Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 41 (2008) 2289–2297

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/pr

Generative models for similarity-based classification

Luca Cazzanti[a], Maya R. Gupta[b,*], Anjali J. Koppal[c]

[a]Applied Physics Lab, Seattle, WA, USA
[b]University of Washington, Seattle, WA, USA
[c]University of California, Berkeley, CA, USA

Received 13 June 2007; received in revised form 26 October 2007; accepted 8 January 2008

Abstract

A maximum-entropy approach to generative similarity-based classifiers model is proposed. First, a descriptive set of similarity statistics is assumed to be sufficient for classification. Then the class-conditional distributions of these descriptive statistics are estimated as the maximum-entropy distributions subject to empirical moment constraints. The resulting exponential class-conditional distributions are used in a maximum a posteriori decision rule, forming the *similarity discriminant analysis* (SDA) classifier. Simulated and real data experiments compare performance to the k-nearest neighbor classifier, the nearest-centroid classifier, and the potential support vector machine (PSVM).

Keywords: Similarity; Maximum entropy; Discriminant analysis

1. Overview

Similarity-based classifiers classify a test sample x given only the pairwise similarities for the a test sample x and a set of training samples $\{x_i\}$, $i=1,\ldots,n$ [1–4]. The training samples' class labels are also given and denoted $\{y_i\}$ for $i=1,\ldots,n$. A similarity function s is a mapping that accepts two samples x, z from some sample space $x, z \in \mathscr{B}$, and returns a real number. That is, $s : \mathscr{B} \times \mathscr{B} \to \Omega$, where $\Omega \subset \mathbb{R}$. It is useful to think of the sample space $\mathscr{B}$ as an abstract space, such as "the space of all proteins," or "the space of all blogs." The similarity $s(x, z)$ is some judgement of how near samples x and z are, but similarities are not required to satisfy metric properties or any specific mathematical properties. The term "similarity-based classification" is also used when the given information is "dissimilarities," where a dissimilarity is a judgement of how far two samples are, but is not required to satisfy any specific mathematical properties.

Similarity-based learning is a useful approach when samples are described by categorical variables. For example, DNA is described as a sequence of bases, A, T, G, and G. Similarity-based learning is of course appropriate when the similarity or dissimilarity between samples is not a metric. For example, driving-times between any two given locations is not a metric, as it is often not symmetric and can violate the triangle inequality. Categorical variables and non-metric similarities/disssimilarities are common in fields such as bioinformatics, information retrieval, and natural language processing [1,4]. Also, similarity-based learning may be a better model than standard Euclidean-space learning for how humans classify, as psychologists have shown that metrics do not account for human judgements of similarity in complex situations [5–7]. Laub et al. have shown that non-metric similarities lead to information that can be useful for pattern recognition [8].

The simplest method for similarity-based classification is the nearest neighbor classifier, which determines the most-similar training sample to the test sample, and classifies the test sample as its most-similar neighbor's class. In fact, nearest neighbor classifiers using a tangent distortion [9] and a shape similarity metric [10] have both been shown to achieve lower error than metric k-NN for the MNIST character recognition task.

In this paper, we propose maximum-entropy generative similarity-based classifiers, which we term similarity discriminant analysis (SDA). We provide a review of the different

[*] Corresponding author.
E-mail addresses: luca@apl.washington.edu (L. Cazzanti), gupta@ee.washingtonedu (M.R. Gupta).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.01.005

approaches to similarity-based classification, and discuss how the proposed generative architecture ties together many of these approaches. We compare the resulting log-linear SDA classifier to the state-of-the-art in similarity-based classification on benchmark data sets and on an illustrative simulated example.

Reviews of similarity functions relevant for pattern recognition can be found in Refs. [11,12]. Many similarity functions are information theoretic, including information content similarity [13], mutual information similarity [14,15], residual entropy similarity [16], and the similarity defined by the compressability of one sample given another [17,18].

2. Review of similarity-based classifiers

Similarity-based classifiers make decisions based on the outputs of a pairwise similarity function s and an explicit description of the sample space $\mathscr{B}$ is not required. That is, the similarity function s can be treated as a black box by the classifier. If in fact the sample space $\mathscr{B}$ is a known set of categorical features, then naive Bayes, neural nets, and decision trees can also be applied.

2.1. Nearest neighbors

Experiments have shown that nearest neighbors can perform well on practical similarity-based classification tasks [2,9,10,19]. Condensed near-neighbor strategies replace the set of training samples for each class with a set of prototypes for that class. Usually the prototype set is an edited set of the original training samples (also called edited nearest neighbors), but the prototypes do not need to be from the original training set. Many authors have considered strategies for condensing near-neighbors for similarity-based classification to increase classification speed, decrease the required memory, and possibly attain better performance [3,20–23].

2.2. Nearest centroid

An extreme form of condensed near-neighbors is to replace each class's training samples by one prototypical sample, often called a *centroid*. The resulting "nearest-centroid" classifier can be considered a simple parametric model [20], but lacks a probabilistic structure. The nearest-centroid approach classifies x as the class

$$\hat{y} = \arg \max_{h=1,\ldots,G} s(x, \mu_h), \tag{1}$$

where μ_h is the representative centroid for the class h. A standard definition for the centroid of a set of training samples is the training sample that has the maximum total similarity to all the training samples of the same class [3,20]:

$$\mu_h = \arg \max_{\mu \in \mathscr{X}_h} \sum_{z \in \mathscr{X}_h} s(z, \mu), \tag{2}$$

where $\mathscr{X}_h$ is the set of training samples from class h.

