Relational Discriminant Analysis and Its Large Sample Size Problem

Robert P.W. Duin
Pattern Recognition Group, Department of Applied Physics
Delft University of Technology, The Netherlands

Abstract

Relational discriminant analysis is based on a similarity matrix of the training set. It is able to construct reliable nonlinear discriminants in infinite dimensional feature spaces based on small training sets. This technique has a large sample size problem as the size of the similarity matrix equals the square of the number of objects in the training set. In this paper we discuss and initially evaluate a solution that drastically decreases training times and memory demands.

1. Introduction

Traditionally the main approach in statistical pattern recognition is feature based: Objects are individually described by features. Classes of objects are represented in a feature space by sets of feature vectors. Discriminants are constructed in such a feature space separating these sets, possibly based on density estimations of the classes. This approach suffers from a feature size - sample size dilemma. Better, more complete object representations, yield feature spaces of higher dimensionalities which on their turn demand larger sample sizes in order to realize the possibility of better discriminants [1]. Consequently the small sample size problem is permanent issue in the pattern recognition literature [2].

In this paper we will discuss the possibilities, advantages and drawbacks of an entirely different way of object representation. Instead of an individual description by features now the set of available objects is represented by all object relations, like distances, in a similarity matrix. These relations may be computed from a feature representation, but might also be computed directly from digitized or even analogue object differences.

In the relational approach the training set is represented by a square $m*m$ matrix D containing all pairwise relations d(x_i,x_j) between the objects x_i and x_j, (i,j = 1,m). For the moment we will assume that these relations are distances. Other similarity measures, however, are equally possible. A new object x is now classified by a function S(•) of the distances d(x_i,x) between this object and (a subset of) the training objects. During training this subset and the parameters in S(•) have to be optimized. In case the similarities are based on a feature description, which, we repeat, is not necessary, a larger feature set may or may not result in a more accurate D. It does not, however, imply the need of a larger training set. In fact, for any size m of the training set classifiers S(•) can be computed. So there is no small sample size problem.

In contrary, we have a large sample size problem, which is of a computational nature. As the size of D is $m*m$, memory demands and computing times may grow quadratically with the size of the training set. It is the purpose of this paper to address this problem and to show that there are effective training methods that circumvent the large sample size problem for a good deal. These methods are inspired by, but not identical to the support vector methods that have recently been proposed in the machine learning community by Vapnik and others [3], [4], [5]. As a result we have a set of interesting new pattern recognition methods that offer a new way of representing expert knowledge: distance or similarity measures instead of features. As will be shown, these methods may be nonlinear, and can, for feature based representations, operate in feature spaces of any dimensionality and may be based on small training sets.

This paper elaborates further on ideas first presented in [9], [10], [11] on featureless pattern recognition. The relational approach addressing the large sample size problem is presented here for the first time.

2. Relational Discriminant Analysis

Let the set of training objects be given by $X = \{x_1, ... , x_m\}$. D(X,X) is its known $m*m$ similarity matrix. A new object x

should now be classified using the (m,1) vector y = D(X,x), containing its similarity values to the training set. In a first approach the set of linear classifiers

$$S(x) = \sum_{i=0,m} w_i y_i = \boldsymbol{w} \bullet \boldsymbol{y} \qquad (1)$$

is considered. A constant term w_0 has been added by defining $y_0 = 1$. Note that this classifier is linear in $\boldsymbol{w}$ and $\boldsymbol{y}$ but might be nonlinear in the original measurements, depending on the definition of D(•). In particular it should be realized that instead of the original D(•) any $D' = K(D)$ may be used if K(•) is a monotonic function like exp(•), log(•) or power. In this way the relative importance of larger and smaller similarity values can be changed.

