Dimensionality Reduction for Classification[⋆]

Comparison of Techniques and Dimension Choice

Frank Plastria[1], Steven De Bruyne[2], and Emilio Carrizosa[3]

[1] Vrije Universiteit Brussel Frank.Plastria@vub.ac.be
[2] Vrije Universiteit Brussel Steven.De.Bruyne@vub.ac.be
[3] Universidad de Sevilla ecarrizosa@us.es

Abstract. We investigate the effects of dimensionality reduction using different techniques and different dimensions on six two-class data sets with numerical attributes as pre-processing for two classification algorithms. Besides reducing the dimensionality with the use of principal components and linear discriminants, we also introduce four new techniques. After this dimensionality reduction two algorithms are applied. The first algorithm takes advantage of the reduced dimensionality itself while the second one directly exploits the dimensional ranking. We observe that neither a single superior dimensionality reduction technique nor a straightforward way to select the optimal dimension can be identified. On the other hand we show that a good choice of technique and dimension can have a major impact on the classification power, generating classifiers that can rival industry standards. We conclude that dimensionality reduction should not only be used for visualisation or as pre-processing on very high dimensional data, but also as a general pre-processing technique on numerical data to raise the classification power. The difficult choice of both the dimensionality reduction technique and the reduced dimension however, should be directly based on the effects on the classification power.

1 Introduction

Dimensionality reduction of the feature space has long been used in data mining. So is principal component analysis, first introduced in [3], often used as a pre-processing step for problems with an extreme high dimensionality. Linear discriminant analysis, first introduced in [2], on the other hand, while sharing many properties, is used to solve the problem immediately by reducing the dimensionality of the feature space to one. The optimal number of dimensions for principal component analysis has been investigated many times [6], but with the goal of finding the number of non-trivial components. We on the other hand use another criterion, namely maximization of the 10-fold cross validation results of the classifiers yielded by the algorithms. The optima for both these criteria do not necessarily coincide.

[⋆] Partially supported by the research project OZR1372 of the Vrije Universiteit Brussel

We start by defining the different techniques we use. Some of them are standard techniques while others are new. Later we compare them to see which one delivers superior results.

To evaluate the effect of the dimensionality reduction, we use two-class data sets on which we apply two classification algorithms. The first one is an algorithm that normally functions in the original feature space and we are interested to see if the reduction of the dimensionality can have positive effects nonetheless. The second algorithm is a new algorithm that directly uses the properties of the new feature space introduced by the transformation and reduction of the dimensionality.

A full overview of each possible combination of dimensionality reduction technique, dimension choice and algorithm is presented and analysed.

2 Dimensionality Reduction Techniques

The reduction of the dimension of the attribute vector is executed in three steps. First, we compute a square transformation matrix of dimension the number of attributes. Second, the attribute vectors are transformed by multiplying them with this matrix. Finally, the actual reduction consists of keeping a fixed number of the most discriminating new attributes (features).

Several techniques to compute an effective transformation matrix are evaluated. Principal components and linear discriminants are selected as standard solutions. The principal components orientate the data so that the variance is maximized firstly along the first axis, secondly along the second axis etc. Principal components ignore the existence of multiple classes. Linear discriminants however take into account the existence of multiple groups by trying to find orientations that favour a high spread of the total data while avoiding those yielding a high spread within the different homogenous groups. These two techniques are complemented by four new ones. The first one, which we named principal separation, is by idea similar to linear discriminants but looks for orientations in which pairs of instances of different classes lie far apart. The three final approaches exploit the fact that only two groups exist and that a straightforward candidate for the first orientation can be defined by the means of the two groups. Since this does not yield a complete transformation matrix, it is complemented by orientations computed using each of the aforementioned techniques.

2.1 Notations

For a general matrix $M \in \mathbb{R}^{d \times p_M}$

- d : the original dimension of the data (number of attributes)
- $Mean(M) \in \mathbb{R}^{d \times 1}$: the mean of the instances of M
- $Cov(M) \in \mathbb{R}^{d \times d}$: the covariance matrix of M
- $Mom(M) \in \mathbb{R}^{d \times d}$: the matrix of second moments (around the origin) of M
- $Eig(M) \in \mathbb{R}^{d \times d}$: the matrix of eigenvectors of M

Note: $Cov(M) = Mom(M) - Mean(M).Mean(M)^t$

2.2 Data

- A : the matrix of p_A columns representing the instances of the first set
- B : the matrix of p_B columns representing the instances of the second set
- $T = [A, B]$: the matrix of $p_T = p_A + p_B$ columns representing the instances of both sets

2.3 Reduction operation

- R : the transformation matrix
- $n < d$: the dimension after reduction

R is assumed to be ordered row-wise. Usually the rows of R are the eigenvectors of some matrix, ordered by decreasing eigenvalues.