The nearest-centroid classifier is analogous to the nearest-mean classifier in Euclidean space, which is the optimal Euclidean-based classifier if one assumes Gaussian class-conditional distributions and that each class covariance is the identity matrix.

2.3. Embed in Euclidean space

One can embed the training and test samples in an Euclidean space using multi-dimensional scaling [24], and then use standard statistical learning methods in the Euclidean feature space. More generally, the data can be embedded in a pseudo-Euclidean space for classification [2,25]. The embedding approach can also be used for clustering, for example [26] embed samples based on pairwise similarities in a low-dimensional Euclidean space by computing a multi-dimensional scaling solution subject to an entropy constraint. This results in an Euclidean embedding that maximizes the separation between clusters in a data set, while maintaining as much as possible the original pairwise similarity structure of the data. For most nonlinear embedding methods, classifying a new test sample requires re-computing the metric space embedding for all the data. If the underlying similarity relationships are not well represented by a metric distance, the embedding may be relatively high-dimensional, invoking the curse of dimensionality. On the other hand, the Procrustes approach of embedding the training samples in a low-dimensional Euclidean space may fail to sufficiently capture the similarity relationships between the samples [5,6,23,27].

2.4. Use the similarities to training samples as features

Similarity-based classification problems can be turned into standard Euclidean-based learning problems by treating the $n \times 1$ vector of similarities between a test sample and the n training samples as a feature vector [2,28,29]. Graepel et al. [28] propose a separating hyperplane classifier using this approach. Duin et al. [2,29] consider various standard learning techniques for this approach, including a regularized Fisher linear discriminant classifier for this space.

An issue with using the vector of similarities as a feature vector is that the feature vector size is equal to the number of training samples, causing Bellman's curse-of-dimensionality difficulties for learning [30]. As investigated by Pekalska et al. [2], one way to mitigate the problem that the dimension of the feature space is equal to the number of training samples is to regularize the covariance matrix when applying linear discriminant analysis (LDA) or quadratic discriminant analysis (QDA). Another approach they suggest for solving the dimensionality problem is to use only a subset of the training samples to define the feature vector. The results of Pekalska et al. show that, on average over their different experiments, linear classifiers built on the similarity vectors achieve similar errors as 1-nearest neighbor, except in cases of severe noise, where the 1-nearest neighbor has high error. Also, their similarity-based linear classifiers generally perform slightly better than first embedding the training samples in an metric space and then applying a linear classifier.

2.5. Generalized support vector machines

One approach to building support vector classifiers for similarity-based classification is to use the matrix of pairwise training sample similarities as a kernel. If the similarity matrix is symmetric and positive definite or conditionally positive definite as defined by Schölkopf [31], then the similarity matrix can be used as a kernel in standard support vector machines. A generalized support vector machine, the *potential support vector machine* (PSVM), has been developed that can be used with any similarity matrix [4,32]. Experiments in this paper compare the proposed generative approach to PSVM.

3. Maximum-entropy architecture for generative similarity-based classifiers

Let a test sample x be a realization of the random variable $X \in \mathscr{B}$, where $\mathscr{B}$ is the sample space. Let the similarity function be some function $s : \mathscr{B} \times \mathscr{B} \to \Omega$, where $\Omega \subset \mathbb{R}$. For ease of presentation, we assume that the sample space $\mathscr{B}$ is finite and discrete, such that the space of the possible pairwise similarities Ω is also finite and discrete; the continuous case is a trivial generalization. Let $Y \in \mathscr{G}$ be the random variable denoting the class label associated with x, where $\mathscr{G}$ is a finite set of G classes. Let $C(g, h)$ be the cost of classifying x as class g if the true class is h.

An optimal classifier is the theoretical Bayes classifier [30], which assigns a test sample x the class $\hat{y}$ that minimizes the expected misclassification cost

$$\hat{y} = \arg \min_{f=1,\ldots,G} \sum_{g=1}^{G} C(f, g) P(Y = g|x), \tag{3}$$

where $C(f, g)$ is the cost of classifying the test sample x as class f if the true class is g and is independent of x. In practice the distribution $P(g|x)$ is generally unknown.

We propose to create a generative model by assuming that the relevant information about X's class label is captured by some finite set $\mathscr{T}(X)$ of descriptive statistics. Some example choices for $\mathscr{T}(X)$ are

$$\mathscr{T}(X) = \{s(X, \mu_1), s(X, \mu_2), \ldots, s(X, \mu_G)\}, \tag{4}$$

$$\mathscr{T}(X) = \{s(X, X_1), s(X, X_2), \ldots, s(X, X_n)\}, \tag{5}$$

$$\mathscr{T}(X) = \{s(X, \mu_1), (s(X, \mu_1) - E[s(X, \mu_1)])^2, \ldots\},$$

$$\mathscr{T}(X) = \left\{ \sum_{z \in \mathscr{X}_1} s(X, z), \sum_{z \in \mathscr{X}_2} s(X, z), \ldots \right\}.$$

Given the assumption that the relevant information is contained in $\mathscr{T}(x)$, the classification rule (3) for a particular test sample x is to classify x as the class $\hat{y}$ that solves

$$\arg \min_{f=1,\ldots,G} \sum_{g=1}^{G} C(f, g) P(Y = g|\mathscr{T}(x)).$$

Using Bayes rule, this is equivalent to the problem

$$\arg \min_{f=1,\ldots,G} \sum_{g=1}^{G} C(f, g) P(\mathscr{T}(x)|Y = g) P(Y = g). \tag{6}$$

Next, we assume that each unknown class-conditional distribution $P(\mathscr{T}(x)|Y = g)$ has the same average value as the training sample data from class g. That is, we assume that the mth descriptive statistic $\mathscr{T}_m(x)$ has mean equal to the training sample mean:

$$E_{P(\mathscr{T}(x)|g)}[\mathscr{T}_m(X)] = \frac{1}{n_g} \sum_{z \in \mathscr{X}_g} \mathscr{T}_m(z), \tag{7}$$

for $g = 1, \ldots, G$ and $m = 1, \ldots, M$, and where n_g is the number of training samples in class g. Given these $M \times G$ constraints, there is some compact and convex feasible set of G class-conditional distributions $P(\mathscr{T}(x)|Y = g)$. A feasible solution will always exist because the constraints are based on the data.