In a two-class problem with target outcomes -1, +1 for the two classes stored in a vector λ, the weights $\boldsymbol{w}$ directly follow from the desired outcomes for the training set:

$$S(X) = \boldsymbol{w}^T Y = \boldsymbol{w}^T D(X,X) = \lambda^T \qquad (2)$$

So $\boldsymbol{w}^T = \lambda^T Y^{-1}$ (3)

In case rank(Y) < m the mean square error solution has to be used (Fisher’s Linear Discriminant).

The similarity matrix D can be considered to be a pattern matrix: a set of m objects defined by m relational features. These features describe the similarities with all training objects. The following observations can now be made:

1. A classification problem in which sample size and feature size are equal clearly suffers from the small sample size problem. In section 3 this will be discussed further.
2. Instead of the standard linear classifier (1), trained by (3) any other classifier based on the training set as represented by D might be used as well.
3. One of the solutions for the small sample size problem is feature reduction. In case of relational features, feature selection is similar to editing the training set.
4. Large training sets cause problems as they involve the manipulation (e.g. inversion, like in (3)) of $m*m$ matrices. This point and the previous one will be discussed further in section 4.

3. Small Sample Size Problem

The computation and performance of linear classifiers as discussed in the previous section for sample sizes m in the order of the dimensionality has been extensively investigated by us during the recent years, e.g. [6], [7], [8]. A typical learning curve for a 30-dimensional Gaussian problem is sketched in fig. 1. For m > 30 the Fisher discriminant is used. For m = 30 an exact solution as in (3) is obtained and for m < 30 a pseudo-inverse is used, see [6]. Sample sizes equal to the dimensionality appear to be maximally bad due to some resonance phenomenon: the noise in the training set is maximum and entirely reflected in the discriminant. For larger training sets the noise is decreased by averaging, for smaller training sets the total amount of noise is less, see [8]. As a consequence the point where the sample size equals the feature size should definitely be avoided. Generally there are four options:

1. reduce the feature size (feature selection)
2. reduce the sample size, either at random or systematically. This is the basis of our work in [6] and of the support vector classifier [3], [4], [5].
3. enlarge the sample size.
4. enlarge the feature size.

As we argue in [12], the options 3 and 4 can both be realized by adding noise, in different ways. In connection with the relational approach discussed in this paper, the options 1 and 2 both reduce the similarity matrix D, but in different ways: columns versus rows. Similarity measures are often symmetric, but, nevertheless, there is an important difference.

Reducing samples (rows) implies that the *point of operation* is moved to the left in fig. 1. This corresponds with less samples in a feature space of constant dimensionality. The training set is thereby reduced into a support set [3]. The pay off of having a larger training set has to be found by the careful selection of this support set.

Reducing features (columns) implies that the *maximum* in the curve of fig. 1 shifts to the left. This corresponds to a space of lower dimensionality. In this space all training points are still represented.

The difference between the two approaches becomes clear by observing what happens if new objects become available after the reduction. In the case of a support set this does not help unless this set is recomputed. In case of a computed feature subset new objects may still be helpful as thereby the object set in the relational feature space grows.

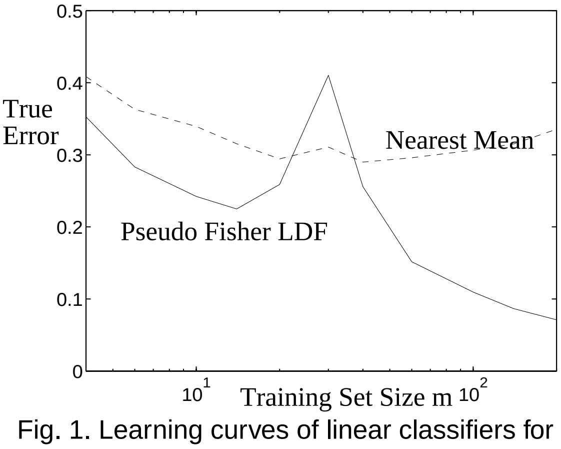

Fig. 1. Learning curves of linear classifiers for a Gaussian 30 dimensional problem.