The dimension reduction consists of calculating RT and dropping all rows except the first n. The result is a new feature space.

2.4 Principal Components

We define the transformation matrix based on principal components as

$$R = Eig(Cov(T))$$

2.5 Linear Discriminants

Let

$$S_{\mathrm{W}} = \frac{p_A Cov(A) + p_B Cov(B)}{p_T}$$

$$S_{\mathrm{B}} = Cov(T) - S_{\mathrm{W}}$$

Then we define the transformation matrix based on linear discriminants as

$$R = Eig(S_{\mathrm{W}}^{-1} S_{\mathrm{B}})$$

2.6 Principal Separation Components

Define the $d \times (p_A p_B)$ matrix $A \ominus B$, as consisting of all d-vectors $a - b$ for any pair of d-vectors $a \in A$ and $b \in B$.

We want to keep the 'spread'

$$\sum_{a-b \in A \ominus B} \|a - b\|^2$$

as high as possible after reduction. In a way similar to principal components analysis, this is obtained using transformation matrix

$$R = Eig(Mom(A \ominus B))$$

Since the complexity of calculating $Mom(M)$ is in general $O(p_M d^2)$ the calculation of $Mom(A \ominus B)$ seems at first glance to be an ominous $O(p_A p_B d^2)$. This complexity is however strongly reduced to $O((p_A + p_B)d^2) = O(p_T d^2)$ thanks to the following result.

Theorem 1 $Mom(A \ominus B) = Mom(A) + Mom(B) - Mean(A)Mean(B)^t - Mean(B)Mean(A)^t$

Proof
For any $1 \leq i, j \leq d$ we have:

$$Mom(A \ominus B)_{ij} = \frac{1}{p_A p_B} \sum_{a \in A} \sum_{b \in B} (a_i - b_i)(a_j - b_j)$$

$$= \frac{1}{p_A p_B} \sum_{a \in A} \sum_{b \in B} (a_i a_j - b_i a_j - a_i b_j + b_i b_j)$$

$$= \frac{1}{p_A} \sum_{a \in A} a_i a_j + \frac{1}{p_B} \sum_{b \in B} b_i b_j - \frac{1}{p_A p_B} \sum_{a \in A} \sum_{b \in B} a_i b_j - \frac{1}{p_A p_B} \sum_{a \in A} \sum_{b \in B} b_i a_j$$

$$= \frac{1}{p_A} \sum_{a \in A} a_i a_j + \frac{1}{p_B} \sum_{b \in B} b_i b_j - (\frac{1}{p_A} \sum_{a \in A} a_i)(\frac{1}{p_B} \sum_{b \in B} b_j) - (\frac{1}{p_B} \sum_{b \in B} b_i)(\frac{1}{p_A} \sum_{a \in A} a_j)$$

$$= Mom(A)_{ij} + Mom(B)_{ij} - Mean(A)_i Mean(B)^t_j - Mean(B)_i Mean(A)^t_j$$

□

2.7 Mean Component Methods

Let

$$p = Mean(A) - Mean(B)$$

We define the first row of the transformation matrix based on means as

$$R_1 = \frac{p}{||p||}$$

We define the remaining rows $R_{2..n}$ as the $n-1$ first rows of the aforementioned techniques after projection of the instances on the hyperplane perpendicular on R_1. This yields the following three variants:

- principal mean components
- linear mean discriminants
- principal mean separation components

3 Classification Algorithms

3.1 Optimal Distance Separating Hyperplane

Here we use a linear classifier, chosen so as to minimize sum of the euclidean distances of misclassified instances to the separating hyperplane as proposed in [5]. By reducing the dimension, we hope not only to solve the problem of the high complexity the algorithms have concerning the dimensionality of the feature space [4, 7], but also hope to reduce overfitting. The classifier is obtained by way of the Grid-Cell VNS algorithm as was proposed in [7].

3.2 Eigenvalue-based Classification Tree

Since the ranking of *RT* indicates the importance of the features of the new feature space, it is straightforward to make a selection for a split in a hierarchical structure. We therefore we propose a new kind of classification tree in which the split in the top node is done based on the first feature, in the nodes on the second level the splits are done based on the second feature etc. The split value is calculated by taking that midpoint between two consecutive instances that minimizes the number of misclassifieds. The expansion of the tree ends either when only instances of one class remain for the given node or when the level of the node equals the reduced dimension n. Note that no feature is used more than once in each branch.

4 Computational Results

Table 1 shows the best average results of ten 10-fold cross validations on six data sets from the UCI [8] database when applying all six reduction techniques and varying dimensionality. As a reference, the results of some industry standards such as support vector machines with linear kernels (SVM) and C4.5 classification trees (C4.5), as well as a 1-dimensional principal component (PCA) or a linear discriminant analysis (LDA) are also presented. We can see that, combined with a good dimensionality reduction, our new algorithms (Best ODSH, Best EVCT) can compete with these standards.

