Approximate Distance Classification

Adam H. Cannon Lenore J. Cowen Carey E. Priebe
Department of Mathematical Sciences
The Johns Hopkins University
Baltimore, MD, 21218

Abstract

We investigate the use of a class of nonlinear projections from a high-dimensional Euclidean space to a low-dimensional space in a classification (supervised learning) context. The projections developed by Cowen and Priebe approximately preserve interclass distances. Projected data was obtained from data sets that have been extensively studied in the machine learning and statistical pattern recognition communities, and analyzed in the projected space using standard statistical techniques. A simple implementation involving no pre-processing or data dependent adjustments produced results that are near-competitive with the best known established classification rates on these benchmark data sets. Thus even in moderate dimensional spaces the utility and robustness of classification schemes based on these projections is demonstrated.

1 Introduction

Classification and clustering in high dimensions are notoriously difficult problems. Conventional methods that work well on low-dimensional data are often unable to uncover sufficient structure for data in higher-dimensional spaces. Instead of searching for clustering structure of the original, high-dimensional observations, it is common practice to employ dimension reduction methods. The question "How to project?" naturally arises. Some success has been achieved in moderately high-dimensional spaces using linear projections combined with projection pursuit methods (see Huber [8] and Asimov [1]), but finding useful linear projections in very high dimensions frequently remains a barrier. In 1997, Cowen and Priebe [4] introduced a class of nonlinear projections that is easy to construct and has been demonstrated to preserve clustering structure in high-dimensional data sets that strongly cluster. The motivation behind their work is to reduce dimensionality while approximately preserving intercluster distances. Consequently the classification and clustering techniques based on them are referred to as *Approximate Distance Classification and Clustering* methods or *ADC* methods for short.

The *ADC* projections Cowen and Priebe present in [4] and [10] are a family of projections to low-dimensional space, each indexed by a subset of the observations (called the *witness set*). In Cowen and Priebe's paper, *ADC* was presented as a "battle-axe in a dark room", a crude tool that could preserve some , and sometimes enough, clustering structure so that classification and clustering could be accomplished in 10^6-dimensional space, a dimensionality for which conventional methods fail for theoretical and computational reasons. In this paper, it is shown that on data with a few number of classes in moderately high dimensional-spaces, we can build simple classifiers based on 1-dimensional *ADC* projections[1] that are surprisingly competitive with the best, most highly tuned methods on the data sets we examined.

In order to build a classifier based on *ADC* projections we needed to solve three problems:

- **Sample Problem:** How many projections do we need to generate to get some that are useful?
- **Recognition Problem:** How do we distinguish the "best" (or most useful) projections from the rest?
- **Resolution Problem:** How do we classify an observation if our analysis would give conflicting class labels for different projections?

The first of these problems is addressed to some degree by Cowen and Priebe [4]. In the sequel we explore possible solutions to the second and third problems and test their merits experimentally on known benchmark data sets. The data sets considered in this study have 2-3 classes, no missing data values, and range in dimensionality from 4 to 30 dimensions.

[1] We believe that one-dimensional projections sufficed partially because of the low number of classes in the chosen data sets. See discussion in Section 9.

The techniques in this paper extend immediately to higher dimensional data; in this case the issue was that most of the benchmarks we found to compare with were only of moderately high dimensions. Extending to data sets with a large number of classes on the other hand, will probably require projections to j-dimensional space (for $j > 1$), rather than just the 1-dimensional projections.

2 The Method

2.1 Problem Formulation

The classification or supervised learning problem may be described in the following way using the notation of Devroye, Györfi and Lugosi [5]. Suppose we are given D_n, a collection of n labelled observations in $\Re^d$, where for each observation $X_i \in \Re^d$, we are also given an associated class label $Y_i \in \{1, \ldots, C\}$, where C is finite. We call D_n the labeled training data and assume that it is a representative sample of the general population of interest. Given a new unclassified observation $X \in \Re^d$, we wish to predict its associated class label Y.