As prescribed by Jaynes' principle of maximum entropy [33], we propose selecting the unique class conditional distributions that satisfy (7) and maximize entropy. Maximum-entropy distributions have the maximum possible uncertainty, and in that sense are the least assumptive solution. Given a set of moment constraints, the maximum-entropy solution is known to have exponential form [34]. Selecting the maximum-entropy distribution subject to constraints is analogous to the generative classifier *QDA*. QDA models each class-conditional distribution as a Gaussian [30], which is the maximum-entropy distribution given the class's training samples' empirical mean vector and covariance matrix.

For the gth class, solving the M constraints specified by (7) for the maximum-entropy distribution yields

$$\hat{P}(\mathscr{T}(x)|g) = \prod_{m=1}^{M} \gamma_{gm} e^{\lambda_{gm} \mathscr{T}_m(x)} \tag{8}$$

$$= \prod_{m=1}^{M} \hat{P}(\mathscr{T}_m(x)|g), \tag{9}$$

where the parameters $\{\lambda_{gm}, \gamma_{gm}\}$ have unique solutions which satisfy the constraints defined by (7). The equality given in (9) establishes that under the maximum-entropy assumption the statistics comprising the set $\mathscr{T}(x)$ are conditionally independent given the class label. Thus, one could equivalently describe this model as the maximum-entropy solution given the constraints

$$E_{P(\mathscr{T}_m(x)|g)}[\mathscr{T}_m(X)] = \frac{1}{n_g} \sum_{z \in \mathscr{X}_g} \mathscr{T}_m(z) \tag{10}$$

for $g = 1, \ldots, G$ and $m = 1, \ldots, M$, because the estimated $P(\mathscr{T}(x)|Y = h)$ is the same.

Substituting the maximum-entropy solution (8) into (6) creates the SDA classification rule: classify x as the class $\hat{y}$ which solves

$$\arg \max_{f=1,\ldots,G} \sum_{g=1}^{G} C(f, g) P(g) \prod_{m=1}^{M} \gamma_{gm} e^{\lambda_{gm} T_m(x)}. \tag{11}$$

The expression in (11) shows that under the SDA model the similarity statistics are conditionally independent given the class label. Although one does not expect this conditional independence to be strictly valid, the hypothesis is that it will be an effective model, just as the naive Bayes' model that features are independent is optimistic but useful.

3.1. SDA based on class centroids

As an illustrative example, we consider in more depth the two-class SDA classifier using the descriptive statistics given in (4) and zero–one misclassification costs (that is, $C(f, g) = 0$ if $f = g$ and $C(f, g) = 1$ otherwise). In this case, the SDA classification rule (11) is: choose class 1 if

$$\frac{\hat{P}(s(x,\mu_1)|Y=1)}{\hat{P}(s(x,\mu_1)|Y=2)}\frac{\hat{P}(s(x,\mu_2)|Y=1)}{\hat{P}(s(x,\mu_2)|Y=2)}\frac{P(Y=1)}{P(Y=2)} > 1. \qquad (12)$$

Applying the maximum-entropy solution for the class-conditional distributions, (12) becomes: choose class one if

$$\frac{\gamma_{11}e^{\lambda_{11}s(x,\mu_1)}}{\gamma_{21}e^{\lambda_{21}s(x,\mu_1)}}\frac{\gamma_{12}e^{\lambda_{12}s(x,\mu_2)}}{\gamma_{22}e^{\lambda_{22}s(x,\mu_2)}}\frac{P(Y=1)}{P(Y=2)} > 1. \qquad (13)$$

This SDA classifier uses the same information about the test sample as the nearest-centroid classifier, $s(x, \mu_1)$ and $s(x, \mu_2)$, but models the probability distribution of these statistics under the hypothesis that the sample belongs to class one or to class two. The probability distributions of the similarities capture the characteristic average deviation for each class and the average cross-class deviations. It is helpful to group the terms in (12) into the ratio term $\hat{P}(s(x, \mu_1)|Y = 1)/\hat{P}(s(x, \mu_1)|Y = 2)$ and the ratio term $\hat{P}(s(x, \mu_2)|Y=1)/\hat{P}(s(x, \mu_2)|Y=2)$. The first of these ratio terms establishes whether the similarity between the test sample x and the class one centroid μ_1 is better explained probabilistically by assuming x is from class one or from class two. Likewise, the second ratio term establishes whether the similarity $s(x, \mu_2)$ is better explained probabilistically by the hypothesis that x is from class one or from class two. For example, consider the case in which class one training samples are tightly clustered around μ_1, but class two training samples have on average low similarity to μ_2. Then even if a test sample x is slightly more similar to μ_1 such that $s(x, \mu_1) > s(x, \mu_2)$, SDA can learn that class one points should be very similar to μ_1, and can correctly classify x as a class two sample. This is analogous to the action of QDA in the case that class one's variance is very low compared to class two's variance.

