Local Similarity Discriminant Analysis

Luca Cazzanti LUCA@APL.WASHINGTON.EDU
Applied Physics Lab, University of Washington, Seattle WA 98195

Maya R. Gupta GUPTA@EE.WASHINGTON.EDU
Dept. of Electrical Engineering, University of Washington, Seattle WA 98195

Abstract

We propose a local, generative model for similarity-based classification. The method is applicable to the case that only pairwise similarities between samples are available. The classifier models the local class-conditional distribution using a maximum entropy estimate and empirical moment constraints. The resulting exponential class conditional-distributions are combined with class prior probabilities and misclassification costs to form the *local similarity discriminant analysis* (local SDA) classifier. We compare the performance of local SDA to a non-local version, to the local nearest centroid classifier, the nearest centroid classifier, k-NN, and to the recently-developed potential support vector machine (PSVM). Results show that local SDA is competitive with k-NN and the computationally-demanding PSVM while offering the advantages of a generative classifier.

1. Similarity-based Classification

Similarity-based learning methods make inferences based only on pairwise similarities or dissimilarities between a test sample and training samples and between pairs of training samples [Bicego et al., 2006, Pekalska et al., 2001, Jacobs et al., 2000, Hochreiter & Obermayer, 2006]. The term *similarity-based learning* is used whether the pairwise relationship is a similarity or a dissimilarity. The similarity/dissimilarity function is not constrained to satisfy the properties of a metric. Similarity-based learning can be applied when the test and training samples are not described as points in a metric space. This occurs when the samples are described as feature vectors but the relevant relationship between samples is a similarity or dissimilarity function that does not obey the mathematical rules of a metric. Another case is when the samples are not described as feature vectors at all, but pairwise similarity or dissimilarity information is available. Such similarity-based learning problems arise naturally in bioinformatics, information retrieval, natural language processing, and with geospatial data. Similarity-based learning is also a model for how humans learn, based on psychological evidence that metrics do not account for human judgements of similarity in complex situations [Tversky, 1977, Tversky & Gati, 1978, Gati & Tversky, 1984].

A nearest-centroid classifier is a simple model-based approach to similarity-based classification [Weinshall et al., 1999], and parallels work by psychologists suggesting that humans learn based on similarity to prototypical samples [Rosch, 1973]. In recent work, we generalized the nearest-centroid classifier by estimating a generating distribution for the similarity between the test sample and the class centroids [Gupta et al., 2007]. We termed this generative similarity-based classifier *similarity discriminant analysis* (SDA), because like quadratic discriminant analysis (QDA), SDA estimates class-conditional distributions that are the maximum entropy distributions given empirical statistics. An advantage of such model-based classifiers is their interpretability. However, classifying based on one centroid for each class has too much model bias to be a flexible general-purpose classifier. In the metric learning case, a common solution is to form a more flexible Gaussian mixture model classifier [Hastie et al., 2001]. A disadvantage to the mixture model approach is that parameters must be estimated and the classifier can have relatively high variance due to its sensitivity to some of these parameters, in particular to the estimated number of mixture components. Further, to form a mixture model, clustering must be done

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

in similarity-based space, which may be ill-posed if the similarity or dissimilarity function does not satisfy symmetry or other metric properties.

In this paper we propose a different approach to reduce the model bias: we apply SDA to a local neighborhood about the test sample. The resulting *local SDA* classifier trades-off model bias and estimation variance depending on the neighborhood size, which we cross-validate. Local classification algorithms have traditionally been weighted voting methods, including classifying with local linear regression, which can be formulated as a weighted voting method [Hastie et al., 2001]. In 2000, a local nearest-means classifier was proposed [Mitani & Hamamoto, 2000, Mitani & Hamamoto, 2006], although their definition of the neighborhood was the union of the k nearest neighbors from each class. More recently, it was proposed to apply a support vector machine to the k nearest neighbors of the test sample [Zhang et al., 2006].

Prior work in similarity-based classification is reviewed in the next section. In Section 3 we introduce the local SDA classifier. Experiments with real data and with simulations in Section 4 compare local SDA's effectiveness against other similarity-based classifiers. The paper closes with a summary and some open questions.

