DMP Report: Summer 2006
Parametric models for similarity-based classification

Anjali Koppal
University of California, Berkeley

Abstract

Keywords: parametric, classifier, similarity-based.
A parametric classifier builds class models and classifies test points as being from classes that they are most likely to belong to, given the class models. Over the summer, I worked with Professor Maya Gupta (Department of Electrical Engineering, University of Washington), and Luca Cazzanti (Applied Physics Laboratory, University of Washington), to develop a parametric classifier that builds models based on only the similarity between samples. The parametric classifier was compared to other traditional classifiers, including near-neighbor methods, centroid-based classifiers, and support vector machines.

1 Introduction

Common real-life classification problems only provide relational information about the data, such as pairwise similarities. Often, information about the actual data samples is unavailable. One exampleis a protein classification problem from Hochreiter and Obermayer [2006], where the pairwise dissimilarities between protein structures is easily available, but the description of each structure is not available. Classifiers must therefore be able to classify data given only pairwise relational information. A new classifier proposed by Cazzanti and Gupta is a similarity-based parametric classifier that performs comparably well to highly sophisticated classifiers like support vector machines (SVMs) and traditional near neighbor approaches.
In Section 2, I will discuss various similarity metrics and existing classification mechanisms. In Section 3, I explain the proposed similarity-based parametric classifier. Finally, in Section 4, I will describe a few experiments that compare the proposed classifier with other existing classifiers. The results and analyses presented in this paper are explained in detail in Cazzanti et al.. A draft of this paper can be requested by email to gupta@ee.washington.edu.

2 Background

2.1 Similarity metrics

There are many ways of defining the similarity metric for a dataset. The simplest similarity measure is Hamming similarity which is the number of common features. Tversky [1977] studied similarity from the perspective of Psychology, and argued that how humans judge similarity is different from distance, in that it may not be symmetric and may not follow the triangle inequality law. Lin [1998] proposed the following similarity metric which has shown to be an example of the Tversky model [Cazzanti and Gupta, 2006]:

$$s(A,B) = \frac{log(P(A \cap B))}{log(P(A \cup B))}$$

Lin's similarity thus gives two objects a high similarity value if they share a large number of features. An entropy-related similarity metric that incorporated information about the context was developed by Cazzanti and Gupta [2006]. Contexual information is important in developing a similarity metric. For example, if a large number of objects in a set are identical, an object that has one different feature must be considered very dissimilar, in the given context. In this approach, the similarity between A and B is defined to be:

$s(A,B) =$
$-H(R|A \cap B \in R) + \frac{H(R|A \setminus B \in R)}{2} + \frac{H(R|B \setminus A \in R)}{2}$
where $H(R)$ is the entropy of a random object R picked from a given distribution.
In the extreme case, if $A = B$, $A \cap B$ completely describes one and only one object, and so $H(R|A \cap B \in R) = 0$. Also, since $A \setminus B = B \setminus A = 0$, the entropy associated with the other two terms is maximum. Thus

$s(A, A)$ has a large positive value. Context is incorporated in the fact that for different distribution probabilities (different $P(R)$s, therefore different $H(R)$s), the similarity values are different.

2.2 Near Neighbors Classification

Near Neighbors is the simplest example of nonparametric techniques of classification [Duda et al., 2001]. The k-nearest-neighbor classification rule is: Classify test point t as being in the class that the maximum of its k nearest neighbors belong to. The k nearest neighbors are k training points that are closest to t. When the points are in Euclidean space, "closest" is defined as the minimum Euclidean distance to t. In a similarity context, "closest" is defined as the maximum similarity (or, minimum dissimilarity). Choosing the value of k is crucial to the near neighbor method. A common method used to determine the value of k is cross-validation.
An example of the nearest-neighbor method is presented in Figure 1. The test point is in black, and its nearest neighbors are all red points, so it is classified as being in class "red".

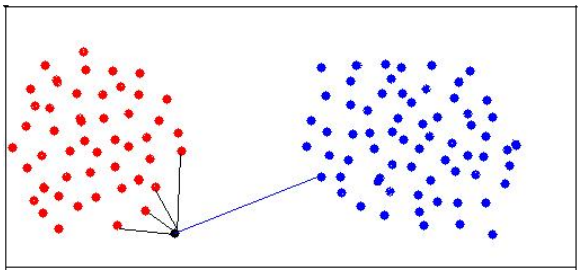

Figure 1: The test point "black" is classified as class "red" because its nearest neighbors are points from class "red".