4. Large Sample Size Problem

The methods used for reducing the similarity matrix should be judged on both, their feasibility as well as their performance. We have argued above that reduction in one way or another is necessary before good generalizable discriminants can be computed. In this section we will face the problems that arise with increasing training sets. A direct application of (3) yields for large sample sizes an increasing noise sensitivity and a decreasing performance. Reduction will become even more necessary. The manipulation of the similarity matrix, however, becomes for larger sample sizes, say over 1000, a serious computational problem. This has been partially solved by Burges and Schölkopf [16]. The computational demands, however, are still heavy.

Because of the above it may be highly interesting to study relational discriminant methods that apply a feature reduction on the similarity matrix instead of an object reduction as thereby all objects remain represented. The question arises what an efficient reduction is for relational features. Especially for large datasets it becomes less important to study optimal selection methods as the relational features are similar for neighboring objects. Moreover, each representation has some intrinsic dimensionality: the maximum number of independent directions that is essential for the representation. Suppose that this intrinsic dimensionality is k, then almost any subset of k+1 objects used for the feature representation constructs a k-dimensional subspace which is, except for some linear transformation, identical to the optimal subspace. So, for linear solutions like (3) a random feature selection may be a good start, i.e. random selection of the objects used for the relational features.

A random selection may be improved in two ways: multiple random trials (stochastic optimization) or by using a systematic approach instead. The latter may be based on an observation we borrow from the ideas behind the support vector classifier, see [3]: good objects for representing a discriminant are close to the discriminant, i.e. around the margin between the classes. In an iterative way these objects may be found by adding the erroneously classified training objects to an initial object set.

In the next section some examples will be presented illustrating the above ideas.

5. Examples

The first example, see figure 2, shows the use of all training objects in (3) for a simple 2-dimensional problem. Three similarity measures are used: $(x_1 \bullet x_2)^3$, resulting in a 3rd-order polynomial discriminant, $||x_1\text{-}x_2||^2$ yielding a quadratic function and $||x_1\text{-}x_2||$. For this last measure rank(D) = m. Consequently it classifies all objects correctly and is thereby overtrained. For this situation we are in the top of the peak in figure 1.

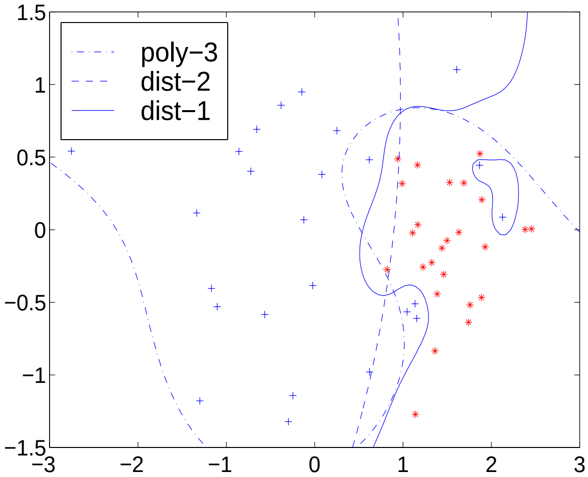

Fig. 2. Examples of classifiers based on all objects.

In figure 3 an example is shown in which the classifier is a linear function of a similarity representation of just two, randomly selected objects. The left figure shows the original 2-dimensional feature space and the final classifier. In the right figure the relational feature space is shown based on the squared Euclidean distances to the two selected objects. These objects are thereby exactly on the axis. In this space a linear classifier between all training objects is computed, which is equivalent to the quadratic classifier in the left figure, the original feature space.

The next example shows the result of the following procedure:

1. select an arbitrary object, k = 1.
2. compute the (m,k) similarity matrix to all training objects
3. compute a linear classifier in the resulting k-dimensional space.
4. If not all objects are correctly classified add the most erroneously classified object (largest distance to the classifier), k = k + 1, go to 2.
5. Remove objects from the selection as long as the training error remains zero.

