Int. J. Appl. Math. Comput. Sci., 2013, Vol. 23, No. 2, 463–471
DOI: 10.2478/amcs-2013-0035

LINEAR DISCRIMINANT ANALYSIS WITH A GENERALIZATION OF THE MOORE–PENROSE PSEUDOINVERSE

TOMASZ GÓRECKI [*], MACIEJ ŁUCZAK [**]

[*] Faculty of Mathematics and Computer Science
Adam Mickiewicz University, Umultowska 87, 61-614 Poznań, Poland
e-mail: tomasz.gorecki@amu.edu.pl

[**] Faculty of Civil Engineering, Environmental and Geodetic Sciences
Koszalin University of Technology, Śniadeckich 2, 75-453 Koszalin, Poland
e-mail: mluczak@wilsig.tu.koszalin.pl

The Linear Discriminant Analysis (LDA) technique is an important and well-developed area of classification, and to date many linear (and also nonlinear) discrimination methods have been put forward. A complication in applying LDA to real data occurs when the number of features exceeds that of observations. In this case, the covariance estimates do not have full rank, and thus cannot be inverted. There are a number of ways to deal with this problem. In this paper, we propose improving LDA in this area, and we present a new approach which uses a generalization of the Moore–Penrose pseudoinverse to remove this weakness. Our new approach, in addition to managing the problem of inverting the covariance matrix, significantly improves the quality of classification, also on data sets where we can invert the covariance matrix. Experimental results on various data sets demonstrate that our improvements to LDA are efficient and our approach outperforms LDA.

Keywords: linear discriminant analysis, Moore–Penrose pseudoinverse, machine learning.

1. Introduction

Linear discrimination is widely used in practice (e.g., face recognition (Song *et al.*, 2007), medicine (Kwak *et al.*, 2002), chemometrics (Cozzolino *et al.*, 2002), etc.). Additionally, LDA is a supervised dimension-reduction method (a special case of canonical correlation analysis), which is important in data mining and machine learning (Shin and Park, 2011). LDA easily handles the case where the within-class frequencies are unequal and their performances have been examined on randomly generated test data. Although relying on heavy assumptions which are not true in many applications, LDA has been proved to be effective (Lim *et al.*, 2000). This is mainly due to the fact that a simple, linear model is more robust against noise, and most likely will not overfit. Also, the linear discriminant function is a linear combination of the measured variables, being easy to interpret. Classical LDA involves a sample covariance matrix which is required to be nonsingular. However, in many applications such as text mining, microarray data classification and face recognition, this matrix can be singular, since the data points are in a very high-dimensional space and the sample size does not exceed this dimension. This is known as the singularity (undersampled) problem or the Small Sample Size (SSS) problem.

When the class sample sizes are small compared with the dimension of the measurement space d, the covariance matrix estimates, especially, become highly variable. Moreover, when the sample size is less than d, not all of their parameters are even identifiable. In this case, the covariance estimates do not have full rank, and thus cannot be inverted. There are a number of ways to deal with this:

- Employ a regularization method. These techniques have been highly successful in the solution of ill- and poorly-posed inverse problems (Titterington, 1985; Friedman, 1989; Kuo and Landgrebe, 2002). However, the computational complexity is very high.

- Try to obtain more reliable estimates of the eigenvalues by correcting the eigenvalue distortion in the sample covariance matrix. Stein *et al.*

(1972), Olkin and Selliah (1975), Dey and Srinivasan (1985), Hong and Yang (1991) as well as Bensmail and Celeux (1996) have studied this approach. Unfortunately, they nearly all require that $\hat{\Sigma}$ (estimate of the covariance matrix) be nonsingular.

- Use sparse covariance selection methods (Dempster, 1972; d'Aspremont *et al.*, 2008).
- Use gradient LDA (Sharma and Paliwal, 2008).
- Subspace method. Project the original samples to a lower dimensional space to make the resulting within-class scatter matrix full-rank. The most widely used subspace method performs PCA firstly to reduce the dimension of the samples (Swets and Weng, 1996). Another commonly known method of this type is called direct LDA (Yu and Yang, 2001).
- Null space method (Chen *et al.*, 2000). All the samples are firstly projected onto the null space of the within-class scatter matrix, where the within-class scatter is zero, and then the optimal discriminant vectors of LDA are those that can maximize the between-class scatter. PCA is used to yield them. Like the regularization method, the computational complexity is also very high.
- Use a pseudoinverse instead of the usual matrix inverse (Tian *et al.*, 1988; Duda *et al.*, 2001).

Problems concerning the small sample size and pseudoinverse appear in the most recent works (Piegat and Landowski, 2012; Röbenack and Reinschke, 2011).

We propose an extension of the last approach. Classically, the Moore–Penrose (MP) inverse is used to find the inverse of a sample covariance matrix; we try to find a specific generalization of the Moore–Penrose inverse. We construct a parametrical family of generalized MP inverses and use it in the linear discrimination method. Then we choose the models with the lowest cross-validation (leave-one-out) error rates and we combine them by a mean rule. With this approach we obtain, in addition to opportunities to work with any data (size), a substantial decrease in the classification error rate compared to LDA.

In our paper we first present the main ideas of LDA in Section 2. In the same section we describe generalized inverses of matrices. At the end of this section we explain our idea of extended LDA. In our paper the performances of the described methods are compared and the error of classification is analyzed. Many real (32) and artificial (6) data sets are used. The methods and data sets used are described in Section 3. The results of the research are explained with charts, where differences between the classifiers are shown. Section 4 contains the results of our experiments on the described data sets, as well as statistical analysis of the results. We conclude with a discussion in Section 5.