2.3 Centroid Approaches

Some classifiers attempt to model a class by determining a representative element, or prototype, for the class. For example, the prototype (or centroid) of a class can be the training sample that is closest (or most similar) to all other training samples from the class. In this way, prototypes, $p_1, p_2, \ldots p_n$ are developed for classes, $c_1, c_2, \ldots c_n$. When a test point t is to be classified, the distance between t and $p_1, p_2, \ldots p_n$ is calculated, and t is classified as being in class i where $d(t, p_i) = \arg\min_{1 \leq j \leq n}(d(t, p_j))$. Thus, the classification rule for the centroid approach is, classify test point x as being in class 1 if:

$$s(x, \mu_1) > s(x, \mu_2)$$

where μ_1 and μ_2 are centroids or representative elements of classes 1 and 2. An adjusted centroid approach would be,

$$\frac{s(x,\mu_1)}{\bar{s}_{11}} > \frac{s(x,\mu_2)}{\bar{s}_{22}}$$

The normalizing factors help to include the intra-class variance information. Figure 2 shows an example of the centroid approach. The red and black dots represent the "centroids" of the 2 gaussian distributions. The test point (the black dot) is closer to the blue centroid, so it is classified as being in class 2.

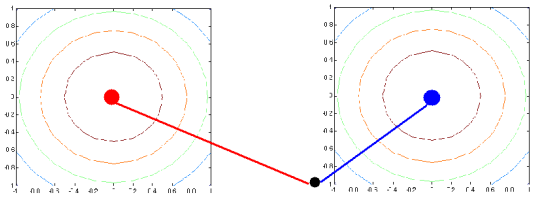

Figure 2: The test point "black" is classified as class "blue" because it is closer to the blue centroid.

2.4 Support Vector Machines

Support Vector Machines fall into the category of classifiers that attempt to model a discriminant; that is, they find a hyperplane that separates samples from class 1 from those of class 2 in some higher dimension [Cristianini and Taylor, 2000]. A hyperplane is essentially a plane in n dimensions. Although the concept of SVMs in Euclidean space is fairly easy to understand, the problem becomes more complicated when the classifier has to deal with similarities, and therefore create hyperplanes in a similarity space. Since similarities need not follow properties of Euclidean space [Tversky, 1977], SVMs must first tranform the similarity space to a manageable space. This is done by defining a "kernel" which is an inner product to convert points in the feature space to points in the "manageable" higher dimensional space. The fact that we do not have to be concerned with how to define the higher dimensional space, just the kernel, is what is known as the "kernel trick" [Cristianini and Taylor, 2000]. A generalized SVM (PSVM) developed by Hochreiter and Obermayer [2006] can use any similarity matrix as a kernel for SVMs, and we compare the proposed classifier to the PSVM.

2.5 Linear/Quadratic Discriminant Analysis

Linear Disciminant Analysis methods assume that each class has an underlying (unknown)distribution. They form discriminant functions, that divide the sample space into as many subspaces as there are classes [Duin et al., 1999] [Pekalska and Duin, 2002]. In the simplest case, the discriminant function, g, is a hyperplane. Thus, in the example (Figure 3), a test point (black) is classified as class "red" if $g(black) > 0$ and class "blue" if $g(black) < 0$.

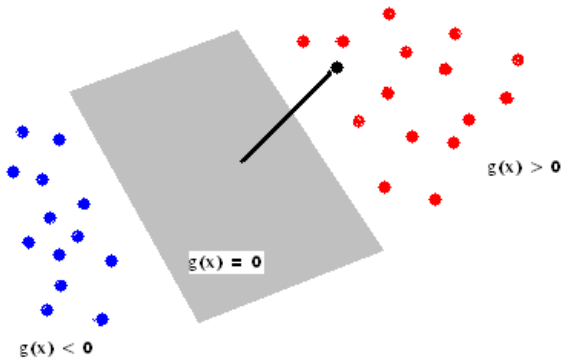

Figure 3: The test point "black" is classified as class "red" because $g(black) > 0$.

