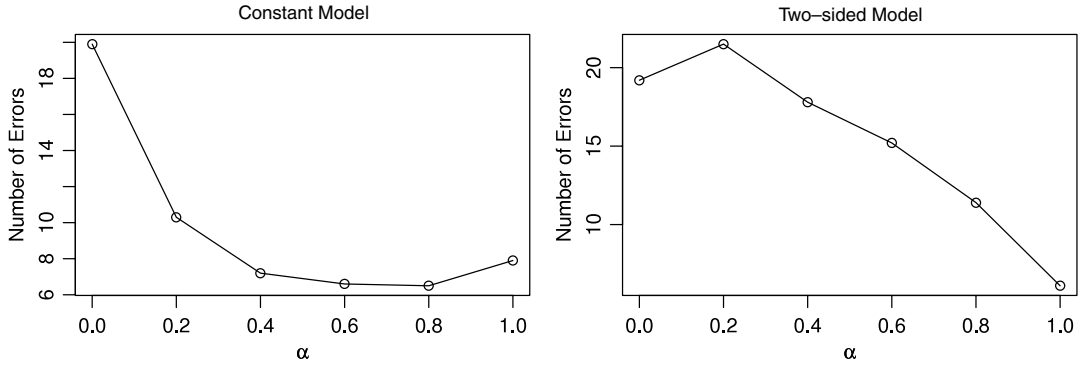


**Fig. 3.** Training and test observations were generated under the constant and two-sided models of Section 2.3. SMDS was performed on the training data for a range of values of $\alpha$, and test set classification was performed. Reported test errors are averaged over 10 repetitions.

That is, (11) equals the objective of (10) in the case that $y_{n+1} = 1$, and (12) is the objective of (10) in the case that $y_{n+1} = 2$. Then, the solution to (10) is as follows:

$$\hat{y}_{n+1} = \underset{k\in\{1,2\}}{\operatorname{argmin}} \left\{ \min_{\mathbf{z}_{n+1}} h_k(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) \right\}, \tag{13}$$

$$\hat{\mathbf{z}}_{n+1} = \underset{\mathbf{z}_{n+1}\in\mathbb{R}^S}{\operatorname{argmin}} \left\{ h_{\hat{y}_{n+1}}(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) \right\}. \tag{14}$$

So (10) involves assigning an observation to class 1 if $\min_{\mathbf{z}_{n+1}} h_1(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) - \min_{\mathbf{z}_{n+1}} h_2(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) < 0$.

However, it turns out that empirically better results are obtained using a slightly different procedure. For some cutpoint $c$, if $\min_{\mathbf{z}_{n+1}} h_1(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) - \min_{\mathbf{z}_{n+1}} h_2(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) < c$ then $\hat{y}_{n+1} = 1$; otherwise, $\hat{y}_{n+1} = 2$. The test configuration point is then given by (14). If $c = 0$, then this is equivalent to solving (9). Instead, we set $c$ equal to the $\frac{\sum_{i=1}^n 1_{y_i=1}}{n}$ quantile of $\{\min_{\mathbf{z}_j} h_1(\mathbf{D}_j, \mathbf{z}_j) - \min_{\mathbf{z}_j} h_2(\mathbf{D}_j, \mathbf{z}_j)\}_{j=1}^n$. In other words, the cutpoint is set to the $\frac{\sum_{i=1}^n 1_{y_i=1}}{n}$ quantile of the results obtained if testing is performed on the training observations.

If $S = 2$, then one can display the training and test configuration points in a single plot in order to visualize the data. An algorithm for solving (9) is proposed in the Appendix. A simulation study evaluating the classification performance of SMDS is presented in Section 4.3.

### 3.2. Comments on the tuning parameter $\alpha$

SMDS can be used to classify test observations whenever the tuning parameter $\alpha$ is positive. We suggest selecting $\alpha$ by cross-validation, as described for instance in Hastie et al. (2009).

Supervision of the training configuration points will increase as $\alpha$ increases, but this does not imply that $\alpha = 1$ will result in the best possible classification of test observations. We illustrate this point in a small simulation study, using the models of Section 2.3 with $p = 10$ features. A training set of $n = 40$ observations and an equally-sized test set were generated. SMDS was performed on the training data with various values of $\alpha$, and then classification of the test observations was performed. The resulting test errors are shown in Fig. 3. In the two-sided model, the fewest test errors are obtained when $\alpha = 1$, whereas in the constant model, the fewest test errors are obtained for an intermediate value of $\alpha$.

### 3.3. Prostate data

We try out the classification approach for SMDS on the prostate data set presented in Section 1. The 102 samples were split into equally-sized training and test sets. SMDS with $\alpha = 0.7$ was performed on half of the observations in order to obtain training configuration points. Then test configuration points were obtained for the test observations. The resulting training and test configuration points are shown in Fig. 4. When $\alpha = 0.7$, 8/51 test observations are misclassified. Though SMDS's classification error rate in this example is not as low as that of other methods (for instance, SVM with a linear kernel gives 6/51 test errors), SMDS also has value as a visualization tool.

### 3.4. USPS handwritten digit data

We apply SMDS to the USPS handwritten digits data set, which consists of 7291 training $16 \times 16$ grayscale images of handwritten digits 0–9, as well as 2007 test images. We performed SMDS on the dissimilarity matrix obtained using *tangent distance*, which has been shown to perform quite well on this problem (Simard et al., 1993; Hastie and Simard, 1998). To reduce the data to a two-class problem, we focus on differentiating between digits 3 and 8. When we restrict the problem to these two digits, the data reduces to a training set of 1200 observations and a test set of 332 observations.