2. Prior Research in Similarity-based Classification

A simple similarity-based classifier is the nearest neighbor classifier, which classifies the test sample as the class of the test sample's most-similar neighbor. Similarity-based nearest neighbor classifiers have achieved very low error on MNIST by using a tangent distortion [Simard et al., 1993], and a shape similarity metric [Belongie et al., 2002]. Experiments using similarity-based nearest neighbor classifiers have shown it to be an effective general-purpose approach in practice [Cost & Salzberg, 1993] and [Pekalska et al., 2001].

Another similarity-based classifier is the 'nearest centroid' classifier, which can be considered a simple parametric model [Weinshall et al., 1999] Let $s(x, z)$ be the similarity between a sample x and a sample z, and let there be a finite set of classes $1, 2, \ldots, G$. 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 [Weinshall et al., 1999, Jacobs et al., 2000]:

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

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

A different approach to similarity-based classification is to embed the training and test samples in an Euclidean space using multi-dimensional scaling, and then use standard statistical learning methods in the Euclidean feature space. Or, the data can be embedded in a pseudo-Euclidean space [Goldfarb, 1985, Pekalska et al., 2001]. One disadvantage to this approach is that if the similarity function does not satisfy the metric properties, then no embedding may be appropriate; for example, forcing samples related by an asymmetric similarity function to be classified using Euclidean distance may be suboptimal. Also, if the underlying similarity relationships are not well represented by a metric distance, a low-error embedding may be relatively high-dimensional, causing curse of dimensionality problems for the subsequent classification. A second disadvantage is that classifying each test sample requires re-computing the embedding based on the test and training samples. This is a problem if all of the test data or training data are not available at one time.

Another general approach to similarity-based classification is to treat the $n \times 1$ vector of similarities between a test sample and the n training samples as a feature vector [Graepel et al., 1999, Duin et al., 1999, Pekalska et al., 2001]. Graepel et al. [Graepel et al., 1999] created a separating hyperplane classifier using this approach. Duin et al. [Duin et al., 1999, Pekalska et al., 2001] considered a regularized Fisher linear discriminant classifier for this n-dimensional space. A serious disadvantage to this general approach is that the dimensionality of the classification problem is equal to the number of training samples n, which may be arbitrarily high.

If the $n \times n$ pairwise similarity matrix between the training samples is symmetric and positive definite, it can be used as a kernel for a support vector machine. A more general approach to support vector machines given pairwise similarities is the *potential support vector machine* (PSVM), which can be used with any similarity matrix [Hochreiter et al., 2003, Hochreiter & Obermayer, 2006]. However, the PSVM also requires enumerating $n \times n$ matrices, which is computationally infeasible given large n. Additionally, the PSVM requires the cross-validation of two parameters. Another support vector machine approach is the SVM-KNN, which uses a similarity kernel but only applies

the support vector machine to the k nearest neighbors of the test sample, which may make SVM's more computationally feasible in certain large datasets [Zhang et al., 2006].

3. Local SDA

We propose a new similarity-based classifier, local similarity discriminant analysis (local SDA). Local SDA is a log-linear generative classifier that models the probability distribution of the similarity between the test sample and the class means of the test sample's neighborhood. In this section we explain how the local SDA classifier follows from the optimal Bayes classifier by a few simple assumptions. Let all test and training samples come from some abstract space of possible samples $\mathcal{B}$, such as the set of all DNA sequences, or the set of all French political blogs, or customers who call a technical support hotline. The Bayes classifier assigns a test sample $x \in \mathcal{B}$ to the class $\hat{y}$ that minimizes the expected misclassification cost [Hastie et al., 2001],

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

where $C(f,g)$ is the cost of classifying the test sample x as class f if the true class is g. In practice the distribution $P(Y = g|x)$ is generally unknown, and thus the Bayes classifier of (3) is an unattainable ideal.