2. Methods

Suppose that a training sample has been collected by sampling from a population P consisting of K unordered subpopulations, classes or groups, which we denote by $G_1, \ldots, G_K$. Each item in P is assumed to be a member of one (and only one) of those classes and an error is incurred if it is assigned to a different one. Measurements on a sample of items are to be used to help assign future unclassified items to one of the designated classes. The i-th observation is a pair denoted by $(\boldsymbol{x}_i, y_i)$, where $\boldsymbol{x}_i$ is a d-dimensional feature vector and y_i is the label for recording class membership. The corresponding pair for an unclassified observation is denoted by $(\boldsymbol{x}, y)$. In this case $\boldsymbol{x}$ is observed but the class label y is unobserved. The goal of classification is to construct a classification rule for predicting the membership of an unclassified feature vector $\boldsymbol{x} \in P$. An automated classifier can be viewed as a method of estimating the posterior probability of membership in G_k. For a given $\boldsymbol{x}$, a reasonable classification strategy is to assign $\boldsymbol{x}$ to that class with the highest posterior probability. This strategy is called the Bayes rule classifier. We denote the posterior probability of membership in G_k by

$$p_k(\boldsymbol{x}) = P(y = k|\boldsymbol{x}). \tag{1}$$

Let π_k be the prior probabilities that a randomly selected observation belongs to class G_k. Suppose also that the conditional multivariate probability density for the k-th class is $f_k(\boldsymbol{x})$. We note that there is no requirement that the densities be continuous; they could be discrete or be finite mixture distributions or even have singular covariance matrices. Now Bayes' theorem yields the posterior probability

$$p_k(\boldsymbol{x}) = \frac{f_k(\boldsymbol{x})\pi_k}{\sum_{i=1}^{K} f_i(\boldsymbol{x})\pi_i} \tag{2}$$

that the observed $\boldsymbol{x}$ belongs to the class G_k. If the maximum does not uniquely define a class assignment for a given $\boldsymbol{x}$, then use a random assignment to break the tie between the appropriate classes.

2.1. Linear discriminant analysis.

In practice it is unwise to use Bayes' rule directly, because to obtain $f_k(\boldsymbol{x})$ we need so much data to get the relative frequencies of all groups for each measurement. It is more practical to assume the distribution and obtain the probability theoretically. So we now make the Bayes rule classifier more specific by the assumption that all multivariate probability densities are multivariate Gaussian (normal), having arbitrary mean vectors and a common covariance matrix. That is, we take f_k to be an $N_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ density,

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \times \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right\}. \quad (3)$$

Under the above assumptions we can write a linear Bayesian classifier as (assign an object $\boldsymbol{x}$ to a group G_k that yields maximum $\delta_k(\boldsymbol{x})$)

$$d_B(\boldsymbol{x}) = \arg\max_k \delta_k(\boldsymbol{x}), \quad (4)$$

where

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln \pi_k \quad (5)$$

is a linear discriminant function. Careful inspection shows that the second term ($\boldsymbol{\mu}_k'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$) is actually the Mahalanobis distance, which is used to measure the dissimilarity between several groups.

In practice, the class means and covariances are not known. They can, however, be estimated from the training set. Usually the maximum likelihood (plug-in) estimate may be used in place of the exact value in the above equations. Although the estimates of the covariance may be considered optimal in some sense, this does not mean that the resulting discriminant obtained by substituting these values is optimal in any sense, even if the assumption of normally distributed classes is correct (Anderson, 1984). Also, any sensible Bayesian rule will not lead to this approach, except either asymptotically or under very restrictive conditions (Enis and Geisser, 1986). Additionally, we have to estimate *a priori* probabilities. These are usually estimated simply by the empirical frequencies of observations in the training set.

2.2. Algorithm.

In linear discriminant analysis, the inverse $\boldsymbol{A}^{-1}$ or the Moore–Penrose inverse $\boldsymbol{A}^{+}$ is used to compute an inverse of the covariance matrix (see Eqn. (3)). The main idea of this paper is to adopt another generalized inverse.

We consider a general (real) $m \times n$ matrix $\boldsymbol{A}$ rank whose may be less than $\min(m, n)$. If $\boldsymbol{M}, \boldsymbol{N}$ are positive definite matrices, and there exist factorizations $\hat{\boldsymbol{N}}'\hat{\boldsymbol{N}} = \boldsymbol{N}$, $\hat{\boldsymbol{M}}'\hat{\boldsymbol{M}} = \boldsymbol{M}$, then

$$\boldsymbol{A}^{+}_{\boldsymbol{MN}} = \hat{\boldsymbol{N}}^{-1}(\hat{\boldsymbol{M}}\boldsymbol{A}\hat{\boldsymbol{N}}^{-1})^{+}\hat{\boldsymbol{M}} \quad (6)$$

satisfies the condition

$$\|\boldsymbol{A}^{+}_{\boldsymbol{MN}}\boldsymbol{y}\|_n \leq \|\boldsymbol{x}\|_n,$$
$$\forall\, \boldsymbol{x} \in \{\boldsymbol{x}\colon \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_m \leq \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_m,\ \forall\, \boldsymbol{z} \in \mathbb{R}^n\},$$

where $\|\boldsymbol{x}\|_n = \sqrt{\boldsymbol{x}'\boldsymbol{N}\boldsymbol{x}}$ and $\|\boldsymbol{y}\|_m = \sqrt{\boldsymbol{y}'\boldsymbol{M}\boldsymbol{y}}$ are norms in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. $\boldsymbol{A}^{+}_{\boldsymbol{MN}}$ is referred to as the minimum $\boldsymbol{N}$-norm $\boldsymbol{M}$-least-squares g-inverse of $\boldsymbol{A}$. When $\boldsymbol{M}, \boldsymbol{N}$ are identity matrices, we use the notation $\boldsymbol{A}^{+}$ and call it the Moore–Penrose inverse (pseudoinverse). For a deeper survey and more details, we refer the readers to Rao and Mitra (1971).