Formally, let D_n be fixed and $g_{D_n} : \Re^d \to \{1, \ldots, C\}$. If g_{D_n} is well defined for all $D_n \subseteq \{\Re^d \times \{1, \ldots, C\}\}^n$, then $g_n : \Re^d \times \{\Re^d \times \{1, \ldots, C\}\}^n \to \{1, \ldots, C\}$ is called a discriminant function or *classifier.* If it is now assumed that the training data is a set of random variable pairs (X_i, Y_i) from the same distribution as (X, Y), when D_n is sampled independently according to the distribution of (X, Y), $L(g_n) = P\{g_n(X) \neq Y\}$ is the probability of error of classifier g_n. Intuitively, a good classifier is one which minimizes the probability of error. If the sequence $\{g_n\}$ is defined for each integer $n > n_0$, then $\{g_n\}$ is called a *classification rule.* For the remainder of the present paper we will usually assume that there are only two classes, 0 and 1. That is, $Y_i \in \{0, 1\}$.

2.2 *ADC* Projections

Given a set of observations in a high-dimensional space we first seek a projection of the data into a lower-dimensional space for which approximate intercluster distances are maintained. In this paper, we map to $\Re^1$.(see [4] for the general definition of the ADC map.)

Definition 1 *Let $S = \{x_1, x_2, \ldots, x_n\}$ be a collection of n vectors in $\Re^d$. Let $D \subset S$, and $\|\cdot\|$ denote the L_2 norm. The associated ADC map is defined as the function*

$$ADC_D : x_i \to \min_{z \in D} \|x_i - z\|.$$

The set D in the above definition will be referred to as the *witness set* that generates its associated projection. Clearly each ADC map is completely determined by the witness set used and each determines a projection from $\Re^d$ to $\Re$. In what follows, we will always choose D entirely from one of the classes, without loss of generality, call it class 1. Note that since the class labels Y_i are known it is easy to choose all the members of D from within the same class in the training set.

2.3 Identifying Good Witness Sets

It is hoped that the projected data will retain some desirable characteristics that will allow the conventional methods to classify accurately. If they do, a standard classical method (We compared several: k-nearest neighbor, standard linear, and standard quadratic discriminant functions, see Section 3.) is trained on the labeled projected training data. This gives a classifier for the projected data set, $g_{n,D}$. When presented with a new unlabeled observation X, the function $g_{n,D}$ classifies by projecting X to one-dimensional space using ADC_D and then labeling $ADC_D(X)$ using the trained conventional discriminant function. We call the conventinal function used in combination with the ADC map the *ADC subclassifier.*

Cowen and Priebe show that if the original data clusters well in some sense, then some witness sets will lead to projections that are of sufficient quality for successful classification. Given training data consisting of n pairs of vectors $(X_i, Y_i) \in \Re^d \times \{0, 1\}$, with $n_0 = \sum_{i=1}^n I_{\{Y_i=0\}}$ and $n_1 = \sum_{i=1}^n I_{\{Y_i=1\}}$, without loss of generality, let us consider witness sets sampled uniformly at random from class 1. If we limit the size of the witness sets to $n_1/2$, there are $\sum_{j=1}^{\lfloor \frac{n_1}{2} \rfloor} C(n_1; j)$ possible witness sets and therefore at most this many projections. (Some witness sets may generate identical projections.) We would like to identify the sets and associated projections that are most useful to us.

There are many existing methods to measure the quality of a projection (see Devroye, Györfi and Lugosi [5], Huber [8], Diaconis and Freedman [6]). Here the quality of a projection generated by a witness set D was measured based on how well a subclassifier would perform on a observation from the training set if instead that observation had been deleted from D_n. That is, the projection generated by D was evaluated with respect to a particular ADC subclassifier g by using the deleted estimate (also called leave-one-out, cross validation, or U-method) for the probability of error, $L(g_{n,D})$, of the subclassifier when applied to the training data projected by ADC_D. If we let $D_{n,i}$ be the training sequence with the ith pair (X_i, Y_i) deleted, the leave-one-out estimate

of $L(g_{n,D})$ is given by:

$$\hat{L}_n(g_{n,D}) := \frac{1}{n}\sum_{i=1}^{n} I_{\{g_{n-1,D}(X_i,D_{n,i})\neq Y_i\}},$$

where the $D_{n,i}$ is included in the argument of the classifier to indicate that it was trained on the deleted sequence $D_{n,i}$. More details on the deleted estimate can be found in Devroye, Györfi and Lugosi [5], chapter 24.