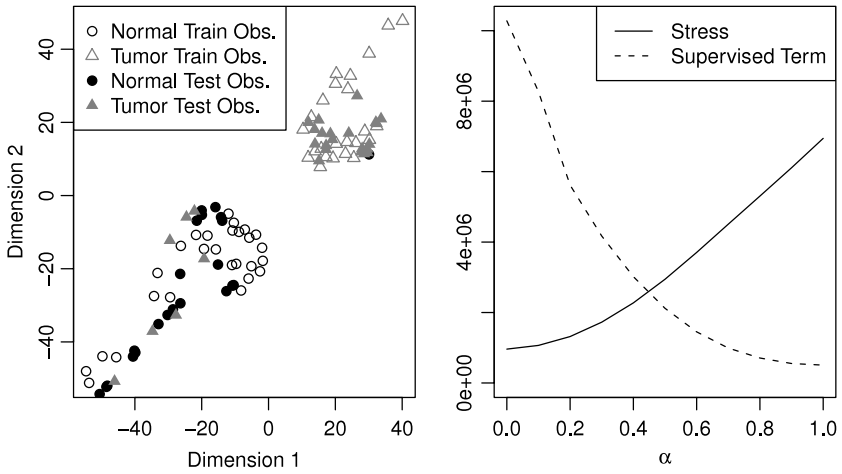


**Fig. 4.** *Left*: On the prostate data, SMDS with $\alpha = 0.7$ was trained on half of the observations and tested on the remaining observations. When $\alpha = 0.7$, most test configuration points are located near the training configuration points for their class. 8/51 test observations are misclassified. *Right*: SMDS was performed using a range of values of $\alpha$. For each value of $\alpha$, the resulting stress (5) and supervised term (6) are shown.

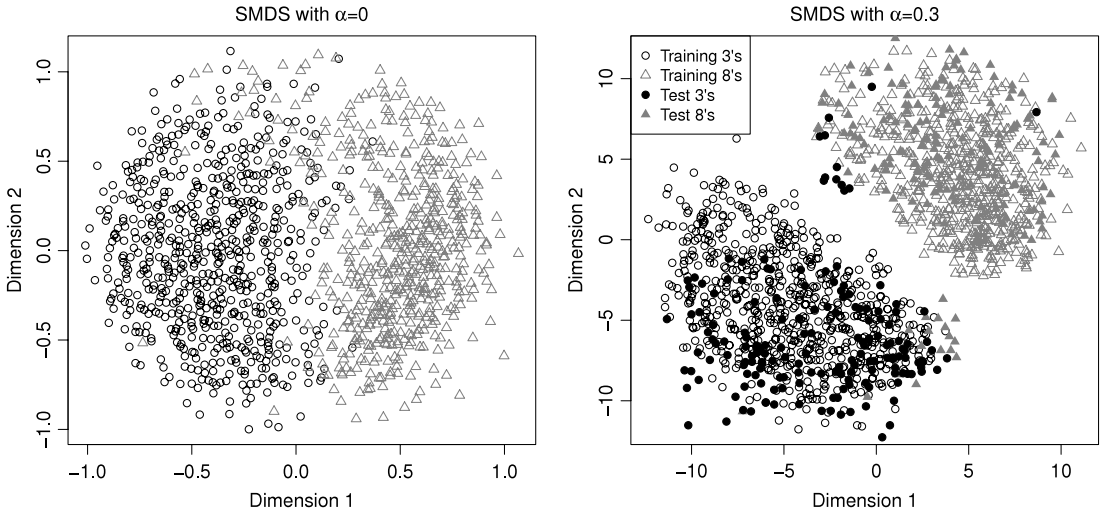


**Fig. 5.** Subset of the USPS handwritten digit data set consisting of 3's and 8's. *Left panel*: The configuration points obtained by applying MDS (SMDS with $\alpha = 0$) to the training set. *Right panel*: SMDS with $\alpha = 0.3$ was trained on the training observations and tested on the test observations, yielding 29/332 test set classification errors.

We first performed MDS, or equivalently SMDS with $\alpha = 0$, on the $1200 \times 1200$ dissimilarity matrix consisting of the training observations. There is a fair amount of separation between the 3's and the 8's (Fig. 5, left panel); however, a number of 3's and 8's lie close to observations of the opposite class. Performing SMDS with $\alpha = 0.3$ on the training observations leads to far better separation (Fig. 5, right panel). We then estimated the configuration points and predicted the class labels for the 332 test observations (Fig. 5, right panel). Most of the test configuration points are located near the training configuration points for the correct digit. However, 29/332 of the test configuration points are misclassified. Note that these misclassified observations tend to be located near the boundary of the configuration points for the other class, suggesting that these observations are not similar to the configuration points in either class. Fig. 6 shows the actual handwritten test digits imposed onto the test configuration points obtained from running SMDS with $\alpha = 0.3$.

This data set has been studied extensively, and extremely low error rates have been obtained on the full 10-class problem using for instance 1-nearest neighbors (an error rate of 2.6% is reported in Hastie and Simard (1998)). SMDS is not a serious contender in terms of classification error, but it does provide a nice visualization tool for the data.

### 3.5. A related proposal

A classification method based on MDS was previously proposed in Cox and Ferry (1993). Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ denote a data matrix consisting of $n$ training observations with measurements on $p$ features, corresponding to the dissimilarity matrix

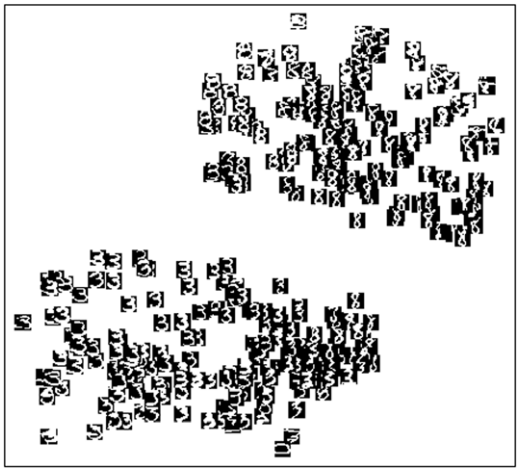

**Fig. 6.** A plot of the test configuration points obtained on digits 3 and 8 of the USPS handwritten digit data set using SMDS with $\alpha = 0.3$. The actual handwritten test digit is superimposed on top of each test configuration point.