Suppose one can evaluate a relevant similarity function $s : \mathcal{B} \times \mathcal{B} \to \Omega$, where $\Omega \subset \mathcal{R}$ [1]. For simplicity, we assume that Ω is a finite discrete space, but it is straightforward to generalize to continuous Ω. Let $X \in \mathcal{B}$ be random test sample with random class label Y, and let $x \in \mathcal{B}$ denote the realization of X. The local SDA classifier model is that all of the relevant information about classifying X depends only on the k nearest (most similar) training samples to X, where the neighborhood size k is learned by cross-validation.

The second assumption of the local SDA classifier model is that the information about X's class label depends only on the similarity between X and its neighborhoods' class means. Let μ_g be the gth class centroid defined by

$$\mu_g = \arg \max_{\mu \in \mathcal{X}_g} \sum_{z \in \mathcal{X}_g} s(z, \mu), \qquad (4)$$

where $\mathcal{X}_g$ is the subset of the k nearest neighbors that are from class g (we will discuss the degenerate case that $\mathcal{X}_g$ is the empty set later).

[1] Some similarity measures also depend on context; for those measures $s : \mathcal{B} \times \mathcal{B} \times \mathcal{C} \to \Omega$, where $\mathcal{C}$ is a space representing the context.

Given the assumption that the relevant information about X's class label is $\{s(X, \mu_g)\}$ for $g = 1, \ldots, G$ and given a particular test sample x, the classification rule (3) can be re-stated: classify x as class $\hat{y}$ that solves

$$\arg \min_{f=1,\ldots,G} \sum_{g=1}^{G} C(f,g) P(Y = g|s(x,\mu_1), \ldots, s(x,\mu_G))$$

Because only the minimizer is of interest, the posterior $P(Y = g|s(x,\mu_1), \ldots, s(x,\mu_G))$ can be replaced by $P(s(x,\mu_1), \ldots, s(x,\mu_G)|Y = g)P(Y = g)$, which is the probability of seeing a particular set of similarities between the test sample x and the G local class centroids $\{\mu_1, \mu_2, \ldots, \mu_G\}$ given that x is a class g sample, and the class priors $P(Y = g)$ are now explicit.

We must estimate the class-conditional distribution $P(s(x,\mu_1), \ldots, s(x,\mu_G)|Y = g)$. To determine a unique and reasonable estimate, we constrain the expectation of each $s(X, \mu_g)$ with respect to each of the G unknown class-conditional distributions such that

$$E_{P(s(x,\mu_1),\ldots,s(x,\mu_G)|Y=g)}[s(X,\mu_h)] = \frac{1}{n_g} \sum_{z \in \mathcal{X}_g} s(z, \mu_h), \qquad (5)$$

for each g, h where n_g is the number of training samples of class g. That is, (5) specifies G constraints on each of the G unknown class-conditional distributions $P(s(x,\mu_1), \ldots, s(x,\mu_G)|Y = g)$, for a total of $G \times G$ constraints. Given these constraints, there is some compact and convex feasible set of G class-conditional distributions. (A feasible solution will always exist because the constraints are a linear combination of the data.)

Applying Jaynes' principle of maximum entropy [Jaynes, 1982], we estimate each class-conditional distribution as the maximum entropy distribution that satisfies the G constraints specified by (5). Given a set of constraints that could be satisfied by multiple (possibly infinite) distributions, the distribution that has the maximum entropy is least assumptive in that it is the distribution closest to the uniform distribution in terms of relative entropy. Due to the convexity of the entropy function, the maximum entropy distribution is always unique if the constraints specify a convex set of feasible distributions. The maximum entropy estimation approach also links local SDA to QDA: in metric-learning, the QDA classifier classifies based on Gaussian class-conditional distributions, which can be derived as the maximum entropy solutions given the empirical training sample covariance matrix and mean vector. For general results about maximum entropy solutions see, for example, [Cover & Thomas,

1991, Van Campenhout & Cover, 1981, Friedlander & Gupta, 2006].