Table 2 shows the optimal dimensionalities and reduction techniques for each pair of algorithm and data set. No dimensionality reduction technique seems superior or inferior to the others, which makes a selection difficult. Although severe dimensionality reductions often yield good results, finding a pattern for an effective selection is difficult. It should also be noted that no clear correlation was found between the results and the eigenvalues yielded by the dimensionality reduction techniques.

Full results can be found in the figures below. Although an optimal selection is difficult without full results, the impact of good pre-processing can be clearly seen here. A reduction to one dimension can give good results, but in many cases a reduction of this magnitude seems too drastic and moving up to two or three dimensions can result in a very significant improvement. On the other hand, the results can decrease rapidly when choosing a higher dimensionality, although for some sets, there seems little gain in reducing the dimensionality at all. The optimality of the dimension seems to be mainly determined by the structure of the data and the fact that the goal is classification and less by the algorithm and the reduction technique.

Table 1. Cross Validations

Data set	d	p_T	Best ODSH	Best EVCT	SVM	C4.5	PCA	LDA
Cancer	9	683	96.8%	97.5%	97.0%	95.4%	97.5%	97.1%
Diabetes	8	768	76.1%	75.6%	76.8%	74.4%	68.1%	75.6%
Echocardiogram	7	74	76.5%	73.4%	70.9%	70.9%	65.2%	67.5%
Glass windows	9	214	93.7%	96.1%	92.2%	93.3%	92.2%	93.2%
Hepatitis	16	150	83.6%	84.5%	84.6%	77.6%	80.9%	84.5%
Housing	13	506	86.8%	81.7%	86.4%	81.9%	77.1%	79.9%

Table 2. Optimal Dimension Reduction

Data Set	d	p_T	ODSH		EVCT	
			Best Tech	Best Dim	Best Tech	Best Dim
Cancer	9	683	LD	1	PC/PSC	1
Diabetes	8	768	PMC	6	LD	1
Echocardiogram	7	74	PSC	2	PMSC	2
Glass windows	9	214	PMC	2	PMC	2
Hepatitis	16	150	PC	1	LD	1
Housing	13	506	PMSC	10	PSC	6