The left figure shows two banana shaped classes. In the right figure the well known spiral problem is shown. The objects that are selected for the relational features are encircled.

Finally a character recognition experiment was run on a large NIST database [14]. We used 20000 numerals ‘0’ - ‘9’ (2000 for each class) as normalized by DeRidder [11], [13] in 16x16 grey value images. The similarity is computed as the Euclidean distance between the 256 pixels of two objects (these distances were raised to the 1.4 power for historical, arbitrary reasons). The total dataset is randomly split in 1000 objects per class for training and 1000 objects per class for testing. Out of the total training set random

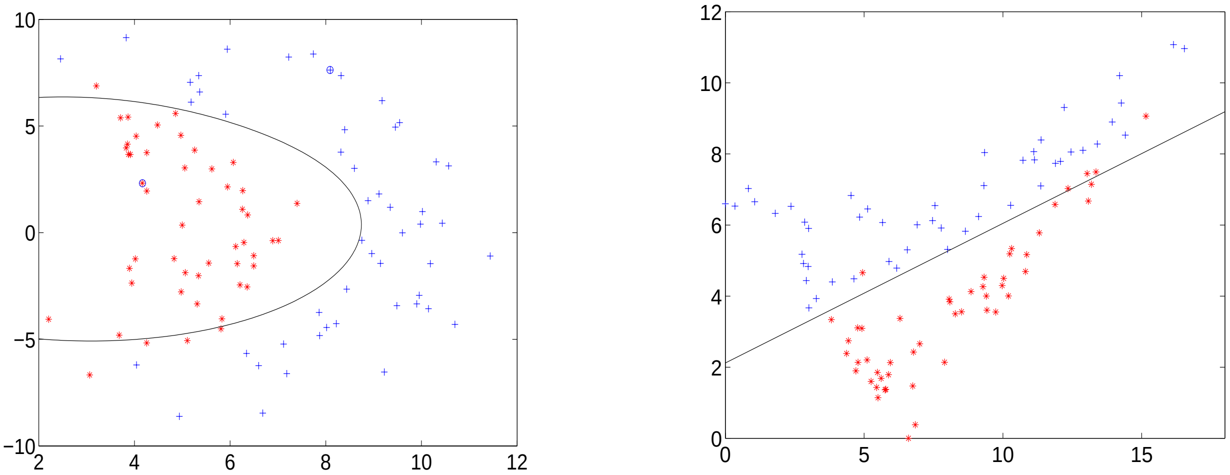

Fig. 3. Left: Original 2-dimensional feature space with final discriminant. Right: transformed feature space based on the distances to two randomly chosen training objects

subsets of 10, 20, 50, 100, 250 and 1000 objects per class are selected. Out of each subset randomly *n* objects per class are selected (*n*=1,2,3,4) to be used as the *k* relational features. So *k* = 10,20,30,40. For these features linear and quadratic classifiers are computed. It appears that the quadratic classifiers perform much better. Results are shown in figure 5. In the left figure the averaged results are shown over 25 experiments with different random selections of the relational features. It appears that *n* = 3 (*k* = 30) is almost sufficient for representing the training set. For small training sets smaller feature sets (*n* = 1,2, so *k* = 10, 20) are better.

The best of our 25 experiments with *n* = 4 (which is just slightly better than the average) is compared in the right figure with some historical results obtained by DeRidder on the same training set [11], [13] using a LeCun neural network [15], the nearest neighbor rule and a 3rd order support vector classifier [3].

The results that are obtained in this character recognition experiment are reasonably good, in particular if the computing time is taken into account. Our relational discriminant needs for the largest experiment with in total 10000 training objects (1000 objects times 10 classes) in an originally 256 dimensional feature space just a few minutes on a Sun-Ultra-1 processor. This is due to the random reduction of the 10000*10000 similarity matrix to a 10000*40 matrix. In the resulting 40-dimensional feature space a discriminant is trained by all 10000 training objects. Also the application of this discriminant to new objects is relatively fast in comparison with the other classifiers as it primarily demands the computation of the distances to just 40 training objects.