2.4 Filtering and Combining Results

So far the procedure is as follows. First, we sample w witness sets from the set of all size s subsets of the training data in class 1. Then, the deleted estimate evaluates each witness set that is chosen. Now we select the r best-scoring witness sets. (How to set w, s and r is discussed in the sequel). r is called the *filtering parameter.* Observe that each of these witness sets implicitly assigns in a natural way a class label to any new unknown observation: namely, the class that the subclassifier on the projected data according to that witness set would assign. If we have reason to believe that the training data closely resembles the test data in distribution, the best-scoring witness sets, would classify with the lowest probability of error.

Note, however, that when $w > 1$, multiple witness sets may not agree on the assigned class label. If there is a conflict, it is resolved using majority vote (w will be set as an odd number). We found that taking $w > 1$ and voting improved the performance of our classifier, as will be seen by our results reported in Section 4.

3 Selecting *ADC* Subclassifiers

In describing the *ADC* procedure recall that the conventional decision function used to classify the projected data was called the *ADC* subclassifier. We compared the *ADC* classification method plugging in the three nonparametric classifiers described below.

k-Nearest Neighbor. The k-nearest neighbor rule is due to Fix and Hodges (1951). According to this rule, given a training sequence D_n, to classify an unlabeled observation, X, we look at the k closest observations to X in the training data and vote to determine a label for X. Formally,

$$g_n(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_{ni} I_{\{Y_i=1\}} > \sum_{i=1}^{n} w_{ni} I_{\{Y_i=0\}} \\ 0 & \text{otherwise,} \end{cases}$$

where $w_{ni} = 1/k$ when X_i is among the k nearest neighbors of x and $w_{ni} = 0$ otherwise.

Linear Discriminant Function. Given an unlabeled observation X, a linear discriminant function is one that is linear in the components of X. In particular, we can write the linear discriminant function h as $h(x) = w^t x + w_0$, where w is a weight vector and w_0 a threshold value. The training data is used to determine values for w and w_0.

A classifier based on a linear discriminant function may be defined

$$g_n(x) = \begin{cases} 1 & \text{if } h(x) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

In our case, since we are classifying one-dimensional data, we use the L_1 distance to the class sample means, $\hat{\mu}_0$ and $\hat{\mu}_1$, weighted by class size to construct a discriminant function. Hence when $\hat{\mu}_1 < \hat{\mu}_0$, (which is what we expect when sampling witness sets from class 1) we set

$$h(x) = -1 \cdot x + \frac{n_0\hat{\mu}_0 + n_1\hat{\mu}_1}{n}.$$

Quadratic Discriminant Function. The quadratic discriminant function is a generalization of the linear function that includes products of pairs of components of X. We will base our quadratic discriminant function on the Mahalanobis distance of the unlabeled observation X to the class sample means $\hat{\mu}_0$ and $\hat{\mu}_1$. Given a distribution with covariance matrix Σ and mean μ, the Mahalanobis distance between a vector x and the mean μ is given by

$$\mathcal{M}(x,\mu) = \sqrt{(x-\mu)^t\Sigma^{-1}(x-\mu)}.$$

Using sample covariance matrices for each class we can define the quadratic discriminant function as

$$h(x) = \mathcal{M}(x,\hat{\mu}_1) - \mathcal{M}(x,\hat{\mu}_0).$$

The classifier based on this function is defined exactly as it was in the linear case. Namely,

$$g_n(x) = \begin{cases} 1 & \text{if } h(x) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

4 Experimental Results

All three of the data sets used here are available from the *Machine Learning Repository* of the Computer Science Department of the University of California at Irvine [9]. We used these data sets because have been extensively studied and are well documented. The "best known results" which we cite on each of these data sets are from the same source.