$\mathbf{D} \in \mathbb{R}^{n\times n}$. First, the matrix $\tilde{\mathbf{D}}$ is computed as

$$\tilde{D}_{ij} = \begin{cases} \gamma D_{ij} & \text{if } i \text{ and } j \text{ are in different classes} \\ D_{ij} & \text{if } i \text{ and } j \text{ are in the same class,} \end{cases} \tag{15}$$

where $\gamma \geq 1$ is a tuning parameter. MDS is performed using $\tilde{\mathbf{D}}$ in order to obtain configuration points $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^S$. Next, ordinary least squares regression is used to fit the model

$$z_{is} = a_s + \mathbf{x}_i^T \boldsymbol{\beta}_s + \epsilon_{is} \tag{16}$$

for each $s = 1, \ldots, S$, where $\mathbf{x}_i$ indicates the $i$th observation, $a_s$ is an intercept term, $\boldsymbol{\beta}_s \in \mathbb{R}^p$ is a coefficient vector, and $\boldsymbol{\epsilon} \in \mathbb{R}^{n\times S}$ is a matrix of noise terms. Then, a test observation $\mathbf{x}_{n+1} \in \mathbb{R}^p$ is assigned a configuration point based on the linear mapping

$$z_{n+1,s} = \hat{a}_s + \mathbf{x}_{n+1}^T \hat{\boldsymbol{\beta}}_s, \tag{17}$$

where $\hat{a}_s$ and $\hat{\boldsymbol{\beta}}_s$ are the fitted coefficients for the model (16). Finally, linear discriminant analysis (LDA) is performed in the space of configuration points in order to classify the test observation.

There are a few major differences between SMDS and the proposal of Cox and Ferry (1993). The latter relies on specific modeling assumptions: namely, that there is a linear relationship between the raw data and the configuration points, and that there is a linear boundary between the classes in the space of configuration points. These assumptions can be weakened, for instance by allowing quadratic terms in (16) or by performing quadratic discriminant analysis instead of LDA. However, SMDS is inherently more flexible, and we expect it to perform better than Cox and Ferry (1993) in settings where the boundary between the classes is nonlinear. Moreover, the proposal of Cox and Ferry (1993) for obtaining test set configuration points is somewhat unnatural, as it does not involve minimizing the stress. On the other hand, SMDS provides a natural approach for obtaining test set configuration points. We compare the performances of our proposal and that of Cox and Ferry (1993) on three simulated examples in Section 4.3.

## 4. Supervised multidimensional scaling for bipartite ranking

In this section, we show that the proposal of Section 3 can be seen as a solution to the bipartite ranking problem when the configuration points used are one-dimensional: that is, when $S = 1$ in (3).

### 4.1. The bipartite ranking problem

In the past ten or so years, *ranking* has become an active area of research in the machine learning community (see e.g. Cohen et al., 1998; Crammer and Singer, 2001; Freund et al., 2003; Agarwal et al., 2005; Agarwal and Niyogi, 2009). The ranking problem can be formulated as follows. Suppose one has a set of $n$ observations in an instance space $\mathcal{X}$, for which a full or partial ranking is available. One wishes to train a ranking model that can predict the relative ranking of a future observation.

For instance, suppose that one wishes to develop an automated algorithm to rank the e-mails in a person's inbox based on their level of interest. The training data consist of $n$ e-mails, and information about the interest level of each training e-mail is available. For example, it might be known that e-mail 3 is more interesting than e-mail 2, but less interesting than e-mail 1. Alternatively, one might know that some subset of the e-mails are "interesting" and that the remainder are

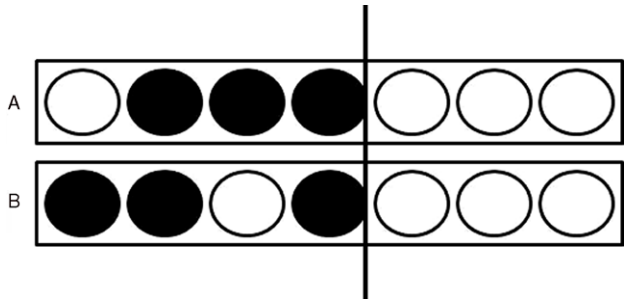


**Fig. 7.** A and B show two different rankings of 7 observations, of which 3 are in class 1 (black) and 4 are in class 2 (white). The cutpoint used to compute classification error is shown as a vertical line. Both rankings have the same classification error of $\frac{1}{7}$, but ranking A has a higher bipartite ranking error than ranking B $\left(\frac{1}{4}\right.$ versus $\left.\frac{1}{12}\right)$.

"uninteresting". The goal is to develop a ranking model to predict the interest level of a new e-mail, relative to the training e-mails.

The *bipartite ranking* problem is a special case of the general ranking problem in which we have $n$ observations in an instance space $\mathcal{X}$, and for the $i$th observation there is a class label $y_i \in \{1, 2\}$. One seeks a function $f : \mathcal{X} \to \mathbb{R}$ that induces a ranking over $\mathcal{X}$. A good ranking function satisfies the property that $f(\mathbf{x}_i) > f(\mathbf{x}_j)$ if $y_i > y_j$ for observations $\mathbf{x}_i$ and $\mathbf{x}_j$.

The bipartite ranking setting is closely related to the binary classification setting, in which one seeks a binary-valued function $h : \mathcal{X} \to \{1, 2\}$ such that $h(\mathbf{x}_i) = y_i$. Given a bipartite ranking algorithm, one could obtain a classifier by classifying an observation to class 2 if $f(\mathbf{x}_i) > c$ and to class 1 otherwise for some threshold $c$. However, the bipartite ranking problem differs from binary classification in that the natural measure for classification performance is the *classification error rate*

$$\frac{1}{n}\sum_{i=1}^{n} 1_{h(\mathbf{x}_i)\neq y_i}, \tag{18}$$

whereas the natural measure of bipartite ranking performance is the *bipartite ranking error rate*, defined as

$$\frac{1}{n_1 n_2}\sum_{i:y_i=1}\sum_{j:y_j=2}\left(1_{f(\mathbf{x}_j)<f(\mathbf{x}_i)} + \frac{1}{2}1_{f(\mathbf{x}_j)=f(\mathbf{x}_i)}\right) \tag{19}$$

where $n_1$ is the number of observations for which $y_i = 1$ and $n_2$ is the number of observations for which $y_i = 2$. (This quantity is better known to statisticians as the complement of the *area under the ROC curve*). That is, in classification, we seek to minimize the number of observations that are assigned to the wrong class. In bipartite ranking, we seek to minimize the number of observation pairs in which an observation in class 1 is assigned a higher ranking than an observation in class 2.