A simpler method to generalize the nearest-centroid classifier to take into account the different distributions of each class would be to directly take into account the average similarity $\bar{s}_{gg}$ between class g training samples and a class g centroid μ_g, where

$$\bar{s}_{gg} = (1/n_g) \sum_{x_j \in \mathscr{X}_g} s(x_j, \mu_g).$$

Then, classify a test sample x as class $\hat{y}$ where

$$\hat{y} = \arg\max_g \frac{s(x, \mu_g)}{\bar{s}_{gg}}. \qquad (14)$$

This is analogous to the Gaussian-derived rule of classifying by the distances to the class means inversely weighted by each class's standard deviation: $\|x - \mu_g\|/\sigma_g$. We term the classifier given in (14) the *nearest-centroid adjusted* classifier.

4. Experiments

4.1. Perturbed centroids simulation

In this two-class simulation, each sample is described by d binary features such that $\mathscr{B}=\{0, 1\}^d$. For simplicity, given samples $x, z \in \mathscr{B}$, the similarity $s(x, z)$ is the number of features of x and z that are the same (called the counting, or Rao similarity). Each class is defined by one prototypical set of features (a centroid) denoted by c_1 and c_2, respectively. Every sample drawn from each class is a class centroid with some features possibly changed, according to a feature perturbation probability. For class one the perturbation probability is $p_1 = \frac{1}{3}$; for class two it is $p_2 = \frac{1}{30}$.

The simulations span several values for the feature dimensions d and are run several times to better estimate mean error rates. Each time the simulation is run, a random $c_1 \in \mathscr{B}$ and $c_2 \in \mathscr{B}$ are chosen uniformly from $\mathscr{B}$. Training and test samples are independently and identically distributed, and each class is equally likely. A training or test sample z drawn from class one starts out as $z = c_1$, but then for each $i = 1, \ldots, d$, z's ith feature is changed so that $z[i] \neq c_1[i]$ with probability p_1. Thus on average, np_1 features will be different than c_1's features. Likewise, a training or test sample v drawn from class two starts out as $v = c_2$, but then for each $i = 1, \ldots, d$, v's ith feature is changed so that $v[i] \neq c_2[i]$ with probability p_2.

For each run of the simulation and for each number of features considered, the neighborhood size k for k-NN is determined independently by leave-one-out cross-validation on the training set of 100 samples; the range of tested values for k is $\{1, 2, \ldots, 20, 29, 39, \ldots, 99\}$. The optimum k is then used to classify 1000 test samples. Similarly, the parameters for the PSVM classifier are cross-validated over the range of possible values $\varepsilon = \{0.1, 0.2, \ldots, 1\}$ and $C = \{1, 51, 101, \ldots, 951\}$.

Classifiers are trained on 100 training samples and tested on 1000 test samples per run; 20 runs are executed for a total of 20,000 test samples. The number of features d ranges from d=2 to 200 in the simulation, but the number of training samples is kept constant at 100, so that $d = 200$ is a sparsely populated feature space. The perturbed centroid simulation results are in Table 1. For each value of d, the lowest mean cross-validation error rate is in bold. Also in bold for each d are the error rates which are not statistically significantly different from the lowest mean error rate, as determined by the Wilcoxon signed rank test for paired differences, with a significance level of 0.05.

The performance of all classifiers increases as d increases. For large d, the feature space is sparsely populated by the training and test samples, which are segregated around their corresponding generating centroids. This leads to good classification performance for all classifiers. For small d, the feature space is densely populated by the samples, and the two classes con-

Table 1
Perturbed centroids experiment—misclasssification percentage for 20,000 test samples for various classifiers and numbers of features

d	SDA	Nearest centroid	k-NN	PSVM
2	35.13	23.47	**15.58**	**16.07**
4	**23.97**	22.54	12.05	**13.01**
8	**12.85**	14.07	**6.19**	**6.21**
12	**10.16**	11.50	4.26	**3.74**
25	**7.36**	11.49	3.49	**2.16**
40	**3.65**	8.79	2.79	**1.33**
50	**2.71**	7.94	2.31	**1.33**
75	2.56	7.83	2.27	**1.43**
100	2.05	5.92	2.16	**1.65**
125	1.67	6.21	1.96	**1.58**
150	**1.23**	4.86	1.44	**1.20**
175	**1.37**	4.28	1.60	**1.33**
200	**1.26**	4.20	1.38	**1.26**

Table 2
Solar flare leave-one-out error

Classifier	Error percentage
Solar flare data: C problem	
1 Nearest neighbor	25.61
3 Nearest neighbors	18.57
5 Nearest neighbors	18.67
Nearest centroid	57.88
Nearest-centroid adjusted	58.35
PSVM	16.79
SDA	17.07
Solar flare data: M problem	
1 Nearest neighbor	4.69
3 Nearest neighbors	3.38
5 Nearest neighbors	3.38
Nearest centroid	36.77
Nearest-centroid adjusted	96.72
PSVM	3.38
SDA	3.38
Solar flare data: X problem	
1 Nearest neighbor	0.56
3 Nearest neighbors	0.56
5 Nearest neighbors	0.47
Nearest centroid	7.97
Nearest-centroid adjusted	4.60
PSVM	0.47
SDA	0.47

siderably overlap, negatively affecting the classification performance.

With few exceptions the PSVM performs best. This is likely because the PSVM classifies a test sample based on a weighted sum of its similarities to the entire training set. In contrast, k-NN makes use of a subset of the training samples and thus has less information available to classify. SDA and nearest centroid also use less information. It is plausible that the ability to make use of all the similarity information in the training set and to optimally weight the similarities to the training samples gives the PSVM a performance advantage over the other techniques. However, in spite of this advantage, the results show that almost always the SDA classifier yields statistically equivalent performance to the PSVM, and in some cases matches or exceeds its results. Thus SDA produces good classification results using fewer information. This quality can be immensely useful when few training samples are available.