To test performance, a five-fold cross validation procedure was implemented on the three sets of data. For each set the data were partitioned into five equally sized cells. Five experiments were performed, in each one a different cell was reserved for test data. Multiple versions of the *ADC* classifier were constructed by changing the subclassifier, witness set size, and filter parameter r. In all of the experiments 30 witness sets were initially sampled from the training data before filtering. [2] The *ADC* classifier was trained on the other four cells and then tested on the heldout test data. The fraction of correctly classified test data was averaged over the 5 experiments and is reported for each variation of the classifier's parameters.

Wisconsin Breast Cancer Data This database was donated to the UCI Repository by Nick Street in November 1995. It consists of 569 observations, each with 30 real valued input features describing characteristics of cell nuclei obtained from a digitized image of a fine needle aspirate of a breast mass. There are two classes, malignant and benign. There are no missing attribute values and the class distribution is 357 benign and 212 malignant.

Three tables are presented in the appendix summarizing our results for the Wisconsin breast cancer data. Witness sets were sampled from the benign class. Each table corresponds to a different choice for the filtering parameter r: 1, 5 and 11. Increasing r tended to improve the classification rate. In each case results were computed for witness set sizes of 5, 10, 20 and 30, and the mean result for each witness set size is reported in the table. Six subclassifiers were used: the linear and quadratic functions described in Section 3, and three versions of k-nearest neighbor with k set to 1, 5, 9. It was easy to get over 93.9%, showing robustness of this method to poorly optimizing the parameters w, s, and r at this misclassification threshold. For comparison purposes, ordinary k nearest neighbors was also implemented for odd $k = 1, \ldots 7$. The best among these was 5 nearest neighbors, and it achieved only 93.1% classification rates. Our best reported result is 95.6% compared to a reported best known result of 97.5%

Iris Plants Database. This is the famous Fisher Iris Data and was donated to the UCI Repository by Michael Marshall in July 1988. Each observation has 4 attributes (plant characteristics). There are 3 classes each referring to a type of iris plant. There are a total of 150 observations, 50 in each class.

For the Fisher Iris data, varying the filtering parameter r had little effect on the results, hence results from only one scheme, with $r = 3$ are tabulated in the tables. Results from experiments that used witness set sizes of 3, 5, and 10 are reported. For witness set sizes over 10, accuracy did not increase significantly and decreased in some cases. All three varieties of subclassification functions were implemented. Due to the relatively small number of observations in each class, k-nearest neighbor was run using $k = 1$, $k = 3$, and $k = 9$ only. Since the Fisher Iris data has 3 classes, 3 passes were made for each witness set size sampling witness sets from one class each time. That is, the classification problem was treated as three 2-class problems, distinguishing C_i and $\overline{C_i}$ for $i = 1, 2, 3$. Each table shows how many class i observations were misclassified when sampling from class i. All parameter variations gave 100 percent correct class 1 classification. Class 2 and 3 classifications proved more sensitive to subclassifier choice, with nearest neighbor rules outperforming linear and quadratic discriminants. Sampling witness sets from class 2 gave a better rate of correct classifications (best was 94.7%) than sampling from class 3 (best was 94%). The best results achievable on this data set is known to be 2-3 misclassifications.

Pima Indian Diabetes Database. This database was donated to the UCI Repository by Vincent Sigillito at the Applied Physics Laboratory of The Johns Hopkins University in 1990. It contains 768 instances each with 8 numeric valued attributes. Patients are females at least 21 years of age and of Pima Indian heritage. There are two classes, tested positive or negative for diabetes. There are no missing attribute values and the class distribution is 268 tested positive and 500 tested negative.

The witness sets were sampled from the negative observations. Only one value of the filtering parameter $r = 5$ is reported with witness set sizes of 5, 10 and 20. Again, the three different subclassification functions were used. As with the Fisher Iris data, k-nearest neighbor subclassifiers were implemented using $k = 1, k = 3$, and $k = 9$. Variance over the 5-fold cross validation seemed significant; we report not only the mean results over the 5 experiments but also the worst and best cases. For comparison purposes, ordinary k nearest neighbors was also implemented for odd $k = 1, \ldots 7$. The best among these was 7 nearest neighbors, and it achieves slightly better performance than we do.