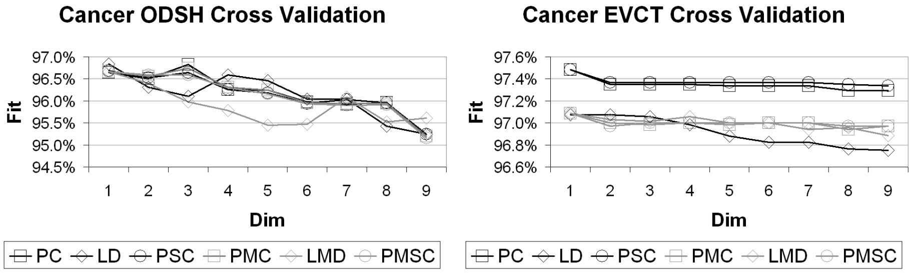

Fig. 1. Cancer set cross validations

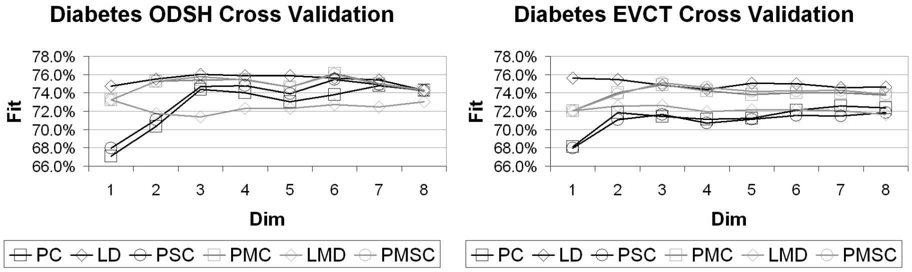

Fig. 2. Diabetes set cross validations

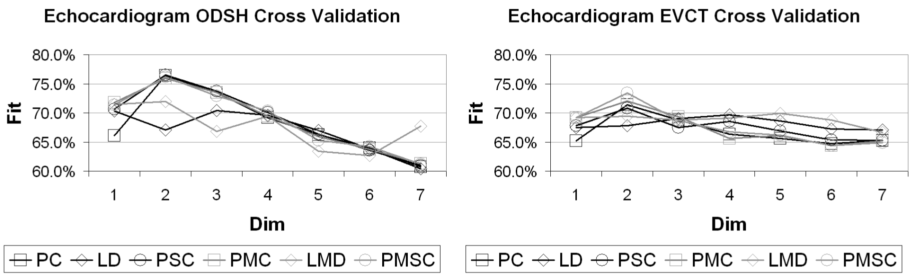

Fig. 3. Echocardiogram set cross validations

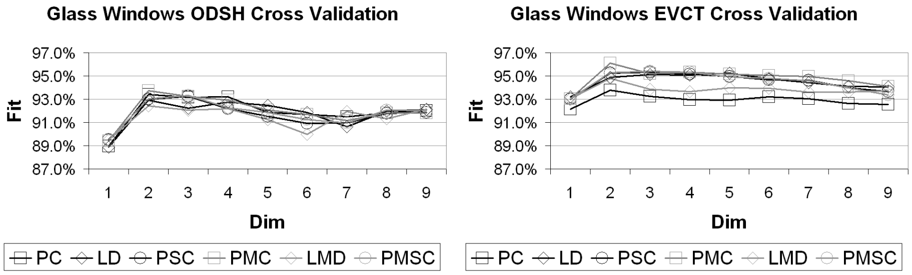

Fig. 4. Glass windows set cross validations

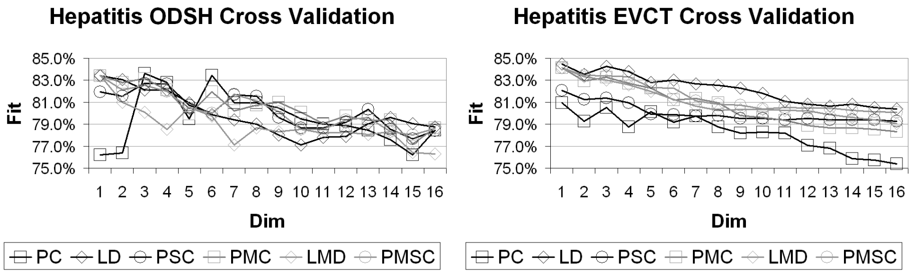

Fig. 5. Hepatitis set cross validations

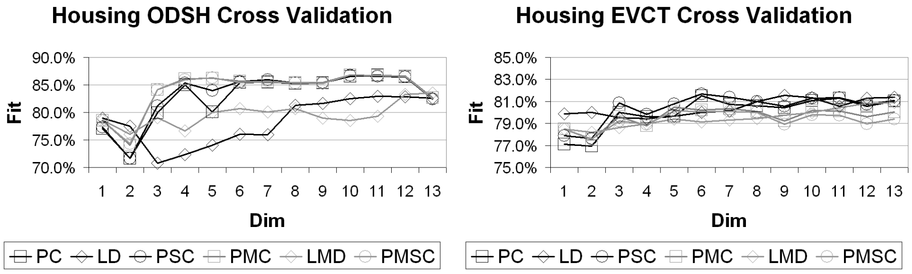

Fig. 6. Housing set cross validations

5 Conclusions

We showed that a good choice of technique and dimension can have a major impact on the classification power. Lowering the dimensionality often significantly reduces the overfitting by an algorithm. On the other hand, one should be careful not to lower the dimensionality too much as the information loss can then rapidly overtake the reduction in overfitting. However, we observe that neither a single superior dimensionality reduction technique nor a straightforward way to select the optimal dimension can be identified.

We conclude that dimensionality reduction should not only be used for visualisation or as pre-processing on very high dimensional data, but also as a general pre-processing technique on numerical data to raise the classification power. The difficult choice of both the dimensionality reduction technique and the reduced dimension however, should be directly based on the effects on the classification power. We are currently researching methods to incorporate the choice of the dimensionality reduction into the algorithm itself [1].

Many of these findings can also be extended to problems with more than two classes or can be used in combination with kernels.

References

1. De Bruyne S., Plastria F. 2-class Internal Cross-validation Pruned Eigen Transformation Classification Trees. *Optimization Online* http://www.optimization-online.org/DB_HTML/2008/05/1971.html
2. Fisher, R.A (1936) The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* 7: 179-188
3. Hotelling, H. (1933) Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24: 417–441.
4. Karam A., Caporossi G., Hansen P. (2007) Arbitrary-norm hyperplane separation by variable neighbourhood search. *IMA J Management Math 2007* 18: 173–189. http://imaman.oxfordjournals.org/cgi/content/abstract/18/2/173?etoc
5. Mangasarian, O.L. (1999) Arbitrary-Norm Separating Plane. *Operations Research Letters* 24: 15–23.
6. Peres-Neto P., Jackson D., Somers K. (2005) How many principal components? stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis* 49: 974–997
7. Plastria F., De Bruyne S., Carrizosa E. Alternating local search based VNS for linear classification. *Optimization Online* http://www.optimization-online.org/DB_HTML/2008/02/1910.html
8. Newman D.J., Hettich S., Blake C.L., Merz C.J. (1998). UCI Repository of machine learning databases http://www.ics.uci.edu/˜mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.