SDA performs better than nearest centroid. This shows that generative models based on the similarity of samples to local or global class centroids provide increased discriminative power over the non-generative centroid-based similarity models.

The similarity-space k-NN performs well, albeit not as well as the PSVM. Compared to SDA, k-NN performs better only at lower dimensions. Thus SDA seems to have an advantage when the feature space is very sparse. This powerful quality can be very useful in practical applications when there are few training samples available.

4.2. Solar Flare data

Like QDA, SDA based on one centroid per class has too much model bias to be appropriate for every classification problem, but will be useful when each class is well-modeled by a centroid, and in problems where a class is described by relatively little data. In this section, we reduce the model bias by using extra information to choose multiple centroids for one of the classes.

The Solar Flare Dataset in the UCI database [35] has 1066 samples describing an active region of the sun over a 24 h period, and each sample is described by 10 categorical features. Each sample also has a corresponding number describing how many C, M, and X flares were recorded during that time, ranging from 0 to 8 flares of each type. We treat the data set as three separate classification problems corresponding to the flare-types C, M, and X. Each flare-type's classification is framed as a two-class problem with class labels "no flares of that type" and "one or more flares of that type." Framing it as a two-class problem made it straightforward to compare with the PSVM classifier, which is naturally a two-class classifier. We used the counting similarity function for simplicity (that is, $s(x, z)$ is the number of features that x and z have in common). For the centroid-based classifiers we treat the "one or more flares of that type," class as being composed of eight subclasses, corresponding to the actual number of flares, and use a centroid for each of these subclasses. For the nearest-centroid and nearest-centroid adjusted classifiers, a test sample was then classified based on whether or not its nearest centroid belonged to the class "one or more flares." For SDA, a test sample x was classified as "no flares" if

$$P(Y = 0|x) > \sum_{k=1}^{8} P(Y = k|x),$$

where k is the number of flares counted for each of the original nine classes. The distribution of the training samples' classes is very uneven: 82.9% of the samples had no C flares, 96.6% of the samples had no M flares, and 99.5% of the samples had no X flares.

As detailed in Table 2, the top similarity-based classifiers for this problem are SDA and PSVM. Their comparative per-

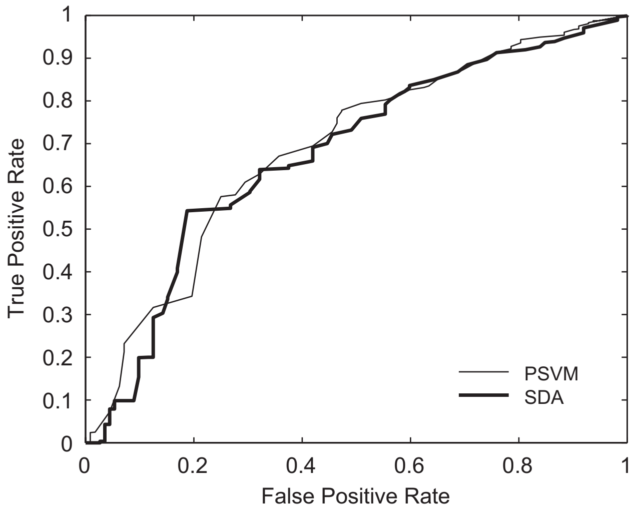

Fig. 1.

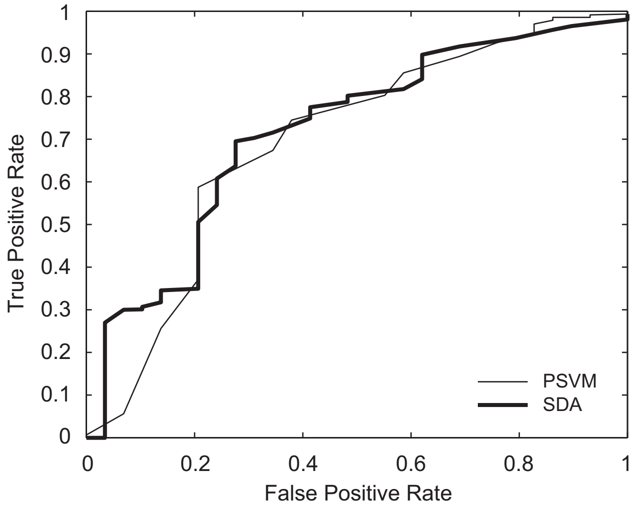

Fig. 2.

formance is shown in Figs. 1–3 for different achievable false positive and true positive rates. The different achievable rates shown correspond to different thresholds set for the SDA and PSVM class comparisons, where the thresholds were chosen to be equally spaced between the smallest range of thresholds that spanned the performance from 0% false positives to 100% false positives. The parameters for the PSVM (C, ε) were calculated by leave-one-out cross-validation on each training set with the default threshold value of 0. The PSVM cross-validation values tested were $\varepsilon \in \{0.1, 0.2, \ldots, 1\}$ and $C \in \{1, 11, \ldots, 1001\}$. Depending on the allowed false positive rate, SDA performs slightly better or worse than the PSVM.