If $\boldsymbol{M}$ is positive semi-definite, then $\|\boldsymbol{y}\|_m$ is a seminorm and the right-hand side of Eqn. (6) does not need to be a g-inverse. We denote this by $\boldsymbol{A}^{*}_{\boldsymbol{MN}}$ and $\boldsymbol{A}^{*}_{\boldsymbol{M}}$ if $\boldsymbol{N} = \boldsymbol{I}$.

We use $\boldsymbol{A}^{*}_{\boldsymbol{M}}$ with a special form of matrix $\boldsymbol{M}$. Precisely, we use Eqn. (6) with the assumptions

$$\hat{\boldsymbol{N}} = \boldsymbol{N} = \boldsymbol{I},\ \hat{\boldsymbol{M}} = \boldsymbol{M} = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_m \end{bmatrix}, \quad (7)$$

where $a_i = 0$ or 1 for $i = 1, \dots, m$. This leads to the seminorm

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}'\boldsymbol{M}\boldsymbol{x}} = \sqrt{x_{j_1}^2 + x_{j_2}^2 + \dots + x_{j_k}^2}, \quad 1 \leq k \leq m,$$

for $\boldsymbol{x} = (x_1, x_2, \dots, x_m) \in \mathbb{R}^m$ ($a_i x_i^2 = 0$ for $a_i = 0$, $j_s = i$ for $a_i = 1$). Then Eqn. (6) assumes the form

$$\boldsymbol{A}^{*}_{\boldsymbol{M}} = (\boldsymbol{M}\boldsymbol{A})^{+}\boldsymbol{M}. \quad (8)$$

Thus we can use $\boldsymbol{\Sigma}^{*}_{\boldsymbol{M}}$ instead of $\boldsymbol{\Sigma}^{-1}$ to compute the inverse of the covariance matrix $\boldsymbol{\Sigma}$ (see Eqn. (3)). Note that we do not have to compute the determinant of the covariance matrix $\boldsymbol{\Sigma}$ to obtain posterior probabilities (see Eqn. (2)).

We only take ones and zeros on the diagonal of the matrix $\boldsymbol{M}$ because it can be proved (see Appendix) that $\boldsymbol{A}^{*}_{\boldsymbol{M}}$ depends only on whether or not the coefficients a_i are zeros.

We try two algorithms for choosing ones and zeros on the diagonal of the matrix $\boldsymbol{M}$. In the first one (ALG1) we pass through all the combinations of ones and zeros on the diagonal of matrix $\boldsymbol{M}$. In the second (ALG2), we take only diagonals with at most one zero. Then, in both the cases, we choose the linear discriminant models with the lowest (different) $\kappa = 1, 2, 3$ cross-validation (leave-one-out) error rates. There can be many different models with the same cross-validation error value. The models are combined by the mean combiner, i.e., posterior probabilities are computed by

$$p_k = \frac{1}{n}(p_{1k} + p_{2k} + \dots + p_{nk}),$$

where p_{ik} is the posterior probability of the i-th model. Thus we consider six subalgorithms: ALG1-1, ALG1-2, ALG1-3, ALG2-1, ALG2-2, ALG2-3, where the second number is from $\kappa = 1, 2, 3$ (see above).

3. Computational experiments

3.1. Real data sets. We performed experiments on 20 data sets with less than 15 features and 12 data sets with more than 15 features. The data sets were chosen in such a way that they had different numbers of features of particular types and different numbers of examples, and there were some data sets with two-class distribution and some with more than two classes. In Table 1 the characteristics of the data sets are given, showing the variety of training set sizes, the number of classes and dimensionality.

The data sets *chemistry* and *irradiation* come from the work of Morrison (1990), *fish* from that of Hawkins and Rasmussen (1978), *football* from Gleim's (1984), *risk* from Dillon and Goldstein's (1984) and *turtles* from StatSoft (2007). The remaining data sets come from the UCI Machine Learning Repository (Frank and Asuncion, 2010). When necessary, we removed observations for which there were missing values.

For real data sets the classification errors were estimated by the leave-one-out and bootstrap methods. The former method was used to find "the best" (with the smallest error rates) diagonals of matrix $\boldsymbol{M}$. In the next step, we construct our models (with the mean combiner). For each of these models, we calculated the bootstrap classification error rate (1000 repetitions). We finally took as the error rate of our method the mean of these bootstrap error rates.

Table 1. Information on the data sets.

Name of the data set	Number of features	Number of classes	Number of instances
breastW	9	2	683
car	6	4	1728
chemistry	3	4	45
echo	6	2	108
fish	4	3	36
football	6	3	90
glass	9	6	214
golf	4	2	14
hayes	5	3	132
heart	13	2	270
heartC	13	5	297
heartH	10	5	261
heartS	10	5	105
iris	4	3	150
irradiation	3	4	45
liver	6	2	345
risk	2	3	87
thyroid	5	3	215
turtles	6	2	48
wine	13	3	178
hepatitis	16	2	137
ionosphere	33	2	351
kr-vs-kp	36	2	3196
libras	90	15	360
musk	166	2	476
parkinsons	22	2	195
sonar	60	2	208
spam	57	2	4601
spectf	44	2	267
statlog	19	7	2310
vote	16	2	300
wave	21	3	125

3.2. Artificial data sets. We also performed experiments with artificial data. We used data described by Friedman (1989). Each experiment consisted of one hundred replications of the following procedure. First $N = 40$ class identity labels were randomly drawn. Then, conditioned on each label, measurement vectors were drawn from the appropriate class distribution. The prior probability of each of the three classes was taken to be equal, so that the expected number of observations in each class was 13.3. However, the actual number in any particular replication was itself a random variable. Each such training data set was used to construct the classifier. An additional (test) data set of size $N = 100$ was then randomly generated from the same population and classified with the rule derived from the training set. The absolute test error is the average test misclassification risk over the one hundred replications for each of the classification rule. Friedman created six different data sets: F1, F2, F3, F4, F5, F6. For each data set we used $d = 50$ features, so LDA is ill-posed.