3 Similarity Based Parametric Classifier

The parametric classifier's classification rule is classify test point x as being in class 1 if,

$$\frac{P(y=1|x)}{P(y=2|x)} > 1$$

The classifier describes sample point x by 2 parameters: $s(x, \mu_1)$ and $s(x, \mu_2)$, that is the similarity of the test point to the centroids of the 2 classes.

$$\frac{P(y=1|s(x,\mu_1),s(x,\mu_2))}{P(y=2|s(x,\mu_1),s(x,\mu_2))} > 1$$

Using Bayes' rule, the classification rule then becomes

$$\frac{P(s(x,\mu_1),s(x,\mu_2)|y=1)P(y=1)}{P(s(x,\mu_1),s(x,\mu_2)|y=2)P(y=2)} > 1$$

The terms $P(y = 1)$ and $P(y = 2)$ are known as class priors; they are the probabilities of seeing a sample from class 1 or 2.
An assumption made here is that, given the class label, the similarity of x to μ_1 and μ_2 are independent. Thus the rule becomes

$$\frac{P(s(x,\mu_1)|y=1)P(s(x,\mu_2)|y=1)P(y=1)}{P(s(x,\mu_1)|y=2)P(s(x,\mu_2)|y=2)P(y=2)} > 1$$

The problem remains to compute the probabilities. This is done in the following way: we develop a constraint on the probability values using the maximum likelihood estimate (MLE) of a similarity statistic [Duda et al., 2001] and then find the maximum entropy solution to the constraint. It has been shown that the maximum entropy is also the maximum likelihood with mild assumptions [Jaynes, 1998].

1. MLE constraint:
   The constraint developed is the MLE estimate of $E[S_{12}]$, where
   $s_{12} = \sum_{x \in 1} s(x, \mu_2)$.
   $E[S_{12}] = \sum_{s_{12} \in S} s_{12} P(s_{12}) = \bar{s}_{12}$
   This constraint is called a moment constraint.

2. Maximum Entropy Solution:
   The maximum entropy solution to this problem is known to have the following solution:
   $P(s_{12}) = \gamma_{12} exp(\lambda_{12} s_{12})$.

3. Solving by Optimization:
   Using the Matlab function fminsearch, the constraint is numerically solved to determine values for γ and λ.

4 Experiments

The parametric classifier was compared to nearest-neighbor methods, centroid approaches, and SVMs. Both experiments on real datasets and simulations were conducted on all these classifiers and the results for three experiments have been presented below. For a detailed explanation of all the experiments, please refer to Cazzanti et al..

4.1 Protein Data Set

The Protein data set is a collection of pairwise similarities for 226 proteins, and it is available at the UCI Machine Learning repository [Newman et al.,

1998]. Following Hochreiter and Obermayer [2006], we used 213 proteins from four classes: "HA"(72 samples), "HB"(72 samples),"M"(39 samples), "G"(30 samples). Table 1 shows the percentage misclassification for the four "one class vs the rest" problems for each of the classifiers. Percentage misclassification is the defined to be the percentage of false positives and false negatives for each class. The parametric classifier performs better than the other parametric model classifiers (nearest centroid and nearest centroid-adjusted), but is unable to correctly distinguish between samples in class "HA" and "HB".

Protein Data - Percentage Misclassification				
Classifier	Class Label			
—	HA	HB	M	G
1 Nearest Neighbor	76.99	50.70	13.14	13.14
3 Nearest Neighbors	83.09	53.52	15.49	14.08
5 Nearest Neighbors	74.17	46.94	14.08	13.14
Nearest Centroid	29.57	41.78	0	12.20
Nearest Centroid(adj)	30.04	25.35	3.75	21.59
PSVM	1.40	1.87	0.46	0
Parametric classifier	28.63	29.10	0	1.40

Table 1: Percentage of misclassification for various classifiers. Total number of samples = 213. PSVM parameters: $C = 100$, $\epsilon = 0.2$. The classification problem that was solved was one class vs the rest using leave-one-out cross-validation.

As is evident from the results, samples from classes "HA" and "HB" are difficult to distinguish. The parametric classifier fares better than all other classifiers but the PSVM.

4.2 Solar Flare Database

The Solar Flare Database [Newman et al., 1998] consists of 1066 data samples that are classified into 8 classes $(0, 1, 2, 3, 4, 5, 6, 7, 8)$, where the class label reflects the number of predicted solar flares.. We compared the parametric classifier to the PSVM for the "C-Flare", "M-Flare", and "X-Flare" sub-datasets. Because of the severe class bias(the ratio of number of "0 flares" samples to number of "≥ 1 flares" samples is approximately $4:1$ for "C-Flares", $29:1$ for "M-Flares", and $177:1$ for "X-Flares"), we converted the problem to a binary classification: 0 or ≥ 1. The classifiers were compared using a Receiver Operating Characteristics (ROC) Curve [Fawcett, 2006]. The classifiers are run for different threshold values, and the (false positive, true positive) points are plotted. False positive rate is the fraction of class 2 ("≥ 1 Flares") samples incorrectly classified as class 1 ("0 flares"). True positive rate is the fraction of class 1 samples that are correctly classified as class 1. Figure 4 shows the resulting ROC curves for the "C-flares". In the low false positive-low true positive rate region, the similarity-based parametric classifier performs better than the PSVM. In the upper right regions (high false positive, high true positive), both the classfiers perform comparably.