To further illustrate the difference between bipartite ranking and binary classification, Fig. 7 shows two different rankings of a set of seven observations that fall into two classes. In the first example, the bipartite ranking error is large, and in the second it is quite small. Both examples have the same classification error.

Returning to the e-mail example, suppose that the $n$ training e-mails are each labeled as either "interesting" or "not interesting". Then a binary classifier would classify each e-mail in a set of incoming e-mails as either "interesting" or "not interesting". A bipartite ranking algorithm would instead provide a ranking of the incoming e-mails, from "extremely interesting" to "not at all interesting". One could then read the new e-mails in order of predicted interest.

### 4.2. Supervised unidimensional scaling for the ranking problem

Let $\mathbf{D}$ denote a $n \times n$ dissimilarity matrix for the $n$ training observations, and suppose that one wishes to obtain a ranking of the $n$ observations based only on $\mathbf{D}$. The problem seems vague, but a reasonable approach would be to seek a set of $n$ configuration points $z_1, \ldots, z_n \in \mathbb{R}$ such that the interpoint distances of the configuration points approximate the interpoint dissimilarities of the observations: that is, such that $|z_i - z_j| \approx D_{ij}$. In fact, this is a solved problem: the solution is given by least squares *unidimensional scaling* (UDS), the special case of MDS in which the configuration points are one-dimensional. The UDS criterion is

$$\underset{z_1,\ldots,z_n}{\text{minimize}}\left\{\frac{1}{2}\sum_{i,j}(D_{ij} - |z_i - z_j|)^2\right\}. \tag{20}$$

Therefore, UDS provides an unsupervised ranking of the observations.

We now extend UDS to the bipartite ranking setting in which each of $n$ training observations corresponds to a class label, $y_i \in \{1, 2\}$. If an unlabeled test observation is available, we want an automatic way to estimate its ranking and predict its class label. That is, we wish to

1. Extend the UDS criterion (20) so that it results in a ranking that is consistent with the class labels $y_1, \ldots, y_n$ and the inter-observation dissimilarities $D_{ij}$.

**Table 1**
For a number of values of $n$ (the number of observations) and $p$ (the number of features), the classification errors of $L_1$-penalized logistic regression, SMDS with $\alpha = 1$, SMDS with $\alpha$ chosen by cross-validation, the method of Cox and Ferry (1993) (CF), LDA, KNN, SVM with a linear kernel (LK) and SVM with a polynomial kernel of degree 3 (PK) are compared on the test data. Tuning parameters were chosen by cross-validation on the training data. Three models were used: the *two-sided*, *linear*, and *constant* models described in the text. Results are averaged over 50 simulated data sets. Within each row, the two methods with the lowest error rates are shown in bold.

| $p$ | $n$ | Model | $L_1$ logistic | SMDS $\alpha = 1$ | SMDS w/CV | CF | LDA | KNN | SVM LK | SVM PK |
|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 20 | Two-sided | 0.502 | **0.264** | **0.295** | 0.505 | 0.501 | 0.336 | 0.501 | 0.422 |
| 5 | 50 | Two-sided | 0.5012 | **0.2288** | **0.2432** | 0.4976 | 0.5044 | 0.2904 | 0.4996 | 0.4388 |
| 15 | 20 | Two-sided | 0.512 | **0.089** | **0.115** | 0.482 | 0.489 | 0.164 | 0.487 | 0.467 |
| 15 | 50 | Two-sided | 0.5016 | **0.0664** | **0.0664** | 0.5096 | 0.522 | 0.0996 | 0.51 | 0.4224 |
| 5 | 20 | Linear | 0.212 | **0.131** | 0.142 | **0.134** | 0.176 | 0.183 | 0.151 | 0.24 |
| 5 | 50 | Linear | 0.1984 | 0.1324 | **0.13** | **0.1252** | 0.1336 | 0.154 | 0.1388 | 0.19 |
| 15 | 20 | Linear | 0.164 | **0.115** | **0.107** | 0.145 | 0.275 | 0.142 | 0.133 | 0.253 |
| 15 | 50 | Linear | 0.11 | **0.0752** | **0.0744** | 0.0796 | 0.1248 | 0.086 | 0.0796 | 0.2564 |
| 5 | 20 | Constant | 0.312 | 0.289 | 0.264 | **0.239** | **0.254** | 0.314 | 0.257 | 0.318 |
| 5 | 50 | Constant | 0.2812 | 0.2172 | **0.2024** | **0.1996** | 0.206 | 0.2536 | 0.204 | 0.29 |
| 15 | 20 | Constant | 0.218 | 0.249 | **0.144** | 0.2 | 0.261 | 0.165 | **0.124** | 0.278 |
| 15 | 50 | Constant | 0.1236 | 0.1472 | 0.0996 | **0.0972** | 0.112 | 0.124 | **0.0932** | 0.238 |

2. Obtain a ranking $z_{n+1}$ for an unlabeled test observation for which dissimilarities with the training observations are known. This test ranking should satisfy the property that $|z_i - z_{n+1}| \approx D_{i,n+1}$. Also, we want $z_i > z_{n+1}$ if $y_i > y_{n+1}$ and $z_i < z_{n+1}$ if $y_i < y_{n+1}$, although these properties cannot be verified since $y_{n+1}$ is unknown.