6. Discussion

Relational discriminant analysis offers the possibility to compute nonlinear classifiers from small or large training sets. The original object representations should be such that similarity matrices can be computed. This is possible for any feature representation as well as for other representations. This enables the use of different kinds of prior knowl-

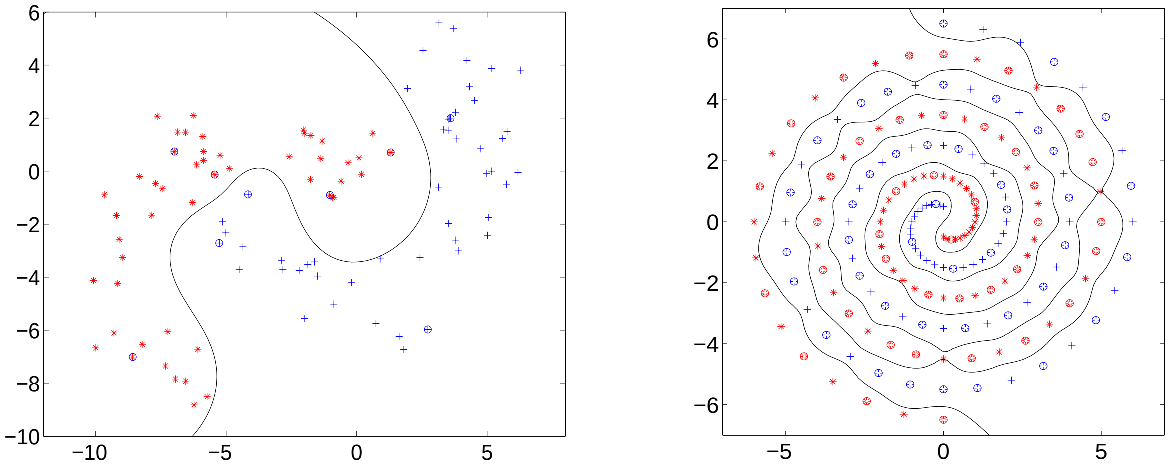

Fig. 4. Two 2-dimensional examples of a systematic procedure removing all training errors.

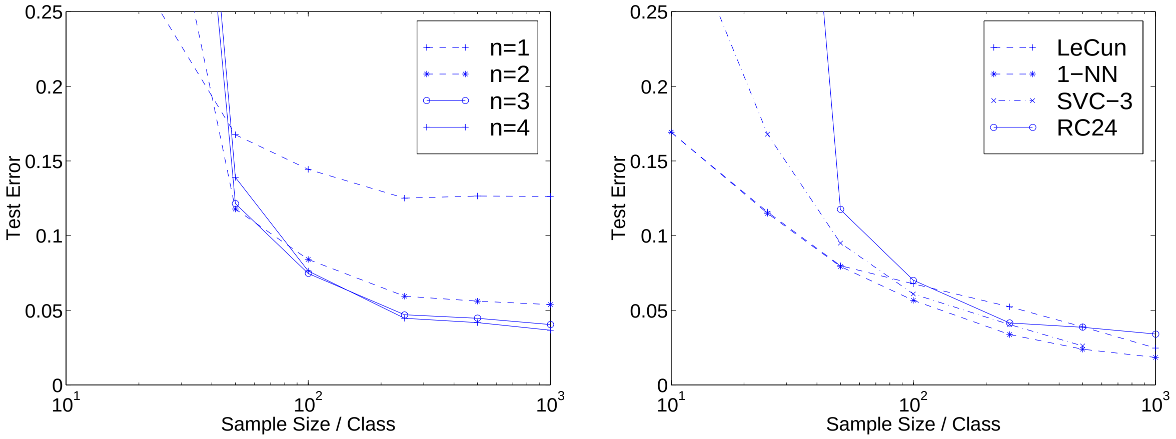

Fig. 5. Character recognition experiment. Left the learning curves averaged over 25 experiments with a relational discriminant. Right the best of these 25 experiments compared with some historic results.

edge for the definition of the object relations, (dis)similarities or distances.