Given a set of moment constraints such as given in (5), it is easy to show that the maximum entropy solution has an exponential form,

$$\hat{P}(s(x,\mu_1),\ldots,s(x,\mu_G)|Y=g) = \gamma_g e^{\left(\sum_{h=1}^{G}\lambda_{gh}s(x,\mu_h)\right)}, \quad (6)$$

where $\{\gamma_g, \lambda_{g1}, \lambda_{g2}, \ldots, \lambda_{gG}\}$ are a unique set of scalars that ensures that the constraints specified by (5) are satisfied and that estimated distribution is non-negative and normalized.

The maximum entropy estimate given in (6) can be re-written as a product of exponential distributions,

$$\begin{aligned}\hat{P}(s(x,\mu_1),\ldots,s(x,\mu_G)|Y=g) &= \prod_{h=1}^{G}\gamma_{gh}e^{\lambda_{gh}s(x,\mu_h)} \\ &\triangleq \prod_{h=1}^{G}\hat{P}(s(x,\mu_h)|Y=g). \quad (7)\end{aligned}$$

where $\prod_h \gamma_{gh} = \gamma_g$, and we have defined the exponential distributions $\hat{P}(s(x,\mu_h)|Y=g)$ in (7). Note that each distribution $\hat{P}((s(x,\mu_h)|Y=g)$ is also the maximum entropy distribution that satisfies the constraint

$$E_{P(s(x,\mu_h)|Y=g)}[s(X,\mu_h)] = \frac{1}{n_g}\sum_{z\in\mathcal{X}_g} s(z,\mu_h). \quad (8)$$

Thus, the joint maximum entropy estimate $\hat{P}(s(x,\mu_1),\ldots,s(x,\mu_G)|Y=g)$ is the product of maximum-entropy marginal distributions, which implies that $s(x,\mu_g)$ is conditionally independent of $s(x,\mu_h)$ given x's class label for all $g,h \in \{1,\ldots,G\}$.

Substituting the estimated probability distribution from (6) into (3) yields the local SDA classification rule: classify x as the class $\hat{y}$ which solves

$$\arg\max_{f=1,\ldots,G}\sum_{g=1}^{G}C(f,g)\left(\prod_{h=1}^{G}\gamma_{gh}e^{\lambda_{gh}s(x,\mu_h)}\right)\hat{P}(g) \quad (9)$$

where the parameters $\{\lambda_{gh},\gamma_{gh}\}$ must satisfy the $G\times G$ constraints specified by (5) and the normalization constraint. The class priors $\hat{P}(g)$ are estimated from the k nearest-neighbors; we use a Bayesian estimate, also known as Laplace correction [Jaynes, 2003]. One can solve for the parameters $\{\lambda_{gh},\gamma_{gh}\}$ directly (as we do for our experiments) or use a standard optimization method to find the maximum entropy distribution given the moment and normalization constraints.

If the number of neighbors $k \leq 3$, then the local SDA model is difficult to estimate; in this case the local SDA classifier reverts to the local nearest centroid. This strategy enables local SDA to gracefully handle small k. If there are no neighbors from the gth class in the neighborhood of the test sample X such that $\mathcal{X}_g$ is the empty set, then the probability that X is from class g is considered to be zero, and class g is not included in the rule (9).

4. Experiments

We compare the local SDA classifier to a global SDA (using all training samples to estimate the class-conditional distributions), to the nearest centroid and to a local-version of the nearest centroid classifier, to k-NN, and to the PSVM [Hochreiter & Obermayer, 2006].

4.1. Simulations: Multimodal Perturbed Centroids

To further analyze and compare similarity-based classifiers, we consider a two-class simulation, where each class is composed of two prototypical samples and perturbations of those prototypical samples. To generate pairwise similarities, each sample is drawn randomly from $\mathcal{B} = \{0,1\}^d$. In the first set of results, the pairwise similarity $s(x,z)$ between samples $x,z \in \mathcal{B}$ is the counting similarity, which is the number of features in agreement between x and z.

Each class is characterized by 2 prototypical samples, c_{11}, c_{12} for class one, and c_{21}, c_{22} for class two. Each time the simulation is run, the centroids $c_{11}, c_{12}, c_{21}, c_{22}$ are drawn independently and identically using a uniform distribution over $\mathcal{B}$.