These tasks are easily accomplished using the methods proposed in the previous sections. We solve (3) with $S = 1$:

$$\underset{z_1,\ldots,z_n\in\mathbb{R}}{\text{minimize}} \left\{ \frac{1}{2}(1-\alpha)\sum_{i=1}^{n}\sum_{j=1}^{n}(D_{ij} - |z_i - z_j|)^2 + \alpha \sum_{i:y_i=1, 1\le i\le n} \sum_{j:y_j=2, 1\le j\le n} (D_{ij} - (z_j - z_i))^2 \right\}. \tag{21}$$

The resulting configuration points can be thought of as a bipartite ranking for the $n$ training observations. To determine the ranking and class label of an unlabeled test observation for which $D_{i,n+1}$ is known for $i = 1, \ldots, n$, we solve (9) with $S = 1$:

$$\underset{z_{n+1}\in\mathbb{R}, y_{n+1}\in\{1,2\}}{\text{minimize}} \left\{ \frac{1}{2}(1-\alpha)\sum_{i=1}^{n+1}\sum_{j=1}^{n+1}(D_{ij} - |z_i - z_j|)^2 + \alpha \sum_{i:y_i=1, 1\le i\le n+1} \sum_{j:y_j=2, 1\le j\le n+1} (D_{ij} - (z_{js} - z_{is}))^2 \right\}. \tag{22}$$

### 4.3. A simulation study

In this section, we explore the performance of SMDS with $S = 1$ as a classification method, and as a method for bipartite ranking. Test set classification error is used to evaluate the performance of SMDS as a classification method, and test set bipartite ranking error is used to evaluate its performance for bipartite ranking. We simulated data under three models, two of which were described in Section 2.3. In the third, there are again $n$ observations and $p$ features.

- For the *linear model* (so called because there is a linear trend in the observations, as a function of the observation index), the $i$th observation is generated as follows:

$$\mathbf{x}_i \sim N\left(\frac{3\mathbf{i}}{\mathbf{n}}, \mathbf{I}_p\right), \tag{23}$$

where $\frac{3\mathbf{i}}{\mathbf{n}}$ is a $p$-vector with elements $\frac{3i}{n}$. The first $\frac{n}{2}$ observations are in class 1, and the remaining $\frac{n}{2}$ observations are in class 2.

We expect SMDS to perform very well in the two-sided model, for which the decision boundary is highly non-linear and so methods that use a linear decision boundary will fail. We also expect SMDS to perform well in the linear model, in which there is a natural ranking of the observations as assumed by SMDS. On the other hand, methods that assume a linear decision boundary will exhibit their best performance under the constant model, and so we expect that SMDS may not perform well relative to some other methods under this simpler setting.

We generated training and test sets of equal sizes. All methods were trained on the training set, and performance was evaluated on the test set. $L_1$-penalized logistic regression is described in Park and Hastie (2007). The support vector machine (SVM), LDA, and $k$-nearest neighbors (KNN) are discussed in Hastie et al. (2009). The method of Cox and Ferry (1993) was performed using $S = 1$ in the MDS step. All tuning parameters were selected using cross-validation on the training data. Average test set classification errors and bipartite ranking errors (over 50 training/test sets) are reported in Tables 1 and 2. SMDS tends to outperform the other methods when the model is two-sided or linear. Even in the constant model, which is designed for methods that use a linear decision boundary, SMDS is pretty competitive with the other methods.

**Table 2**
For a number of values of $n$ (the number of observations) and $p$ (the number of features), the bipartite ranking errors of $L_1$-penalized logistic regression, SMDS with $\alpha = 1$, SMDS with $\alpha$ chosen by cross-validation, the method of Cox and Ferry (1993) (CF), SVM with a linear kernel (LK), SVM with a polynomial kernel of degree 3 (PK), and LDA are compared on the test data. Tuning parameters were chosen by cross-validation on the training data. Three models were used: the *two-sided*, *linear*, and *constant* models described in the text. Results are averaged over 50 simulated data sets. Within each row, the two methods with the lowest error rates are shown in bold.

| $p$ | $n$ | Model | $L_1$ logistic | SMDS $\alpha = 1$ | SMDS w/CV | CF | SVM LK | SVM PK | LDA |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 20 | Two-sided | 0.502 | **0.2454** | **0.2788** | 0.5252 | 0.4341 | 0.4214 | 0.4956 |
| 5 | 50 | Two-sided | 0.5012 | **0.193** | **0.2069** | 0.4968 | 0.4478 | 0.4423 | 0.5004 |
| 15 | 20 | Two-sided | 0.512 | **0.1162** | **0.1462** | 0.4882 | 0.4145 | 0.4179 | 0.4836 |
| 15 | 50 | Two-sided | 0.5016 | **0.0646** | **0.0646** | 0.503 | 0.4525 | 0.4444 | 0.5164 |
| 5 | 20 | Linear | 0.212 | **0.058** | **0.0736** | 0.2756 | 0.1084 | 0.1121 | 0.1032 |
| 5 | 50 | Linear | 0.1984 | 0.0565 | **0.0538** | 0.2823 | **0.0563** | 0.0635 | 0.0594 |
| 15 | 20 | Linear | 0.164 | **0.0296** | **0.027** | 0.495 | 0.117 | 0.136 | 0.2118 |
| 15 | 50 | Linear | 0.11 | 0.0173 | **0.0148** | 0.4604 | **0.0169** | 0.0433 | 0.0484 |
| 5 | 20 | Constant | 0.312 | 0.203 | **0.1904** | 0.2576 | 0.2028 | 0.2134 | **0.1586** |
| 5 | 50 | Constant | 0.2812 | 0.1593 | 0.1432 | 0.2912 | **0.1207** | 0.1633 | **0.1232** |
| 15 | 20 | Constant | 0.218 | 0.1306 | **0.048** | 0.374 | 0.135 | **0.1088** | 0.1943 |
| 15 | 50 | Constant | 0.1236 | 0.0669 | 0.0456 | 0.4206 | **0.0271** | 0.1029 | **0.0416** |

## 5. Extension of SMDS to multiple classes

In this section, we extend SMDS to the case where $K > 2$ classes are present in the training data using a *one versus all* approach, as described for instance in Rifkin and Klautau (2004). Let $i = 1, \ldots, n$ be the indices of the training observations, and let $n+1$ be the index of the test observation. Let $y_i \in \{1, \ldots, K\}$ denote the class label for observation $i$; $y_{n+1}$ is unknown. Then, *multi-class SMDS* is a straightforward extension of two-class SMDS. Rather than an $S$-dimensional configuration point for each observation, we obtain a $S \times K$-dimensional matrix, where column $k$ is composed of the $S$-vector obtained by performing SMDS with 2 classes as follows: class 1 is the set of observations for which $y_i = k$ and class 2 is the set of observations for which $y_i \neq k$. In greater detail, we proceed as follows to obtain training configuration points for observations $1, \ldots, n$ and a test configuration point and predicted class label for observation $n + 1$:

1. For $k = 1, \ldots, K$:
   (a) Let class 1 consist of the observations $\{i : y_i = k, 1 \leq i \leq n\}$ and let class 2 consist of the observations $\{i : y_i \neq k, 1 \leq i \leq n\}$.
   (b) Compute training configuration points, the test configuration point, and a cutpoint $c$ as specified in Sections 2.1 and 3.1.
   (c) Compute
   $$a_k = \min_{\mathbf{z}_{n+1}} h_1(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) - \min_{\mathbf{z}_{n+1}} h_2(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) - c \tag{24}$$
   where $h_1$ and $h_2$ are as specified in (11) and (12).
2. Assign the test observation to the class for which $a_k$ is smallest.
3. For each training observation, arrange the $K$ training configuration points into a $S \times K$ matrix.
4. For the test observation, arrange the $K$ test configuration points into a $S \times K$ matrix.

Steps 3 and 4 of the above procedure are not necessary for classifying the test observation, but may provide a useful tool for data visualization.

Let us consider a specific example. Suppose that we generate data consisting of $n$ observations and $p$ features, where the $i$th observation is generated as follows:

$$\mathbf{x}_i \sim \begin{cases} N(\mathbf{0}, \mathbf{I}_p) & \text{if observation } i \text{ is in class 1,} \\ N(\mathbf{1}, \mathbf{I}_p) & \text{if observation } i \text{ is in class 2,} \\ N(-\mathbf{1}, \mathbf{I}_p) & \text{if observation } i \text{ is in class 3.} \end{cases} \tag{25}$$

Here, $\mathbf{0}$ and $\mathbf{1}$ are $p$-vectors of 0's and 1's, respectively. We let $p = 30$, with $n = 20$ observations per class in the training set and an equally-sized test set. Consider two ways to visualize the test data.

1. Perform MDS on the test data with 3-dimensional configuration points.
2. Train multi-class SMDS with $S = 1$ on the training data and then compute the triple of configuration points for each test observation.

The resulting plots are shown in Fig. 8. The three classes show some separation along the first MDS component. However, multi-class SMDS results in very clear separation between the three classes. We point out that the multi-class SMDS plots in Fig. 8 were made without knowledge of the test set class labels: test labels were only used to indicate the class of each observation in the figure.

## 6. Discussion

MDS is a classical tool for obtaining a set of $n$ configuration points $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^S$ such that $\|\mathbf{z}_i - \mathbf{z}_j\|_2 \approx D_{ij}$, where $D_{ij}$ represents the pairwise dissimilarity between observations $i$ and $j$. In this paper, we have extended MDS in order to

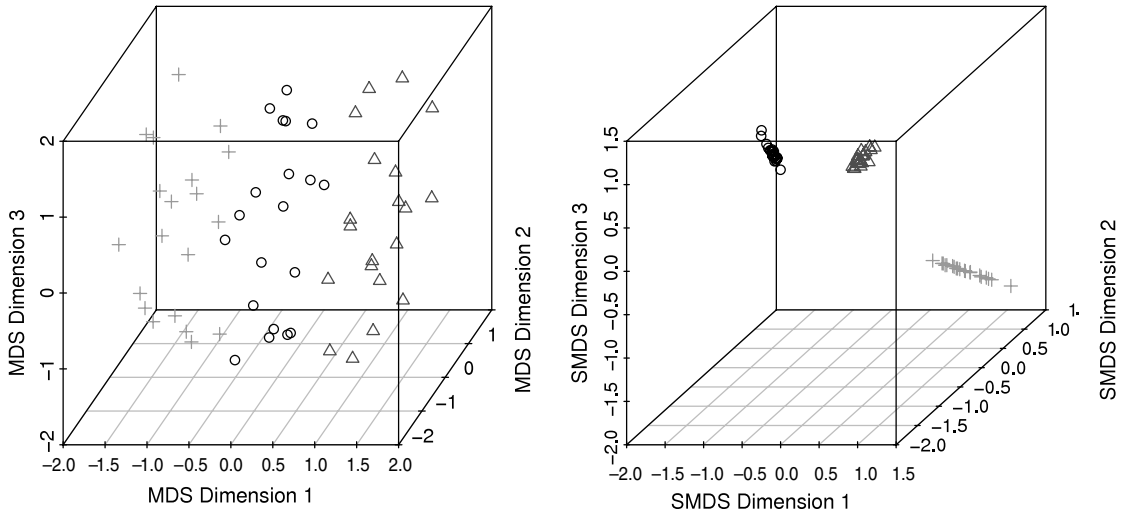


**Fig. 8.** Each test observation belongs to class 1 (circles), class 2 (triangles), or class 3 (crosses). *Left*: MDS was performed with $S = 3$. *Right*: Multi-class SMDS with $S = 1$ was trained on the training set, and the resulting triple of configuration points was computed for each test observation. The class $k$ observations tend to have the smallest values in the $k$th configuration point.

incorporate an outcome measurement $y_i$ for observation $i$ in such a way that $z_{is} > z_{js}$ tends to occur when $y_i > y_j$. In addition to providing a tool for data visualization, our proposal naturally leads to a classification method that can also be interpreted as a method for bipartite ranking. Though our proposal's classification performance is mixed, it shows promise especially in cases that are difficult for most standard classifiers, in which the optimal decision boundary is highly nonlinear.

We have suggested a simple majorization approach for solving the SMDS problem. Improvements to this algorithm could lead to configuration points yielding a smaller value of the objective; we leave this as a topic for future work. An R package implementing SMDS will be made available on CRAN, http://cran.r-project.org/.

## Acknowledgements

We thank an associate editor and two referees for pointing us to Cox and Ferry (1993) and for providing a number of useful suggestions. We thank Jacob Bien for helpful conversations and for sharing R code with us, and we thank Trevor Hastie for providing FORTRAN code for computing tangent distance. Robert Tibshirani was supported by National Science Foundation Grant DMS-9971405 and National Institutes of Health Contract N01-HV-28183.