In the computational process we used the PRTools 4.1.4 program (`http://www.prtools.org`). This is a Matlab (version R2009b) based toolbox for pattern recognition (van der Heijden *et al.*, 2004). In each procedure we used default parameters.

4. Results

4.1. Classification error rates. In ALG1 we take all possible models, i.e., for all combinations of ones and zeros in the diagonal of matrix $\boldsymbol{M}$, while in ALG2 we take diagonals with at most one zero. It is possible to take other numbers of ones on the diagonal: at least n ones, where $0 \leq n \leq d$ (d being the number of features). Then for $n = d$ it is an LDA method, for $n = d - 1$ it becomes ALG2, and for $n = 0$ it is ALG1. For other values of n we have different algorithms. If $n_2 < n_1$, then an algorithm for n_2 includes all models from an algorithm with n_1. Thus, the smaller n, the smaller the CV error rate of the algorithm. But this is not necessarily true of the bootstrap error rate. In Fig. 1 the two error rates depending on the value of n are shown. We can see that bootstrap error does not follow the CV error—for some values of n it does not increase. Because of this we consider only the cases $n = 0$ (ALG1) and $n = d - 1$ (ALG2) in the paper. Another reason is that ALG2 has to employ far fewer models ($d{+}1$) than ALG1 (2^d). For example, an algorithm with $n = d/2$ has only the number of models of an algorithm with $n = 0$ (ALG1). Thus it seems sensible to take into consideration only two boundary algorithms: ALG1 and ALG2. ALG1 is limited to data sets with smaller number of features because of an exponential computational complexity (2^d models, d—the number of features). However, ALG2 has to construct only $d + 1$ models, so it can be used to larger data sets. In practice, the main algorithm is ALG2, and with ALG1 we can only check how close the two algorithms are.

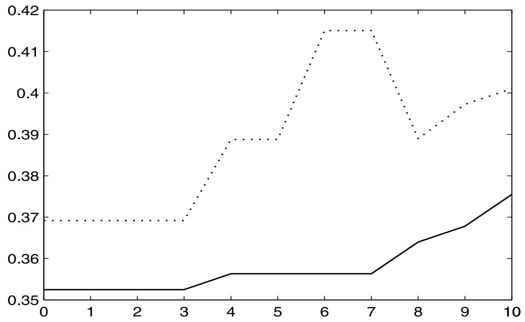

Fig. 1. Error rates depending on the value of n for the *heartH* data set ($d = 10$, $\kappa = 1$), where n means at least n ones on the diagonal of matrix $\boldsymbol{M}$ (— CV error rate, $\cdots$ bootstrap error rate).

In Fig. 2 we can observe the behavior of our algorithm. We see how three simple linear models (found by the algorithm, with a minimal CV error rate) are combined into a more sophisticated nonlinear model. The circled point is classified correctly by our algorithm. In this simple example we see that our method might improve the standard LDA one.

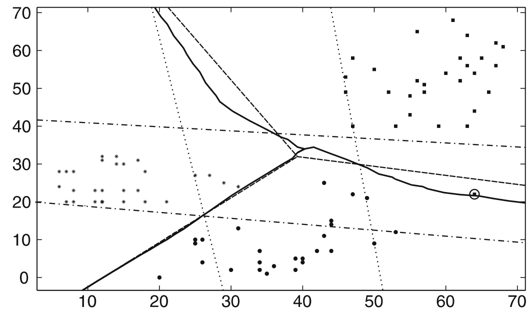

Fig. 2. Decision boundaries for the *risk* data set (— ALG1-3, $\cdots$ linear method for $\boldsymbol{M} = \left[\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right]$, $- \cdot -$ linear method for $\boldsymbol{M} = \left[\begin{smallmatrix}0 & 0\\ 0 & 1\end{smallmatrix}\right]$, $--$ LDA).

The main result is shown in Table 3. For data sets with more than 15 features we used only ALG2, because of the computational complexity of ALG1. All the algorithms improve the standard LDA method on almost all data sets. Generally, we can order the algorithms (in terms of mean error rate) as follows:

$$\text{ALG*-1} < \text{ALG*-2}, \qquad \text{ALG*-1} < \text{ALG*-3}.$$

Combining more models improves classification. For example, ALG1-3 is practically always better than LDA. The only exception is for the *risk* data set. This data set has only two features, so there are only four possible models and ALG1-3 combines almost all (three) of them. But sometimes ALG1-3 is the only algorithm better than the LDA method—this holds for the *heart* and *thyroid* data sets. ALG2 is worse than ALG1 but much faster; it has far fewer models to combine, especially for the subalgorithm ALG*-3. For data sets with a large number of features, the subalgorithms of ALG2 are more similar, and practically all improve on the LDA method.

In Table 2, results for artificial data sets with 50 features and 40 observations are shown. The covariance matrix is not invertible for these datasets, and the Moore–Penrose inverse is used in the standard LDA method. We can see that combining more models also improves classification. The subalgorithm ALG2-3 is better than LDA for all datasets.

Table 2. Test error rates on Friedman's data sets. In the columns: absolute test error rate for the LDA method (computed according the procedure described in Section 3.2) and relative test error rates for subalgorithms ALG2-1 to ALG2-3, computed by the formula (alg*-* − LDA)/LDA, where alg*-* is the absolute test error rate of ALG*-*. All the subalgorithms columns are followed by columns with the average number of linear models combined in the subalgorithms.