Every sample drawn from each class is a perturbed version of one of the two class prototypes, where the class labels are drawn independently and identically with probability 1/2. A training or test sample z drawn from class one is randomly selected to be $z = c_{11}$ or $z = c_{12}$ with probability 1/2, and then for each $i = 1,\ldots,n$, z's ith feature is flipped (0 changed to 1 or 1 changed to 0) with probability 1/3. Thus on average, a randomly drawn sample based from class 1 will have $n/3$ features that are different from one of the class prototype's features. Likewise, a training or test sample v drawn from class two starts out as $v = c_{21}$ or $v = c_{22}$ with probability 1/2, but then for each $i = 1,\ldots,n$, v's ith feature is flipped (0 changed to 1 or 1 changed to 0) with probability 1/30.

The number of features n ranges from $n = 2$ to $n = 200$ in the simulation, but the number of training samples is kept constant at 100, so that $n = 200$ is a sparsely populated feature space. For each run of the sim-

Table 1. Perturbed centroid simulations: percentage of misclassification error on test set.

	% Perturbed Centroids Simulation Misclassification Error (neighborhood size)					
# Features	Local SDA	Local Nearest Centroid	SDA	Nearest Centroid	k-NN	PSVM
2	**26.41**	**26.41**	47.52	38.98	**26.41**	27.16
4	13.80	13.68	34.77	34.84	**13.26**	17.18
8	**9.23**	9.25	29.32	26.77	9.29	12.62
12	**5.61**	6.47	31.20	27.05	6.25	8.72
25	**3.11**	4.37	28.75	25.90	4.03	4.08
40	2.88	4.25	30.84	28.23	3.94	**2.21**
50	2.94	4.89	27.77	30.12	4.35	**1.77**
75	2.04	3.21	26.38	27.74	2.75	**0.95**
100	2.21	3.03	25.39	24.58	2.60	**1.52**
125	2.46	2.96	25.51	24.83	2.68	**1.59**
150	1.55	1.80	25.00	26.55	1.76	**1.00**
175	1.93	2.38	25.32	21.40	2.02	**1.29**
200	1.44	1.61	23.87	19.28	1.49	**1.10**

ulation and for each number of features considered, the neighborhood size k is determined by leave-one-out cross-validation on the training set. The optimum k is then used to classify 1000 test samples with local SDA. The same procedure is used to optimize k for the k-nearest neighbor classifier and for the local nearest centroid classifier. Each simulation was run twenty times, and the mean error rates for the resulting $20,000$ test samples are given in Table 3.

The results show that local SDA is consistently the best model-based classifier, and for most numbers of features performs slightly better than k-NN. The PSVM does slightly worse than the local classifiers for 12 features and less, but is the best classifier for 40 dimensions and higher. For local SDA and local nearest centroid, the larger number of features are challenging for this problem because they choose one of the training samples as a class prototype. Because $n/3$ and $n/30$ features are on average flipped from the true class 1 and class 2 centroids respectively, as the number of features n increases the training samples become worse choices for the class centroids. Still, the local model-based methods perform relatively well even for 200 features.

Figure 1 shows an example of the performance over the neighborhood size for local SDA and local nearest centroid for 8 features.

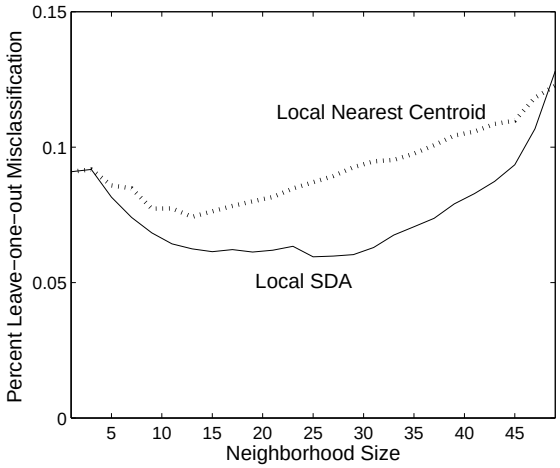

Figure 1. Plot of leave-one-out misclassification error versus the neighborhood size for the perturbed centroids simulation with 12 features.