## Appendix

We refer the interested reader to Borg and Groenen (2005) for a helpful reference on majorization algorithms in general and on the MDS majorization algorithm in particular. Here, we discuss the majorization algorithms used for SMDS.

*A majorization algorithm for Eq. (3)*

Consider the problem of minimizing (3) with respect to $\mathbf{z}_k$. We take a majorization approach. We consider two cases: $y_k = 1$ and $y_k = 2$.

*Case* 1: $y_k = 1$. Minimizing (3) with respect to $\mathbf{z}_k$ is equivalent in this case to minimizing

$$(1-\alpha)\sum_{j\neq k}\|\mathbf{z}_k-\mathbf{z}_j\|^2 - 2(1-\alpha)\sum_{j\neq k}D_{kj}\|\mathbf{z}_k-\mathbf{z}_j\|_2 + \alpha\sum_{j:y_j=2}\|\mathbf{z}_k-\mathbf{z}_j\|^2 - 2\frac{\alpha}{\sqrt{S}}\sum_{j:y_j=2}D_{kj}\sum_{s=1}^{S}(z_{js}-z_{ks}). \tag{26}$$

And a majorizing function for (26) is

$$(1-\alpha)\sum_{j\neq k}\|\mathbf{z}_k-\mathbf{z}_j\|^2 - 2(1-\alpha)\sum_{j\neq k}D_{kj}\frac{\sum_{s=1}^{S}(z_{ks}-z_{js})(\tilde{z}_{ks}-z_{js})}{\|\tilde{\mathbf{z}}_k-\mathbf{z}_j\|_2}$$
$$+\alpha\sum_{j:y_j=2}\|\mathbf{z}_k-\mathbf{z}_j\|^2 - 2\frac{\alpha}{\sqrt{S}}\sum_{j:y_j=2}D_{kj}\sum_{s=1}^{S}(z_{js}-z_{ks}) \tag{27}$$

by the Cauchy–Schwarz inequality.

Then, the derivative of (27) with respect to $z_{ks}$ takes the form

$$2(1-\alpha)\sum_{j\neq k}(z_{ks}-z_{js}) - 2(1-\alpha)\sum_{j\neq k} D_{jk}\frac{\tilde{z}_{ks}-z_{js}}{\|\tilde{\mathbf{z}}_k-\mathbf{z}_j\|_2} + 2\alpha\sum_{j:y_j=2}(z_{ks}-z_{js}) + \frac{2\alpha}{\sqrt{S}}\sum_{j:y_j=2} D_{kj}. \tag{28}$$

Setting the derivative equal to zero and solving for $z_{ks}$, we see that the update is

$$z_{ks} \leftarrow \frac{1}{(n-1)(1-\alpha)+n_2\alpha}\left[(1-\alpha)\sum_{j\neq k} z_{js} + (1-\alpha)\sum_{j\neq k} D_{jk}\frac{\tilde{z}_{ks}-z_{js}}{\|\tilde{\mathbf{z}}_k-\mathbf{z}_j\|_2} + \alpha\sum_{j:y_j=2} z_{js} - \frac{\alpha}{\sqrt{S}}\sum_{j:y_j=2} D_{kj}\right]. \tag{29}$$

*Case* 2: $y_k = 2$. Repeating the previous argument, the update is

$$z_{ks} \leftarrow \frac{1}{(n-1)(1-\alpha)+n_1\alpha}\left[(1-\alpha)\sum_{j\neq k} z_{js} + (1-\alpha)\sum_{j\neq k} D_{jk}\frac{\tilde{z}_{ks}-z_{js}}{\|\tilde{\mathbf{z}}_k-\mathbf{z}_j\|_2} + \alpha\sum_{j:y_j=1} z_{js} + \frac{\alpha}{\sqrt{S}}\sum_{j:y_j=1} D_{kj}\right]. \tag{30}$$

Combining these two cases, our majorization approach for solving (3) is as follows:

1. Initialize the configuration points $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^S$.
2. For $k = 1, \ldots, n, 1, \ldots, n, \ldots$:
   (a) Let $\tilde{\mathbf{z}}_k$ denote the current estimate of $\mathbf{z}_k$.
   (b) If $y_k = 1$, then update $\mathbf{z}_k$ using (29). If $y_k = 2$, then update $\mathbf{z}_k$ using (30).

Observe that each iteration of this algorithm decreases the objective (3) because each iteration decreases the majorizing function (27). As is done with MDS, we recommend running the algorithm using several random starts and choosing the configuration that results in the lowest value of the objective (3).

*A majorization algorithm for Eq. (9)*

We now consider solving (9). The two cases $\hat{y}_{n+1} = 1$ and $\hat{y}_{n+1} = 2$ are treated separately.

*Case* 1: $\hat{y}_{n+1} = 1$. We must minimize $h_1(\mathbf{D}_{n+1}, \mathbf{z}_{n+1})$ (11) with respect to $\mathbf{z}_{n+1}$: that is, one must find $\mathbf{z}_{n+1}$ that minimizes

$$\begin{aligned} &-\frac{2\alpha}{\sqrt{S}}\sum_{i:y_i=2,1\leq i\leq n}\sum_{s=1}^{S} D_{i,n+1}(z_{is}-z_{n+1,s}) + \alpha\sum_{i:y_i=2,1\leq i\leq n}\|\mathbf{z}_i-\mathbf{z}_{n+1}\|^2 \\ &+(1-\alpha)\sum_{i=1}^{n}\|\mathbf{z}_i-\mathbf{z}_{n+1}\|^2 - 2(1-\alpha)\sum_{i=1}^{n} D_{i,n+1}\|\mathbf{z}_i-\mathbf{z}_{n+1}\|_2. \end{aligned} \tag{31}$$