Data set	LDA	ALG2-1		ALG2-2		ALG2-3	
F1	42.98	-0.07	3.1	0.23	10.2	-0.88	21.3
F2	46.09	-1.35	2.8	-1.45	10.7	-1.48	22.6
F3	50.22	1.47	2.2	-0.44	7.2	-0.28	17.4
F4	39.86	0.30	2.2	-0.25	8.7	-0.28	19.6
F5	60.41	-0.55	2.0	-0.41	9.4	-0.78	21.5
F6	41.70	0.50	3.3	-0.62	12.1	-0.58	26.2
MEAN		0.05	2.6	-0.49	9.7	-0.71	21.4

4.2. Statistical comparison of classifiers. For statistical comparison, we take algorithms ALG2-* and LDA on all real data sets. We test the null hypothesis that all classifiers perform the same and the observed differences are merely random. We used the (Iman and Davenport, 1980) test, which is a nonparametric equivalent of ANOVA.

Let R_{ij} be the rank of the j-th of K classifiers on the i-th of N data sets and $R_j = \frac{1}{N}\sum_{i=1}^{N} R_{ij}$. The test

Table 3. Bootstrap error rates. In the LDA column we have (absolute) bootstrap error rates for the standard LDA method. In the columns ALG1 and ALG2 we have relative bootstrap error rates for all subalgorithms ALG1-1 to ALG2-3, computed by the formula alg*-* – LDA/LDA, where alg*-* is the absolute bootstrap error rate of ALG*-*. All subalgorithm columns are followed by columns with the numbers of linear models combined in the subalgorithms.

Name	LDA	ALG1						ALG2					
of data set		ALG1-1		ALG1-2		ALG1-3		ALG2-1		ALG2-2		ALG2-3	
breastW	4.00	−13.67	2	−13.14	9	−10.64	14	−0.67	1	−0.74	6	−0.98	8
car	18.62	0.00	1	−0.27	2	−0.61	3	0.00	1	−0.27	2	−0.61	3
chemistry	64.92	−6.88	1	−2.75	5	−2.55	6	−6.88	1	−2.75	5	-2.55	6
echo	27.85	−4.17	3	−6.34	4	−5.22	8	−2.05	1	−5.45	2	−4.72	3
fish	35.79	18.88	2	−4.98	5	−3.88	6	18.88	2	−4.98	4	−0.99	5
football	34.57	−7.16	1	−9.29	2	−13.46	5	−5.07	2	−5.48	3	−4.07	4
glass	38.55	−0.99	4	−2.42	6	−2.08	10	−2.63	1	−1.54	2	2.36	3
golf	50.00	−22.69	1	−29.18	3	−27.68	5	v15.12	1	−16.01	4	−11.75	5
hayes	41.85	−1.20	1	0.00	2	0.28	3	−1.20	1	0.00	2	−0.43	3
heart	16.68	5.12	3	2.32	9	−0.63	26	4.62	2	−1.61	3	−2.10	9
heartC	42.02	−0.70	1	−3.79	2	−2.68	17	−0.77	1	−0.68	7	−1.03	5
heartH	40.10	−7.94	1	−11.55	16	−11.67	43	−0.94	2	−2.39	4	−3.22	6
heartS	62.50	−4.43	2	−4.43	8	−3.85	30	−1.38	2	−2.35	5	−1.68	6
iris	2.45	0.00	1	7.64	3	−0.15	6	0.00	1	7.64	3	−0.15	4
irradiation	67.45	−8.66	4	−5.91	7	−3.80	8	−3.50	3	−1.48	4	−1.48	4
liver	33.01	0.00	1	−2.84	2	−1.97	4	0.00	1	−1.48	2	−2.65	3
risk	1.55	0.00	1	−7.45	2	13.63	3	0.00	1	−7.45	2	13.63	3
thyroid	8.65	17.57	1	20.82	2	−1.54	6	−1.96	3	−5.00	4	−2.87	5
turtles	10.49	−1.31	2	−1.17	23	−10.39	29	13.11	6	6.72	7	6.72	7
wine	2.00	−8.17	1	−19.79	8	−16.79	57	−8.17	1	0.66	6	−6.50	7
MEAN		−2.32	2	−4.73	6	−5.28	14	−0.69	2	−2.23	4	−1.25	5
hepatitis	15.44							−1.89	3	−3.94	8	−3.56	12
ionosphere	14.22							−3.89	1	−4.38	2	−3.74	5
kr-vs-kp	6.25							−2.67	1	−2.53	2	−3.10	3
libras	41.01							−0.45	1	−1.02	2	−0.65	5
musk	24.18							0.11	1	−0.51	2	−0.96	9
parkinsons	13.56							-0.49	7	−0.37	10	−0.32	12
sonar	28.44							0.07	2	−2.00	7	−2.15	10
spam	11.18							-0.84	1	−0.71	4	−0.55	5
spectf	27.07							-1.74	2	−0.88	9	−1.49	18
statlog	8.58							0.00	9	0.00	10	0.00	11
vote	6.02							-5.53	1	−2.14	4	−3.14	9
wave	32.36							-4.05	3	−4.50	5	−4.33	11
MEAN								−1.78	3	−1.91	5	−2.00	9

compares the mean ranks of classifiers and is based on the statistic

$$S' = \frac{(N-1)S}{N(K-1)-S}, \tag{9}$$

where

$$S = \frac{12N}{K(K+1)} \sum_{i=1}^{K} R_i^2 - 3N(K+1) \tag{10}$$

is the Friedman statistic which is distributed according to the F distribution with $K-1$ and $(K-1)(N-1)$ degrees of freedom.