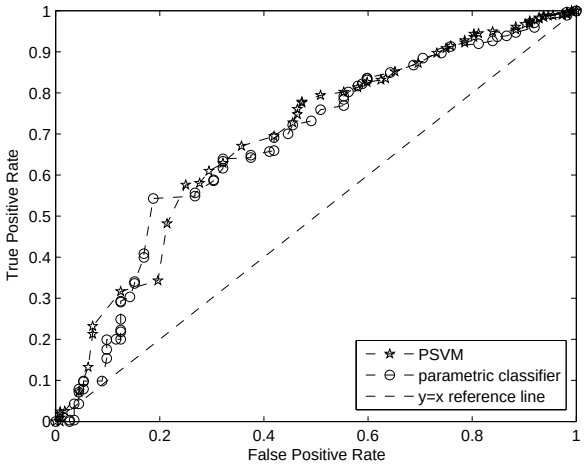

Figure 4: ROC curves for the parametric classifier and the PSVM for the "C labels" sub-dataset of the Solar Flare dataset. Each point represents the (false positive rate, true positive rate) tuple for a particular threshold values. The threshold values for the parametric classifier are

4.3 Varying number of dimensions

For this simulation, we created test and training samples from an inverted distribution (α_i/N for class i). The number of test samples was 100 and the number of training samples 10. The classifiers' error rates were compared over different dimensions (number of features). The parametric classifier (shown in a solid line in Figure 5) has almost half the error rate of the other classifiers for tests with high dimension values.

5 Discussion

The similarity-based parametric classifier proposed in this paper acts on pairwise similarities of data. Since it is independent of the similarity metric, and does not

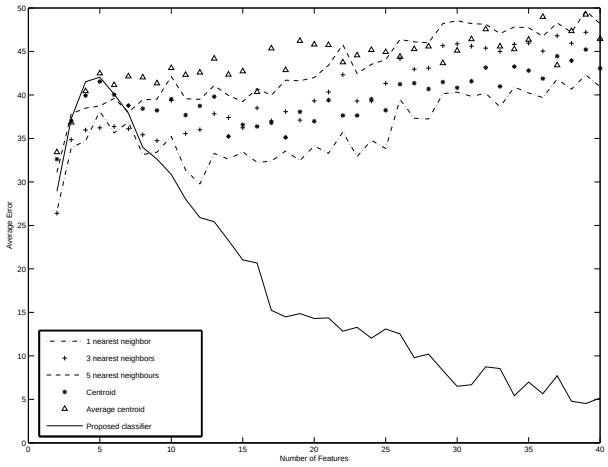

Figure 5: The Inverted distribution simulation. The parametric classifier is shown in a solid line.

need the actual data descriptions, it is a more general way of classifying data. The classifier performs well when the model of a unimodal distribution in similarity space is accurate. The parametric model is not, however, flexible enough to perform well in all casis. Nevertheless, it forms a good basis for a component of a more flexible mixture model.

6 Acknowledgements

I'd like to thank the Distributed Mentor Project and Professor Maya Gupta for an enriching summer.

References

Luca Cazzanti and Maya R. Gupta. Information-theoretic and set-theoretic similarity. In *Proc. International Symposium on Information Theory*. IEEE, October 2006.

Luca Cazzanti and Maya R. Gupta. Similarity based parametric classification.

Luca Cazzanti, Maya R. Gupta, and Anjali J. Koppal. Parametric models for similarity-based classification.

N. Cristianini and J. Taylor. *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, United Kingdom, 2000.

R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, United States of America, 2001.

R. P. W. Duin, E. Pekalska, and D. de Ridder. Relational discriminant analysis. *Pattern Recognition Letters*, 20:1175–1181, 1999.

T. Fawcett. An introduction to ROC analysis. *Pattern Recognition*, 27:861–874, 2006.

S. Hochreiter and K. Obermayer. Support vector machines for dyadic data. *Neural Comput.*, 18(6): 1472–510, June 2006.

E.T. Jaynes. *Probability Theory: The Logic of Science*. 1998.

Dekang Lin. An information-theoretic definition of similarity. *Proc. of the Intl. Conf. on Machine Learning*, 1998.

D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.

E. Pekalska and R. P. W. Duin. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23:943–956, 2002.

Amos Tversky. Features of similarity. *Psychological Review*, (84):327–352, 1977.