Now (31) is less than or equal to the majorizing function

$$\begin{aligned} &-\frac{2\alpha}{\sqrt{S}}\sum_{i:y_i=2,1\leq i\leq n}\sum_{s=1}^{S} D_{i,n+1}(z_{is}-z_{n+1,s}) + \alpha\sum_{i:y_i=2,1\leq i\leq n}\|\mathbf{z}_i-\mathbf{z}_{n+1}\|^2 \\ &+(1-\alpha)\sum_{i=1}^{n}\|\mathbf{z}_i-\mathbf{z}_{n+1}\|^2 - 2(1-\alpha)\sum_{i=1}^{n} D_{i,n+1}\frac{\sum_{s=1}^{S}(z_{is}-z_{n+1,s})(z_{is}-\tilde{z}_{n+1,s})}{\|\mathbf{z}_i-\tilde{\mathbf{z}}_{n+1}\|_2}. \end{aligned} \tag{32}$$

By differentiating (32) and setting it to zero, we arrive at the update

$$\begin{aligned} \mathbf{z}_{n+1} \leftarrow \frac{1}{n(1-\alpha)+n_2\alpha}\Bigg(&-\frac{\alpha}{\sqrt{S}}\sum_{i:y_i=2,1\leq i\leq n} D_{i,n+1} + \alpha\sum_{i:y_i=2,1\leq i\leq n} z_{is} \\ &+ (1-\alpha)\sum_{i=1}^{n} z_{is} - (1-\alpha)\sum_{i=1}^{n} D_{i,n+1}\frac{z_{is}-\tilde{z}_{n+1,s}}{\|\mathbf{z}_i-\tilde{\mathbf{z}}_{n+1}\|_2}\Bigg). \end{aligned} \tag{33}$$

Therefore, the following algorithm can be used to solve (9) if $\hat{y}_{n+1} = 1$, or equivalently to minimize (11):

1. Initialize the configuration point $\mathbf{z}_{n+1} \in \mathbb{R}^S$.
2. Iterate:
   (a) Let $\tilde{\mathbf{z}}_{n+1}$ denote the current estimate of $\mathbf{z}_{n+1}$.
   (b) Update $\mathbf{z}_{n+1}$ using (33).

*Case* 2: $\hat{y}_{n+1} = 2$. We must minimize $h_2(\mathbf{D}_{n+1}, \mathbf{z}_{n+1})$ (12) with respect to $\mathbf{z}_{n+1}$. Repeating the argument in Case 1, we arrive at the update

$$\mathbf{z}_{n+1} \leftarrow \frac{1}{n(1-\alpha)+n_1\alpha}\left(\frac{\alpha}{\sqrt{S}}\sum_{i:y_i=1,1\le i\le n} D_{i,n+1} + \alpha \sum_{i:y_i=1,1\le i\le n} z_{is} + (1-\alpha)\sum_{i=1}^{n} z_{is} - (1-\alpha)\sum_{i=1}^{n} D_{i,n+1}\frac{z_{is}-\tilde{z}_{n+1,s}}{\|\mathbf{z}_i - \tilde{\mathbf{z}}_{n+1}\|_2}\right). \tag{34}$$

An analogous algorithm to that of Case 1 can be obtained using the update (34).

As described in Section 3.1, if $\min_{\mathbf{z}_{n+1}} h_1(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) - \min_{\mathbf{z}_{n+1}} h_2(\mathbf{D}_{n+1}, \mathbf{z}_{n+1}) < c$ then $\hat{y}_{n+1} = 1$; otherwise, $\hat{y}_{n+1} = 2$. The test configuration point is then given by (14). Here, $c$ equals the $\frac{n_1}{n}$ quantile of $\{\min_{\mathbf{z}_j} h_1(\mathbf{D}_j, \mathbf{z}_j) - \min_{\mathbf{z}_j} h_2(\mathbf{D}_j, \mathbf{z}_j)\}_{j=1}^{n}$.

## References


Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., Roth, D., 2005. Generalization bounds for the area under the ROC curve. Journal of Machine Learning Research 6, 393–425.

Agarwal, S., Niyogi, P., 2009. Generalization bounds for ranking algorithms via algorithmic stability. Journal of Machine Learning Research 10, 441–474.

Borg, I., Groenen, P., 2005. Modern Multidimensional Scaling. Springer, New York.

Buja, A., Swayne, D., Littman, M., Dean, N., Hofmann, H., Chen, L., 2008. Data visualization with multidimensional scaling. Journal of Computational and Graphical Statistics 17, 444–472.

Cohen, W., Schapire, R., Singer, Y., 1998. Learning to order things. In: NIPS 1997. vol. 10. pp. 451–457.

Cox, T., Cox, M., 1994. Multidimensional Scaling. Chapman and Hall, London.

Cox, T., Ferry, G., 1993. Discriminant analysis using non-metric multidimensional scaling. Pattern Recognition 26, 145–153.

Crammer, K., Singer, Y., 2001. Pranking with ranking. In: NIPS 2001. vol. 14. pp. 641–647.

Freund, Y., Iyer, R., Schapire, R., Singer, Y., 2003. An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research 4, 933–969.

Green, P., 1989. Multidimensional Scaling: Concepts and Applications. Allyn and Bacon, Needham Heights, Mass.

Hastie, T., Simard, P., 1998. Models and metrics for handwritten digit recognition. Statistical Science 13, 54–65.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning; Data Mining, Inference and Prediction. Springer Verlag, New York.

Kruskal, J., 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29, 1–27.

Park, M.Y., Hastie, T., 2007. An $L_1$ regularization path algorithm for generalized linear models. Journal of the Royal Statistical Society. Series B 69 (4), 659–677.

Rifkin, R., Klautau, A., 2004. In defense of one-vs-all classification. Journal of Machine Learning Research 5, 101–141.

Shepard, R., 1962. The analysis of proximities: multidimensional scaling with an unknown distance function. Psychometrika 27, 125–140.

Simard, P., Cun, Y.L., Denker, J., 1993. Efficient pattern recognition using a new transformation distance. In: Advances in Neural Information Processing Systems. Morgan Kaufman, San Mateo, CA, pp. 50–58.

Singh, D., Febbo, P.G., Ross, K., Jackson, D.G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A.A., D'Amico, A.V., Richie, J.P., Lander, E.S., Loda, M., Kantoff, P.W., Golub, T.R., Sellers, W.R., 2002. Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1, 203–209.