If we take into consideration classification errors on all data sets, then in our case $N = 32$ and $K = 4$. The value of statistic S' is equal to 14.23 and the corresponding critical value is equal to $F(0.95, 3, 93) = 2.70$. Due to the fact that the critical value is lower than the respective statistic (p-value is equal to 1.043×10^{-7}), we can proceed with the post-hoc tests in order to detect significant pairwise differences among all the classifiers. A set of pairwise comparisons can be associated with a set of hypotheses. Any of the post hoc tests which can be applied to non-parametric tests work over a family of hypotheses. The test statistics for comparing the i-th and the j-th classifier is

$$Z = \frac{R_i - R_j}{\sqrt{\frac{K(K+1)}{6N}}}.$$

This statistic is asymptotically normally distributed with zero mean and unit variance.

When comparing multiple algorithms, to retain an overall significance level α, one has to adjust the value of α for each *post hoc* comparison.There are various methods for this. The simple method is to use the Bonferroni correction. There are $m = K(K-1)/2$ comparisons, therefore Bonferroni correction sets the significance level of each comparison to α/m. Demšar (2006) recommend the Nemenyi procedure (Nemenyi, 1963) which is based on this correction. Garcia and Herrera (2008) explain and compare the use of various correction algorithms. They showed that although it requires intensive computation, Bergmann and Hommel's dynamic procedure (Bergmann and Hommel, 1988) has the highest power. This procedure is based on the idea of finding all elementary hypotheses which cannot be rejected. To formulate, we need the following definition:

An index set $I \subseteq \{1, 2, \ldots, m\}$ is called *exhaustive* if exactly all $H_j,\ j \in I$, could be true.

Under this definition, the Bergmann–Hommel procedure works as follows: Reject all H_j with $j \notin A$, where the *acceptance set*

$$A = \bigcup \{I :\ I \text{ exhaustive},\ \min\{P_i : i \in I\} > \alpha/|I|\}$$

is the index set of null hypotheses which are retained. For this procedure, one has to check for each subset I of $\{1, 2 \ldots, m\}$ if I is exhaustive, which leads to intensive computations. A fast algorithm (Hommel and Bernhard, 1994) allows a substantial reduction in computing time.

The smallest level of significance that results in the rejection of the null hypothesis, the p-value, is a useful and interesting datum for many consumers of statistical analysis. When a p-value is within a multiple comparison it reflects the probability error of a certain comparison, but it does not take into account the remaining comparisons belonging to the family. One way to solve this problem is to report the Adjusted p-Value (APV) which takes into account that multiple tests are conducted. An APV can be compared directly with any chosen significance level α. Details about computing the APV can be found in the work of Garcia and Herrera (2008).

The results of multiple comparisons are given in Tables 4 and 5. Those classifiers connected by a sequence of stars have average ranks that are not significantly different from each other. We have two homogeneous disjoint groups of classifiers. Classifiers ALG2-1, ALG2-3 and ALG2-2 are significantly better than LDA.

Table 4. Results of the Bergmann–Hommel post hoc test.

Procedure	Ranks mean		
LDA	3.547	*	
ALG2-1	2.422		*
ALG2-3	2.047		*
ALG2-2	1.984		*

Table 5. p-values and adjusted p-values in the Bergmann–Hommel post hoc test.

Hypothesis	p-value	adjusted p-value
LDA vs. ALG2-2	1.290E−6	7.742E−6
LDA vs. ALG2-3	3.358E−6	1.007E−5
LDA vs. ALG2-1	4.909E−4	9.818E−4
ALG2-1 vs. ALG2-2	0.175	0.526
ALG2-1 vs. ALG2-3	0.245	0.526
ALG2-2 vs. ALG2-3	0.846	0.846

5. Conclusions and future work

Our research has shown that the use of a generalization of the Moore–Penrose pseudoinverse of matrices in the LDA method gives good results. In the general case our method seems to outperform LDA. The proposed method, thanks to its parametrical approach, makes it possible to choose an appropriate model for any data set. In many cases there are no statistical differences between the proposed algorithms, but in spite of this the trends are clearly visible.

Of course, the classification performance of the new algorithm needs to be further evaluated on additional real and artificial data. In our technique we can use methods other than the mean one for combining classifier ensembles. This is the direction of our future research.

Estimation of covariance matrices is important in a number of areas of statistical analysis, including dimension reduction by PCA, classification by QDA, establishing independence and conditional independence relations in the context of graphical models, and setting confidence intervals on linear functions of the means of the components. Many application areas where these tools are used have been dealing with high dimensional data sets, and sample sizes can be very small relative to dimension. Examples include genetic data, brain imaging, climate data and many others. In these areas we should look for further applications of our method.

References

Anderson, T.W. (1984). *An Introduction to Multivariate Analysis*, Wiley, New York, NY.

Bensmail, H. and Celeux, G. (1996). Regularized Gaussian discriminant analysis through eigenvalue decomposition, *Journal of the American Statistical Association* **91**(436): 1743–1748.

Bergmann, G. and Hommel, G. (1988). Improvements of general multiple test procedures for redundant systems of hypotheses, *in* P. Bauer, G. Hommel and E. Sonnemann (Eds.), *Multiple Hypotheses Testing*, Springer, Berlin, pp. 110–115.

Chen, L.-F., Liao, H.-Y. M., Ko, M.-T., Lin, J.-C. and Yu, G.-J. (2000). A new LDA-based face recognition system which can solve the small sample size problem, *Pattern Recognition* **33**(10): 1713–1726.

Cozzolino, D., Restaino, E. and Fassio, A. (2002). Discrimination of yerba mate (*Ilex paraguayensis st. hil.*) samples according to their geographical origin by means of near infrared spectroscopy and multivariate analysis, *Sensing and Instrumentation for Food Quality and Safety* **4**(2): 67–72.

d'Aspremont, A., Banerjee, O. and El Ghaoui, L. (2008). First-order methods for sparse covariance selection, *SIAM Journal on Matrix Analysis and Applications* **30**(1): 56–66.