4.3. Protein data

Many bioinformatics prediction problems are formulated in terms of pairwise similarities or dissimilarities. An example is the protein data set used by [4] and available from the authors. For this data set, pairwise dissimilarity values are calculated using a sequence alignment program, which counts the number of amino acids that differ between two sequences [36]. The sample space $\mathscr{B}$ is not enumerated, so it is not possible to use naive Bayes; classification must be done based only on the pairwise dissimilarity values. As done in Ref. [4], we performed all-against-one classification on the 213 proteins that had the class labels "HA" (72 samples), "HB" (72 samples), "M" (39 samples), and "G" (30 samples). We used the PSVM parameters for this problem that were cross-validated by Obermayer et al. in their paper proposing PSVM, reported to be $C = 100$, and $\varepsilon = 0.2$ [4]. The class priors were estimated to be the empirical probability of seeing a sample from each class, with Laplace correction [37]. The set of possible similarities Ω is needed to solve for the SDA parameters λ and γ, but was not directly available, so Ω was approximated as the set of empirical similarities that occurred in the training samples' similarity matrix. The nearest centroid, nearest-centroid adjusted, and SDA were implemented as simple mixture-models, where each of the four classes was represented by its own centroid.

Table 3 shows the leave-one-out misclassification error (the results were rounded for display). SDA performs better than the nearest-centroid model classifiers and k-NN classifier, but

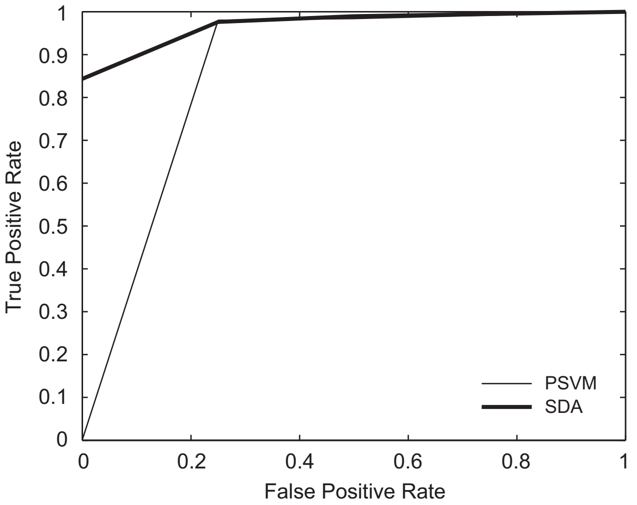

Fig. 3.

Table 3
Protein leave-one-out error

Classifier	Classification problem			
	HA v. all	HB v. all	M v. all	G v. all
1 Nearest neighbor	77	51	13	13
3 Nearest neighbors	85	55	15	14
5 Nearest neighbors	81	50	16	14
Nearest centroid	30	42	0	12
Nearest-centroid adjusted	30	25	4	22
PSVM	1	2	0	0
SDA	29	29	0	1

does poorly compared to the PSVM at distinguishing samples in class "HA" and "HB". The relatively low error rates on class "M" and class "G" suggest that those classes are well-modeled by a unimodal model for the features. The analysis is complicated because one SDA model is used for a combination of three protein classes.

5. SDA and other classification approaches

In the next subsections we discuss the relationship of SDA to QDA and to naive Bayes. We also show that SDA can be viewed as embedding the similarity data into a metric feature space.

5.1. SDA and naive Bayes

Consider the case that each sample is described by a set of binary features, that is, the sample space $\mathscr{B} = \{0, 1\}^N$ for some finite value of N. Given $z \in \mathscr{B}$, let the ith feature be denoted $z[i] \in \{0, 1\}$. Consider the SDA classifier using descriptive statistics $\mathscr{T}(x) = \{x[1], x[2], \ldots, x[N]\}$. In this case, there are N marginal class-conditional distributions to estimate for each class, as per (9). For each marginal distribution there are two unknowns, $P(x[i] = 0|Y = g)$ and $P(x[i] = 1|Y = g)$, and two constraints: the normalization constraint and the expectation constraint given in (7):

$$P(x[i] = 0|g) + P(x[i] = 1|g) = 1,$$
$$0 \times P(x[i] = 0|g) + 1 \times P(x[i] = 1|g) = \frac{1}{n_g} \sum_{z \in \mathscr{X}_g} I_{z[i]=1},$$

where I is the indicator function. These are the same constraints as for the naive Bayes classifier, which estimates class-conditional probabilities for the features as the product of probabilities for each feature. There is only one possible solution for the class-conditional distributions given these constraints. Thus, for binary features and $\mathscr{T}(x) = \{x[1], x[2], \ldots, x[N]\}$, SDA and naive Bayes are equivalent.

5.2. SDA and QDA

QDA is a generative classifier that models each class by a Gaussian class-conditional distribution in a d-dimensional Euclidean feature space [30]. In standard QDA a mean $\hat{u} \in \mathbb{R}^d$ and covariance matrix $\hat{\Sigma} \in \mathbb{R}^d \times \mathbb{R}^d$ are estimated from the training samples for each class, often using maximum likelihood (ML). The class prior $P(Y = g)$ may be either known or estimated, also with ML. The discriminant function $D_{QDA,g}(x)$ for the gth class is the logarithm of the Gaussian class-conditional distribution with the class prior term added:

$$\begin{aligned} D_{QDA,g}(x) = {} & -\tfrac{1}{2}(x - \hat{u}_g)^T \hat{\Sigma}_g^{-1}(x - \hat{u}_g) + \hat{u}_g^T \Sigma_g^{-1} x \\ & -\tfrac{1}{2} \log |\hat{\Sigma}_g| + \log P(Y = g). \end{aligned} \tag{15}$$

A test point $x \in \mathbb{R}^d$ is classified by determining which class-conditional Gaussian distribution is most likely to have generated the test point. The classification rule is written in terms of $D_{QDA,g}$ as

$$\hat{y} = \arg\max_g D_{QDA,g}(x), \tag{16}$$

where for simplicity (0, 1) misclassification costs are assumed.