4.2. Protein Data

Many bioinformatics prediction problems are formulated in terms of pairwise similarities or dissimilarities. For this example dataset, pairwise dissimilarity values were calculated using a sequence alignment program, which counted the number of amino acids that differ between two sequences [Hoffmann & Buhmann, 1997].

The sample space $\mathcal{B}$ is not enumerated, so classification must be done based only on the pairwise dissimilarity values. As in [Hochreiter & Obermayer, 2006], we used the 213 proteins with class labels, "HA" (72 samples), "HB" (72 samples), "M" (39 samples) , and "G" (30 samples). 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 neighborhood sizes for the local classifiers were obtained by cross-validation on each training set using the set of candidate neighborhood sizes $\{1, 2, \ldots 100 \ldots, 100, 110, 120, \ldots, 200\}$

Table 2 shows the percentage of leave-one-out misclassification error for the different similarity-based clas-

sifiers, excluding the PSVM which is specified as a binary classifer. Guessing that all samples were from the most prevalent class would yield a 66.2% error rate. The simple one-centroid per class model of SDA achieves half that error, and works better than the more flexible local nearest centroid classifier. Local SDA, local nearest centroid and k-NN all have the same free parameter, the neighborhood size k, and local SDA is seen to be best suited to this problem.

4.3. Voting

The UCI voting dataset [Newman et al., 1998] records the voting record of 435 members of the US House of Representatives on 16 bills. The binary classification problem is to predict each member's political party affiliation given the voting records. Each of the 16 votes is either a yes, a no, or "neither," so there are 16 features which can each take on 3 possible values. This classification problem can be treated as a similarity-based classification problem by applying a similarity function to the trinary feature space. We consider two different similarity functions: the counting similarity between two US representatives is the number of times they voted identically; and the second similarity considered is the value difference metric (VDM), which is a dissimilarity measure for similarity-based classification that has been shown to be very effective with nearest-neighbor classifiers [Stanfill & Waltz, 1986, Cost & Salzberg, 1993].

The leave-one-out cross-validation error for the different similarity-based classifiers is compared in Table 3, as well as the performance of naive Bayes applied to the feature space (no similarity used). The neighborhood sizes for the local classifiers were cross-validated using the set of candidate neighborhood sizes $\{1, 2, \ldots, 100\}$. The PSVM parameters were cross-validated using the sets of candidate parameters $C = \{1, 51, 101, \ldots 941\}$ and $\epsilon = \{0.1, 0.2, \ldots 1\}$.

The PSVM, with its two degrees of freedom, does slightly better than the other classifiers on this problem. All of the classifiers do better with the VDM similarity than with the counting similarity. The local classifiers perform similarly and choose similarly sized neighborhoods.

5. Consistency

Generative classifiers with a finite number of model parameters, such as QDA or SDA, will not asymptotically converge to the Bayes classifier due to the model bias. In this section we show that, like k-NN, the local SDA classifier is consistent such that its expected classification error $E_X[L]$ converges to the Bayes error rate L^* under the usual asymptotic assumptions that the number of training samples $n \to \infty$, the neighborhood size $k \to \infty$, but that the neighborhood size grows relatively slowly such that $k/n \to \infty$.

Let the similarity function $s : \mathcal{B} \times \mathcal{B} \to \Omega$, where $\Omega \subset \mathbb{R}$ is discrete and let the largest element of Ω be termed s_{max}. Let X be a test sample and let the training samples $\{X_1, X_2, \ldots, X_n\}$ be drawn identically and independently. Re-order the training samples according to decreasing similarity and label them $\{Z_1, Z_2, \ldots Z_n\}$ such that Z_k is the kth most similar neighbor of X.