Dempster, A. (1972). Covariance selection, *Biometrics* **28**(1): 157–175.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research* **7**(1): 1–30.

Dey, D.K. and Srinivasan, C. (1985). Estimation of a covariance matrix under Stein's loss, *The Annals of Statistics* **1**(4): 1581–1591.

Dillon, W. and Goldstein, M. (1984). *Multivariate Analysis: Methods and Applications*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, Wiley, New York, NY.

Duda, R., Hart, P. and Stork, D. (2001). *Pattern Classification*, Wiley, New York, NY.

Enis, P. and Geisser, S. (1986). Optimal predictive linear discriminants, *Annals of Statistics* **2**(2): 403–410.

Frank, A. and Asuncion, A. (2010). UCI Machine Learning Repository, University of California, Irvine, CA, `http://archive.ics.uci.edu/ml`.

Friedman, J. H. (1989). Regularized discriminant analysis, *Journal of the American Statistical Association* **84**(405): 165–175.

Garcia, S. and Herrera, F. (2008). An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, *Journal of Machine Learning Research* **9**(12): 2677–2694.

Gleim, G.W. (1984). The profiling of professional football players, *Clinics in Sports Medicine* **3**(1): 185–197.

Hawkins, A.D. and Rasmussen, K.J. (1978). The calls of gadoid fish, *Journal of the Marine Biological Association of the United Kingdom* **58**(4): 891–911.

Hommel, G. and Bernhard, G. (1994). A rapid algorithm and a computer program for multiple test procedures using logical structures of hypotheses, *Computer Methods and Programs in Biomedicine* **43**(3–4): 213–6.

Hong, Z.-Q. and Yang, J.-Y. (1991). Optimal discriminant plane for a small number of samples and design method of classifier on the plane, *Pattern Recognition* **24**(4): 317–324.

Iman, R. and Davenport, J. (1980). Approximations of the critical region of the Friedman statistic, *Communications in Statistics—Theory and Methods* **9**(6): 571–595.

Kuo, B.-C. and Landgrebe, D.A. (2002). A covariance estimator for small sample size classification problems and its application to feature extraction, *IEEE Transactions on Geoscience and Remote Sensing* **40**(4): 814–819.

Kwak, N., Kim, S., Lee, C. and Choi, T. (2002). An application of linear programming discriminant analysis to classifying and predicting the symptomatic status of HIV/AIDS patients, *Journal of Medical Systems* **26**(5): 427–438.

Lim, T.-S., Loh, W.-Y. and Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, *Machine Learning* **40**(3): 203–228.

Morrison, D. (1990). *Multivariate Statistical Methods*, McGraw-Hill Series in Probability and Statistics, McGraw-Hill, New York, NY.

Nemenyi, P. (1963). *Distribution-free Multiple Comparisons*, Ph.D. thesis, Princeton University, Princeton, NJ.

Olkin, I. and Selliah, J. (1975). Estimating covariances in a multivariate normal distribution, *Technical report*, Stanford University, Stanford, CA.

Piegat, A. and Landowski, A. (2012). Optimal estimator of hypothesis probability for data mining problems with small samples, *International Journal of Applied Mathematics and Computer Science* **22**(3): 629–645, DOI: 10.2478/v10006-012-0048-z.

Rao, C. and Mitra, S. (1971). *Generalized Inverse of Matrices and Its Applications*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, Wiley, New York, NY.

Röbenack, K. and Reinschke, K. (2011). On generalized inverses of singular matrix pencils, *International Journal of Applied Mathematics and Computer Science* **21**(1): 161–172, DOI: 10.2478/v10006-011-0012-3.

Sharma, A. and Paliwal, K.K. (2008). A gradient linear discriminant analysis for small sample sized problem, *Neural Processing Letters* **27**(1): 17–24.

Shin, Y.J. and Park, C.H. (2011). Analysis of correlation based dimension reduction methods, *International Journal of Applied Mathematics and Computer Science* **21**(3): 549–558, DOI: 10.2478/v10006-011-0043-9.

Song, F., Zhang, D., Chen, Q. and Wang, J. (2007). Face recognition based on a novel linear discriminant criterion, *Pattern Analysis and Applications* **10**(3): 165–174.

StatSoft, I. (2007). Statistica (data analysis software system), version 8.0, `http://www.statsoft.com`.

Stein, C., Efron, B. and Morris, C. (1972). *Improving the Usual Estimator of a Normal Covariance Matrix*, Stanford University, Stanford, CA.

Swets, D.L. and Weng, J. (1996). Using discriminant eigenfeatures for image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(8): 831–836.

Tian, Q., Fainman, Y., Gu, Z.H. and Lee, S.H. (1988). Comparison of statistical pattern-recognition algorithms for hybrid processing, I: Linear-mapping algorithms, *Journal of the Optical Society of America A: Optics, Image Science and Vision* **5**(10): 1655–1669.

Titterington, D. (1985). Common structure of smoothing techniques in statistics, *International Statistical Review* **53**(2): 141–170.

van der Heijden, F., Duin, R., de Ridder, D. and Tax, D. (2004). *Classification, Parameter Estimation and State Estimation*, Wiley, New York, NY.

Yu, H. and Yang, J. (2001). A direct LDA algorithm for high-dimensional data with application to face recognition, *Pattern Recognition* **34**(10): 2067–2070.

Tomasz Górecki received the M.Sc. degree in mathematics from the Faculty of Mathematics and Computer Science of Adam Mickiewicz University in Poznań in 2001. There, in 2005, he obtained the Ph.D. degree. Currently he is an assistant professor at this university. His research interests include machine learning, times series classification and data mining.