QDA has a dual nature. It is both a Gaussian random vector model on continuous features and the maximum-entropy distribution subject to constraints on $\hat{u}$ and $\hat{\Sigma}$ based on the observed data [34]. In this respect, SDA is like QDA, because it too models a class-conditional generating distribution as the maximum-entropy distribution given moment constraints based on the data. However, SDA is more general than QDA. A major difference between QDA and SDA is that QDA is rooted in an Euclidean representation of the feature vectors, and the class-conditional Gaussian distributions directly model the probability of the test point x. SDA does not rely on the Euclidean assumption, and the class-conditional exponential distributions model some set of descriptive statistics $\mathscr{T}(x)$. Thus, in QDA the tested quantity is the probability of the d-dimensional test feature vector x, but in SDA the tested quantity is the probability of the descriptive statistics $\mathscr{T}(x)$ calculated as functions of the test sample $x \in \mathscr{B}$.

Another major difference between QDA and SDA arises when the Euclidean features are not Gaussian (e.g. they may be discrete or continuous and finite). In this case the QDA assumption that the class-conditional models be Gaussian is incorrect. One may still use (15) to model the gth class discriminant and estimate $\widehat{u_g}$ and $\widehat{\Sigma_g}$ from the data, but the model will be inherently biased. To avoid this problem, one could instead appeal to the dual nature of QDA, and estimate the class-conditional model as the maximum-entropy distribution subject to second order constraints. However, this approach still results in a Gaussian class-conditional model which is only an approximation of the true underlying generative distribution; as such, bias is still a problem. Thus, QDA is limited to class-conditional models for which the Gaussian assumption is a good approximation for the underlying distribution. On the other hand, SDA can seamlessly model both discrete and continuous variables. The exponential class-conditional probability models produced by SDA are applicable to continuous or discrete descriptive statistics of any order. The nature of the available data is an important consideration when determining the most appropriate classification approach. This underscores the value of developing similarity-based techniques that do not rely on metric assumptions and Euclidean feature spaces: SDA creates a new family of flexible classifiers for similarity-based learning that is more general than QDA.

5.3. Descriptive statistics form a derivative feature space

SDA can be interpreted as forming probability models over a feature space where the mth feature for sample x is given by $\mathscr{T}_m(x)$. This is another method to turn a similarity-based classification problem into a metric learning problem: define the features $\mathscr{T}(x)$, assume a metric for that space (such as Euclidean distance), and then one could apply any standard metric statistical classifier to this derived feature space. In fact, as discussed in Section 2.4, some proposed similarity-based classifiers train classifiers on the derived n-dimensional feature space made up of the descriptive statistics given in (5). However, as mentioned in Section 2.4, the curse of dimensionality limits the applicability of this strategy. For example, estimating covariance matrices for class-conditional QDA models in a high-dimensional derivative feature space is generally ill-posed. The simple and flexible SDA architecture based on class centroids does not suffer from as severely from the curse of dimensionality and retains the powerful interpretability properties of generative models.

6. Discussion

The contributions of this paper are threefold: proposing a maximum entropy architecture for generative similarity-based classifiers; relating many of the previously proposed similarity-based classifiers to this architecture, and establishing that the proposed SDA classifiers can have practical advantages in terms of performance, interpretability, and ease of use.

As with LDA and QDA, the power of a generative classifier depends on how well its model matches the true class-conditional distributions. Gaussian mixture model classifiers are an effective approach to Euclidean-based learning, and SDA using mixture models should have many of the same benefits. In this paper, SDA was implemented using mixtures for the Protein and Flare problem based on the subclass information that was available. This simple SDA mixture worked well for the Flare problem, but using one centroid for the subclass HA and HB for the Protein data set was not adequate. This suggests that the HA and HB proteins are not well-modeled by variations on one prototypical protein. The design of general SDA mixture models is an open research question, though the growing literature on finding prototypes for similarity-based classification (see Section 2.1) provides a starting point.

Our experiments showed PSVM to be a flexible and powerful similarity-based classifier, but we ran into computational difficulties enumerating the necessary $n \times n$ pairwise similarity matrix when the number of training samples n was large. Nearest neighbor similarity-based classifiers are flexible, but did not perform as well as the other classifiers in general, and in practice require additional cross-validation to determine the best number of neighbors. In comparison, the proposed SDA classifier computationally scales better than PSVM as n increases, and unlike the PSVM or k-NN, does not require cross-validation. Perhaps the most important advantage of SDA is that it creates probability estimates, which can be combined with priors, misclassification costs, and used with any number of classes or hierarchical models of classes.

Lastly, we note that the choice of similarity function is an important aspect of similarity-based classification. There are many open research questions about the interplay between similarity-based classifiers and similarity measures.

Acknowledgments

This work was supported in part by the United States Office of Naval Research, Code 321, Grant # N00014-05-1-0843, and in part by the CRA-W DMP program. We thank Warren Fox, John Miyamoto, and Santosh Srivastava for helpful discussions.

References

[1] M. Bicego, V. Murino, M. Pelillo, A. Torsello, Special issue on similarity-based classification, Pattern Recognition 39 (2006).

[2] E. Pekalska, P. Paclíc, R.P.W. Duin, A generalized kernel approach to dissimilarity-based classification, J. Mach. Learn. Res. (2001) 175–211.

[3] D.W. Jacobs, D. Weinshall, Y. Gdalyahu, Classification with nonmetric distances: image retrieval and class representation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (6) (2000) 583–600.