Due to great differences in scale, range, and variance among the attributes of the Pima Indian diabetes data,

[2] The number 30 was determined empirically: initial experiments were conducted sampling 10, 20, 30, and 40 witness sets and while accuracy tended to increase as more sets were sampled, the improvement was relatively small after 30.

it may be desirable to attempt some preprocessing of the data before implementing the *ADC* classification algorithm. Here, only the raw data has been analyzed using the same procedure described above.

5 Conclusions

Results on the Wisconsin breast cancer data set and the Fisher iris data set compare very well with previous work on these data. The Pima Indian diabetes results are also nearly competitive with previous work. In all three cases it should be emphasized that these results are obtained using a very simple implementation of the 1-dimensional *ADC* procedure. The classifiers are not data dependent and no preprocessing of data is carried out. Furthermore, considerable robustness to parameter settings is also evident. As witness set size, $|D|$, and filtering parameter r were varied, results remained relatively stable.

Another strength of the procedure presented here is its flexibility with respect to the goodness criterion used for evaluating witness sets. In this paper we have constructed the sets $\mathcal{G}_{c,r}$ using an empirical error minimization approach on the projections generated by the witness sets. However, we are free to use any goodness criterion we choose to evaluate witness sets or projections without altering the underlying algorithm. While there are many existing methods for evaluating projections, it may also be possible to find some desirable structure in the witness set itself to quantify without generating its corresponding projection.

Furthermore, any desired classification function may be used as a subclassifier. Here we have only used three varieties, but optimizing subclassifier selection over bigger classes of functions is another area where we suspect improvements can be made.

All of these flexibilities give the *ADC* approach a modular nature that allows for easy implementaion and should provide great adaptability to a wide range of data sets.

While we showed utility of 1-dimensional *ADC* as a classifier experimentally on data sets which had very few classes and no missing data values, it remains an open problem to extend these methods to data sets with more classes and missing data values. One straightforward way to attack the multiple class case is to consider all 2-class subproblems, as was done here for the 3 classes in Fisher's iris data. However, as the number, C, of classes grows, this may not be the best approach. To find projections that will be good for multiple classes simultaneously, Cowen and Priebe's [4] treatment of the multiple cluster case suggests that it will be necessary to consider the j-dimensional *ADC* projections where $j > 1$ grows as a function of the number of classes.

Acknowledgements

The authors thank Clyde Schoolfield for an initial implementation of ADC on the Iris data in the early stages of this project. This work was partially supported by ONR research grants N00014-96-1-0829 and N00004-96-1-0313.

References

[1] D. Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM J. Sci. Statist. Comp.*, 6:128–143, 1985.

[2] P.J. Bickel and K.A. Doksum. *Mathematical Statistics*. Prentice Hall, 1977.

[3] G. Casella and R.L. Berger. *Statistical Inference*. Wadsworth, Inc., 1990.

[4] L.J. Cowen and C.E. Priebe. Randomized nonlinear projectons uncover high-dimensional structure. *Advances in Applied Math*, 19:319–331, 1997.

[5] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[6] P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *The Annals of Statistics*, 12:793–815, 1984.

[7] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

[8] P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13:435–475, 1985.

[9] C.J. Merz and P.M. Murphy. UCI repository of machine learning databases, 1996.

[10] C.E. Priebe and L.J. Cowen. Approximate distance clustering. Submitted for Publication, 1997.

[11] W.N. Venables and B.D. Ripley. *Modern Applied Statistics with S-Plus*. Springer-Verlag, 1994.