The consistency theorem will use the following lemma:

Lemma 1: Suppose $s(x, Z) = s_{max}$ if and only if $x = Z$ and $P(s(x, Z) = s_{max}) > 0$ where Z is a random training sample. Then $P(s(x, Z_k) = s_{max}) \to 1$ as $k, n \to \infty$ and $k/n \to 0$.

Proof: The proof is by contradiction and is similar to the proof of Lemma 5.1 in [Devroye et al., 1996]. Note that $s(x, Z_k) \neq s_{max}$ if and only if

$$\frac{1}{n} \sum_{i=1}^{n} I_{\{s(x,Z_i)=s_{max}\}} < \frac{k}{n}, \tag{10}$$

because if there are less than k training samples whose similarity to x is s_{max}, the similarity of the kth training sample to x cannot be s_{max}. The left-hand side of (10) converges to $P(s(x, Z) = s_{max})$ as $n \to \infty$ with probability one by the strong law of large numbers, and by assumption $P(s(x, Z) = s_{max}) > 0$. However, the right-hand side of (10) converges to 0 by assumption. Thus, assuming $s(x, Z_k) \neq s_{max}$ leads to a contradiction in the limit. Therefore, it must be that $s(x, Z_k) = s_{max}$.

Theorem: Assume the conditions of Lemma 1. Define L to be the probability of error for test sample X given the training sample and label pairs $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$, and let L^* be the Bayes error. If $k, n \to \infty$ and $k/n \to 0$, then for the local SDA classifier $E_X[L] \to L^*$.

Proof: By Lemma 1, $s(x, Z_i) = s_{max}$ for $i \leq k$ in the limit as $n \to \infty$, and thus in the limit the centroid μ_h of the subset of the k neighbors that are from class h must satisfy $s(x, \mu_h) = s_{max}$ for every class h. Then the constraint (8) on the expected value of the class-conditional similarity is

$$E_{P(s(x,\mu_h)|Y=g)}[s(X, \mu_h)] = s_{max}, \tag{11}$$

which is solved by the pmf $P(s(x, \mu_h)|Y = g) = 1$ if $s(x, \mu_h) = s_{max}$, and zero otherwise. Thus the local

Table 2. Protein classification problem: percentage of leave-one-out misclassification error on 213 samples for the four-class classification problem. Neighborhood sizes are shown in parentheses.

% Protein Misclassification Error (neighborhood size)				
Local SDA	Local Nearest Centroid	SDA	Nearest Centroid	k-NN
8.92 (120)	37.09 (140)	29.58	41.78	20.66 (79)

Table 3. Voting classification problem: percentage of leave-one-out misclassification error on 435 samples with counting similarity. Neighborhood sizes are shown in parentheses.

	% Voting Misclassification Error (neighborhood size)						
	Local SDA	Local Nearest Centroid	SDA	Nearest Centroid	k-NN	PSVM	Naive Bayes
Counting Similarity	6.67 (4)	6.90 (3)	11.72	12.18	6.90 (4)	**4.37**	10.11
VDM Similarity	5.75 (3)	5.75 (3)	10.57	9.66	**4.37** (4)	**4.37**	10.11

SDA classifier (9) becomes

$$\hat{y} = \arg \max_{g=1,\ldots G} \hat{P}(Y = g).$$

The estimated probability of each class $\hat{P}(Y = g)$ is calculated for the neighborhood using a Bayesian or maximum likelihood estimate, and for either estimate $\hat{P}(Y = g) \to P(Y = g)$ as $k \to \infty$ with probability one by the strong law of large numbers. Thus the local SDA classifier converges to the Bayes classifier, and the local SDA average error $E_X[L] \to L^*$.

6. Discussion

We have proposed a flexible, generative similarity-based classifier. Because local SDA produces probability values, it could be combined with metric statistical learning algorithms using probability rules, which may be helpful in the practical case that samples are described both by categorical features or similarities, and by numerical features. The experiments show that local SDA consistently performs better than other model-based classifiers and performs competitively with k-NN. The PSVM can achieve lower error rates, particularly when the number of training samples is small compared to an underlying feature dimension. However, the PSVM may not be as well-suited for mixed-feature problems, asymmetric misclassification costs, multiclass problems, and problems where the number of training samples n is too large to compute with the $n \times n$ similarity matrix.