Maciej Łuczak received the M.Sc. and Ph.D. degrees in mathematics from the Faculty of Mathematics and Computer Science of Adam Mickiewicz University in Poznań, Poland, in 2001 and 2005, respectively. He is currently an assistant professor at the Faculty of Civil Engineering, Environmental and Geodetic Sciences of the Koszalin University of Technology, Poland. His research interests are in the area of machine learning and evolutionary algorithms.

Appendix

Lemma A1. *If $\boldsymbol{A}$ is an $m \times m$ invertible matrix and*

$$\boldsymbol{M} = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_m \end{bmatrix},$$

$$\boldsymbol{N} = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ 0 & b_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & b_m \end{bmatrix},$$

where

$$a_i, b_i \in \mathbb{R},$$
$$a_i = 0 \iff b_i = 0,$$

then

$$(\boldsymbol{MA})^+\boldsymbol{M} = (\boldsymbol{NA})^+\boldsymbol{N}.$$

Proof. Let $\boldsymbol{B}$ be an $m \times n$ matrix of rank r. If $\boldsymbol{B}$ can be partitioned in the form

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \boldsymbol{B}_3 & \boldsymbol{B}_4 \end{bmatrix} \tag{A1}$$

by a permutation of rows and columns, if necessary, such that $\boldsymbol{B}_1$ is an $r \times r$ matrix of rank r, $\boldsymbol{B}_2, \boldsymbol{B}_3, \boldsymbol{B}_4$ are matrices of suitable orders, then

$$\boldsymbol{B}^+ = \begin{bmatrix} \boldsymbol{B}_1'\boldsymbol{P}\boldsymbol{B}_1' & \boldsymbol{B}_1'\boldsymbol{P}\boldsymbol{B}_3' \\ \boldsymbol{B}_2'\boldsymbol{P}\boldsymbol{B}_1' & \boldsymbol{B}_2'\boldsymbol{P}\boldsymbol{B}_3' \end{bmatrix}, \tag{A2}$$

where

$$\boldsymbol{P} = (\boldsymbol{B}_1\boldsymbol{B}_1' + \boldsymbol{B}_2\boldsymbol{B}_2')^{-1}\boldsymbol{B}_1(\boldsymbol{B}_1'\boldsymbol{B}_1 + \boldsymbol{B}_3'\boldsymbol{B}_3)^{-1}.$$

The expression (A2) for $\boldsymbol{B}^+$ is due to Penrose (1956).

Since $\boldsymbol{A}$ is invertible, $\boldsymbol{MA}$ can be partitioned in the form (A1) by a permutation of rows and columns, i.e.,

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{MA} = \begin{bmatrix} \boldsymbol{M}_1\boldsymbol{A}_1 & \boldsymbol{M}_1\boldsymbol{A}_2 \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix},$$

where $\boldsymbol{M}_1$, $\boldsymbol{A}_1$ are invertible and $\boldsymbol{M}_1$ is diagonal. Then

$$\begin{aligned} \boldsymbol{P} &= \big(\boldsymbol{M}_1\boldsymbol{A}_1(\boldsymbol{M}_1\boldsymbol{A}_1)' \\ &\quad + \boldsymbol{M}_1\boldsymbol{A}_2(\boldsymbol{M}_1\boldsymbol{A}_2)'\big)^{-1}\boldsymbol{M}_1\boldsymbol{A}_1\big((\boldsymbol{M}_1\boldsymbol{A}_1)'\boldsymbol{M}_1\boldsymbol{A}_1\big)^{-1} \\ &= \big(\boldsymbol{M}_1\boldsymbol{A}_1\boldsymbol{A}_1'\boldsymbol{M}_1 + \boldsymbol{M}_1\boldsymbol{A}_2\boldsymbol{A}_2'\boldsymbol{M}_1\big)^{-1} \\ &\quad \times \boldsymbol{M}_1\boldsymbol{A}_1\big(\boldsymbol{A}_1'\boldsymbol{M}_1\boldsymbol{M}_1\boldsymbol{A}_1\big)^{-1} \\ &= \big(\boldsymbol{M}_1(\boldsymbol{A}_1\boldsymbol{A}_1' + \boldsymbol{A}_2\boldsymbol{A}_2')\boldsymbol{M}_1\big)^{-1}\big(\boldsymbol{A}_1'\boldsymbol{M}_1\big)^{-1} \\ &= \boldsymbol{M}_1^{-1}\big(\boldsymbol{A}_1\boldsymbol{A}_1' + \boldsymbol{A}_2\boldsymbol{A}_2'\big)^{-1}\boldsymbol{M}_1^{-1}\big(\boldsymbol{A}_1'\boldsymbol{M}_1\big)^{-1} \end{aligned}$$

and

$$\begin{aligned} (\boldsymbol{MA})^+\boldsymbol{M} &= \begin{bmatrix} (\boldsymbol{M}_1\boldsymbol{A}_1)'\boldsymbol{P}(\boldsymbol{M}_1\boldsymbol{A}_1)'\boldsymbol{M}_1 & \boldsymbol{0} \\ (\boldsymbol{M}_1\boldsymbol{A}_2)'\boldsymbol{P}(\boldsymbol{M}_1\boldsymbol{A}_1)'\boldsymbol{M}_1 & \boldsymbol{0} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}_1'(\boldsymbol{A}_1\boldsymbol{A}_1' + \boldsymbol{A}_2\boldsymbol{A}_2')^{-1} & \boldsymbol{0} \\ \boldsymbol{A}_2'(\boldsymbol{A}_1\boldsymbol{A}_1' + \boldsymbol{A}_2\boldsymbol{A}_2')^{-1} & \boldsymbol{0} \end{bmatrix}. \end{aligned}$$

Thus, the above expression does not depend on the submatrix $\boldsymbol{M}_1$. ■

Received: 30 May 2012
Revised: 29 September 2012