[4] S. Hochreiter, K. Obermayer, Support vector machines for dyadic data, Neural Comput. 18 (6) (2006) 1472–1510.

[5] A. Tversky, Features of similarity, Psychol. Rev. 84) (1977) 327–352.

[6] A. Tversky, I. Gati, Studies of similarity, in: E. Rosch, B. Lloyd (Eds.), Cognition and Categorization, Earlbaum, Hillsdale, NJ, 1978.

[7] I. Gati, A. Tversky, Weighting common and distinctive features in perceptual and conceptual judgments, Cognitive Psychol. 16 (1984) 341 –370.

[8] J. Laub, V. Roth, J.M. Buhmann, K.-R. Müller, On the information and representation of non-Euclidean pairwise data, Pattern Recognition 39 (2006) 1815–1826.

[9] P. Simard, Y.L. Cun, J. Denker, Efficient pattern recognition using a new transformation distance, Adv. Neural Inform. Process. Syst. 5 (1993) 50–68.

[10] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.

[11] S. Santini, R. Jain, Similarity measures, IEEE Trans. Pattern Anal. Mach. Intell. 21 (9) (1999) 871–883.

[12] B.S. Everitt, S. Rabe-Hesketh, The Analysis of Proximity Data, Arnold, London, 1997.

[13] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: Proceedings of the International Joint Conference on Artificial Intelligence, 1995, pp. 448–453.

[14] D. Hindle, Noun classification from predicate-argument structures, in: Proceedings of the ACL, 1990, pp. 268–275.

[15] D. Lin, An information-theoretic definition of similarity, in: Proceedings of the International Conference on Machine Learning, 1998.

[16] L. Cazzanti, M.R. Gupta, Information-theoretic and set-theoretic similarity, in: Proceedings of the IEEE International Symposium on Information Theory, 2006, pp. 1836–1840.

[17] C.H. Bennett, P. Gács, M. Li, P.M.B. Vitányi, W. Zurek, Information distance, IEEE Trans. Inform. Theory 44 (1998) 1407–1423.

[18] M. Li, X. Chen, X. Li, B. Ma, P.M.B. Vitányi, The similarity metric, IEEE Trans. Inform. Theory 50 (12) (2004) 3250–3264.

[19] S. Cost, S. Salzberg, A weighted nearest neighbor algorithm for learning with symbolic features, Mach. Learn. 10 (1) (1993) 57–78.

[20] D. Weinshall, D.W. Jacobs, Y. Gdalyahu, Classification in non-metric spaces, Adv. Neural Inform. Process. Syst. 11 (1999) 838–844.

[21] W. Lam, C. Keung, D. Liu, Discovering useful concept prototypes for classification based on filtering and abstraction, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1075–1090.

[22] E. Pekalska, R.P.W. Duin, P. Paclík, Prototype selection for dissimilarity-based classifiers, Pattern Recognition Lett. 39 (2006) 189–208.

[23] M. Lozano, J.M. Sotoca, J.S. Sánchez, F. Pla, E. Pekalska, R.P.W. Duin, Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces, Pattern Recognition 39 (2006) 1827–1838.

[24] G. Young, A.S. Householder, Discussion of a set of points in terms of their mutual distances, Psychometrika 3 (1938) 19–22.

[25] L. Goldfarb, A new approach to pattern recognition, Prog. Pattern Recognition 2 (1985) 241–402.

[26] J.M. Buhmann, T. Hofmann, A maximum entropy approach to pairwise data clustering, in: Proceedings of the International Conference on Pattern Recognition, vol. 2, October 1994, pp. 207–212.

[27] A. Tversky, J.W. Hutchinson, Nearest neighbor analysis of psychological spaces, Psychol. Rev. 93 (1986) 3–22.

[28] T. Graepel, R. Herbrich, K. Obermayer, Classification on pairwise proximity data, Adv. Neural Inform. Process. Syst. 11 (1999) 438–444.

[29] R.P.W. Duin, E. Pekalska, D. de Ridder, Relational discriminant analysis, Pattern Recognition Lett. 20 (1999) 1175–1181.

[30] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, New York, 2001.

[31] B. Schölkopf, The kernel trick for distances, Adv. Neural Inform. Process. Syst. 13 (2001).

[32] S. Hochreiter, M.C. Mozer, K. Obermayer, Coulomb classifiers: generalizing support vector machines via an analogy to electrostatic systems, Adv. Neural Inform. Process. Syst. 15 (2003) 545–552.

[33] E.T. Jaynes, On the rationale for maximum entropy methods, Proc. IEEE 70 (9) (1982) 939–952.

[34] T. Cover, J. Thomas, Elements of Information Theory, Wiley, New York, 1991.

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

[36] T. Hofmann, J.M. Buhmann, Pairwise data clustering by deterministic annealing, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1) (1997).

[37] E.T. Jaynes, Probability Theory: The Logic of Science, Cambridge University Press, Cambridge, 2003.

About the Author—LUCA CAZZANTI received his B.S. and M.S. in Electrical Engineering from the University of Wisconsin, Madison in 1996 and 1998, respectively. In 2007 he completed his Ph.D. in the Department of Electrical Engineering at the University of Washington.

About the Author—MAYA R. GUPTA completed her Ph.D. in the Department of Electrical Engineering of Stanford University in 2003, and has since been an assistant professor in the Department of Electrical Engineering at the University of Washington. She received the United States Office of Naval Research Young Investigator Award in 2007.

About the Author—ANJALI J. KOPPAL received a B.A. in computer science from the University of California, Berkeley in 2007.