A *m*m* similarity matrix on the training set can be interpreted as *m* objects given by *m* features (similarities to training set objects). Its direct use for building a linear classifier is not advisable as exactly for this situation (sample size equals feature size) the learning curve peaks. Using less objects as in the support vector classifier is an option but has the drawback that it is computationally intensive and is entirely defined by the selected object subset. In this paper we found that the selection of a small set of relational features is very well possible: it is fast, both for training and testing, and may perform well. Moreover, it has the advantage over the support vector classifier that new training objects can directly be included.

An open question is still the selection method for finding the relational features: random or systematic. The latter will certainly decrease the training speed. Other open issues are the final discriminant computed on top of the reduced similarity matrix and its possible nonlinear transformations. Here we experimented with both, linear as well as nonlinear discriminants. Nonlinear similarity transformations will have to be studied in relation with this with the similarity measure itself.

7. References

[1]A.K. Jain and B. Chandrasekaran, Dimensionality and Sample Size Considerations in Pattern Recognition Practice, in: P.R. Krishnaiah and L.N. Kanal (eds.), *Handbook of Statistics*, vol. 2, North-Holland, Amsterdam, 1987, 835 - 855.

[2]S.J. Raudys and A.K. Jain, “Small sample size effects in statistical pattern recognition: recommendations for practitioners,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 252-264, 1991.

[3]V.N. Vapnik, *The nature of statistical learning theory*, Springer Verlag, Berlin, 1995.

[4]B. Schölkopf, *Support vector learning*, Oldenbourg Verlag, Munich, 1997.

[5]C.J.C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, 1997.

[6]R.P.W. Duin, *Small sample size generalization*, SCIA'95, Proc. 9th Scand. Conf. on Image Analysis, Volume 2, 1995, 957-964.

[7]M. Skurichina and R.P.W. Duin, Stabilizing classifiers for very small sample sizes, *Proc. 13th Int. Conf. on Pattern. Recognition*, Vol. 2, IEEE Comp. Society Press, 1996, 891-896.

[8]S. Raudys and R.P.W. Duin, On expected classification error of the Fisher Linear Classifier with pseudo-inverse covariance matrix. *Pattern Recognition Letters*, 1998 (in press).

[9]R.P.W. Duin, D. de Ridder, and D.M.J. Tax, Featureless Classification, *First Int. Workshop Statistical Techn. in Pattern Recognition*, Prague, 1997, 37-42.

[10]R.P.W. Duin, D. de Ridder, and D.M.J. Tax, Experiments with object based discriminant functions; a featureless approach to pattern recognition, *Pattern Recognition Letters*, 18, 1997, 1159-1166.

[11]R.P.W. Duin and D. de Ridder, Neural network experiences between perceptrons and support vectors, in: A.F. Clark (ed.), *Proc. of the 8th British Machine Vision Conference*, volume 2, University of Essex, Colchester, UK, 1997, 590-599.

[12]M. Skurichina and R.P.W. Duin, Regularization by adding redundant features, *STIPR98*, Sydney, 1998.

[13]D. de Ridder, *Shared weights neural networks in image analysis*, Master Thesis, Delft Univ. of Techn., February 1996.

[14]C.L. Wilson, M.D. Marris, *Handprinted character database 2*, April 1990. NIST, Advanced Systems division.

[15]Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, Backpropagation applied to handwritten zip code recognition, *Neural Computation*, vol. 1, 1989, 541-551.

[16]. Burges and B. Schölkopf, Improving the accuracy and speed of support vector machines. In: M. Mozer, M. Jordan, and T. Petsche (eds.): *Neural Information Processing Systems*, Vol. 9. MIT Press, Cambridge, MA, 1997.