ADC subclassifier	$\|D\| = 5$	$= 10$	$= 20$	$= 30$
1-nearest neighbor	86.8	87.7	87.7	77.2
5-nearest neighbor	93.0	**93.9**	92.1	86.8
9-nearest neighbor	92.1	93.0	90.4	89.5
Linear Discr.	93.0	93.0	93.0	93.0
Quadratic Discr.	**93.9**	**93.9**	91.2	**93.9**

Table 1: Wisconin Breast Cancer Data: $r = 1$. Best known result: 97.5%, 5-nearest neighbor result: 93.1% Our best *ADC* result with 1 vote: 93.9%

ADC subclassifier	$\|D\| = 5$	$= 10$	$= 20$	$= 30$
1-nearest neighbor	90.4	92.1	90.4	88.6
5-nearest neighbor	92.1	93.9	92.1	91.2
9-nearest neighbor	92.1	93.9	93.0	90.4
Linear Discr.	93.0	91.2	93.0	91.2
Quadratic Discr.	89.5	93.9	93.0	**94.7**

Table 2: WBCD: $r = 5$, Best known result: 97.5%, 5-nearest neighbor result: 93.1%. Our best *ADC* result: 94.7%.

ADC subclassifier	$\|D\| = 5$	$= 10$	$= 20$	$= 30$
1-nearest neighbor	92.1	92.1	91.2	91.2
5-nearest neighbor	92.1	92.1	93.0	93.0
9-nearest neighbor	92.1	93.0	93.0	92.1
Linear Discr.	93.0	91.2	92.1	91.2
Quadratic Discr.	90.4	**95.6**	93.0	93.9

Table 3: WBCD: $r = 11$, Best known result: 97.5%, 5-nearest neighbor result: 93.1%. Our best *ADC* result: 95.6%.

ADC subclassifier	$\|D\| = 3$	$= 5$	$= 10$
1-nearest neighbor	94	93.3	94.7
3-nearest neighbor	93.3	92.0	93.3
9-nearest neighbor	92.6	94.7	94.7
Linear Discr.	89.3	90	91.3
Quadratic Discr.	87.3	86	84.7

Table 4: Fisher Iris Data, Class 2 classification. (Class 1 separation was 100 percent in every column)

ADC subclassifier	$\|D\| = 3$	$= 5$	$= 10$
1-nearest neighbor	93	93.3	87.3
3-nearest neighbor	94	94	91.3
9-nearest neighbor	92	91.3	93.3
Linear Discr.	52	47	45
Quadratic Discr.	87.3	86	88.7

Table 5: Fisher Iris Data, Class 3 classification.

***ADC* subclassifier**	**Mean**	**Worst**	**Best**
1-nearest neighbor	62.3	54.9	66.0
3-nearest neighbor	69.1	64.0	**73.9**
9-nearest neighbor	68.6	61.4	73.2
Linear Discr.	65.4	64.7	66.7
Quadratic Discr.	66.1	64.0	67.0

Table 6: Pima Indian Results, $r = 5$, Witness set size $= 5$, best known result: 76%, 7-nearest neighbor result: mean = 72.8%, best = 75.0%, worst=70.3%. Our best *ADC* result: 73.9%.

***ADC* subclassifier**	**Mean**	**Worst**	**Best**
1-nearest neighbor	66.5	60.1	71.2
3-nearest neighbor	69.1	64.0	**73.9**
9-nearest neighbor	69.1	63.4	72.5
Linear Discr.	65.4	64.1	66.7
Quadratic Discr.	66.1	64.0	66.7

Table 7: Pima Indian Results, Witness set size = 10, best known result: 76%, 7-nearest neighbor result: mean = 72.8%, best = 75.0%, worst=70.3%. Our best *ADC* result 73.9%.

***ADC* subclassifier**	**Mean**	**Worst**	**Best**
1-nearest neighbor	66.5	60.1	71.2
3-nearest neighbor	69.1	64.1	72.5
9-nearest neighbor	72.2	64.1	**74.5**
Linear Discr.	65.4	63.4	67.3
Quadratic Discr.	65.5	64.1	66.7

Table 8: Pima Indian Diabetes Results, Witness set size = 20, best known result: 76%, 7-nearest neighbor result: mean = 72.8%, best = 75.0%, worst=70.3%. Our best *ADC* result 74.5%.