An additional advantage of model-based approaches is that they provide intuitive information about the local similarity characteristics of the data. A local class centroid can be viewed as a representative prototype for the class in the neighborhood of a test sample and the class-conditional probabilities provide an estimate of the local distribution of the similarities to the local centroid. There are many open questions in applying classifiers locally. In particular, it would be interesting to compare the local SDA performance to a mixture of SDA model components. How to best train a mixture of SDA model components is an open question.

References

Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *24*, 509–522.

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

Cost, S., & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, *10*, 57–78.

Cover, T., & Thomas, J. (1991). *Elements of information theory*. New York: John Wiley and Sons.

Devroye, L., Gyorfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. New York: Springer-Verlag Inc.

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

Friedlander, M. P., & Gupta, M. R. (2006). On minimizing distortion and relative entropy. *IEEE Trans. on Information Theory*, *52*, 238–245.

Gati, I., & Tversky, A. (1984). Weighting common and distinctive features in perceptual and conceptual judgments. *Cognitive Psychology*, 341–370.

Goldfarb, L. (1985). A new approach to pattern recognition. *Progress in Pattern Recognition*, *2*, 241–402.

Graepel, T., Herbrich, R., & Obermayer, K. (1999). Classification on pairwise proximity data. *Advances in Neural Information Processing Systems 11*, 438–444.

Gupta, M. R., Cazzanti, L., & Koppal, A. J. (2007). Maximum entropy generative models for similarity-based learning. *IEEE Intl. Symp. on Information Theory*. to appear.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York: Springer-Verlag.

Hochreiter, S., Mozer, M. C., & Obermayer, K. (2003). Coulomb classifiers: Generalizing support vector machines via an analogy to electrostatic systems. *Advances in Neural Information Processing Systems 15*, 545–552.

Hochreiter, S., & Obermayer, K. (2006). Support vector machines for dyadic data. *Neural Computation*, *18*, 1472–1510.

Hoffmann, T., & Buhmann, J. (1997). Pairwise data clustering by deterministic annealing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *19*.

Jacobs, D. W., Weinshall, D., & Gdalyahu, Y. (2000). Classification with nonmetric distances: Image retrieval and class representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *22*, 583–600.

Jaynes, E. T. (1982). On the rationale for maximum entropy methods. *Proc. of the IEEE*, *70*, 939–952.

Jaynes, E. T. (2003). *Probability theory: the logic of science*. Cambridge University Press.

Mitani, Y., & Hamamoto, Y. (2000). Classifier design based on the use of nearest neighbor samples. *Proc. of the Intl. Conf. on Pattern Recognition*, 769–772.

Mitani, Y., & Hamamoto, Y. (2006). A local mean-based nonparametric classifier. *Pattern Recognition Letters*, *27*, 1151–1159.

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

Pekalska, E., Paclíc, P., & Duin, R. P. W. (2001). A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, 175–211.

Rosch, E. (1973). Natural categories. *Cognitive Psychology*, 328 – 350.

Simard, P., Cun, Y. L., & Denker, J. (1993). Efficient pattern recognition using a new transformation distance. *Advances in Neural Information Processing Systems 5*, 50–68.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, *29*, 1213–1228.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 327–352.

Tversky, A., & Gati, I. (1978). Studies of similarity. In E. Rosch and B. Lloyd (Eds.), *Cognition and categorization*. Hillsdale, N.J.: Earlbaum.

Van Campenhout, J., & Cover, T. (1981). Maximum entropy and conditional probability. *IEEE Trans. on Information Theory*, *27*, 483–489.

Weinshall, D., Jacobs, D. W., & Gdalyahu, Y. (1999). Classification in non-metric spaces. *Advances in Neural Information Processing Systems 11*, 838–844.

Zhang, H., Berg, A. C., Maire, M., & Malik, J. (2006). SVM-KNN: discriminative nearest neighbor classification for visual category recognition. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2126 – 2136.