Journal of Machine Learning Research 10 (2009) 747-776 Submitted 9/08; Revised 2/09; Published 3/09

Similarity-based Classification: Concepts and Algorithms

Yihua Chen YHCHEN@U.WASHINGTON.EDU
Eric K. Garcia GARCIAER@U.WASHINGTON.EDU
Maya R. Gupta GUPTA@EE.WASHINGTON.EDU
Department of Electrical Engineering
University of Washington
Seattle, WA 98195, USA

Ali Rahimi ALI.RAHIMI@INTEL.COM
1100 NE 45th Street
Intel Research
Seattle, WA 98105, USA

Luca Cazzanti LUCA@APL.WASHINGTON.EDU
Applied Physics Lab
University of Washington
Seattle, WA 98105, USA

Editor: Alexander J. Smola

Abstract

This paper reviews and extends the field of similarity-based classification, presenting new analyses, algorithms, data sets, and a comprehensive set of experimental results for a rich collection of classification problems. Specifically, the generalizability of using similarities as features is analyzed, design goals and methods for weighting nearest-neighbors for similarity-based learning are proposed, and different methods for consistently converting similarities into kernels are compared. Experiments on eight real data sets compare eight approaches and their variants to similarity-based learning.

Keywords: similarity, dissimilarity, similarity-based learning, indefinite kernels

1. Introduction

Similarity-based classifiers estimate the class label of a test sample based on the similarities between the test sample and a set of labeled training samples, and the pairwise similarities between the training samples. Like others, we use the term *similarity-based classification* whether the pairwise relationship is a similarity or dissimilarity. Similarity-based classification does not require direct access to the features of the samples, and thus the sample space can be any set, not necessarily a Euclidean space, as long as the similarity function is well defined for any pair of samples. Let Ω be the sample space and $\mathcal{G}$ be the finite set of class labels. Let $\psi : \Omega \times \Omega \to \mathbb{R}$ be the similarity function. We assume that the pairwise similarities between n training samples are given as an $n \times n$ similarity matrix S whose (i, j)-entry is $\psi(x_i, x_j)$, where $x_i \in \Omega$, $i = 1, \ldots, n$, denotes the ith training sample, and $y_i \in \mathcal{G}$, $i = 1, \ldots, n$ the corresponding ith class label. The problem is to estimate the class label $\hat{y}$ for a test sample x based on its similarities to the training samples $\psi(x, x_i)$, $i = 1, \ldots, n$ and its self-similarity $\psi(x, x)$.

Similarity-based classification is useful for problems in computer vision, bioinformatics, information retrieval, natural language processing, and a broad range of other fields. Similarity functions may be asymmetric and fail to satisfy the other mathematical properties required for metrics or inner products (Santini and Jain, 1999). Some simple example similarity functions are: travel time from one place to another, compressibility of one random process given a code built for another, and the minimum number of steps to convert one sequence into another (edit distance). Computer vision researchers use many similarities, such as the tangent distance (Duda et al., 2001), earth mover's distance (EMD) (Rubner et al., 2000), shape matching distance (Belongie et al., 2002), and pyramid match kernel (Grauman and Darrell, 2007) to measure the similarity or dissimilarity between images in order to do image retrieval and object recognition. In bioinformatics, the Smith-Waterman algorithm (Smith and Waterman, 1981), the FASTA algorithm (Lipman and Pearson, 1985) and the BLAST algorithm (Altschul et al., 1990) are popular methods to compute the similarity between different amino acid sequences for protein classification. The cosine similarity between term frequency-inverse document frequency (tf-idf) vectors is widely used in information retrieval and text mining for document classification.

Notions of similarity appear to play a fundamental role in human learning, and thus psychologists have done extensive research to model human similarity judgement. Tversky's *contrast model* and *ratio model* (Tversky, 1977) represent an important class of similarity functions. In these two models, each sample is represented by a set of features, and the similarity function is an increasing function of set overlap but a decreasing function of set differences. Tversky's set-theoretic similarity models have been successful in explaining human judgement in various similarity assessment tasks, and are consistent with the observations made by psychologists that metrics do not account for cognitive judgement of similarity in complex situations (Tversky, 1977; Tversky and Gati, 1982; Gati and Tversky, 1984). Therefore, similarity-based classification may be useful for imitating or understanding how humans categorize.

The main contributions of this paper are: (1) we distill and analyze concepts and issues specific to similarity-based learning, including the generalizability of using similarities as features, (2) we propose similarity-based nearest-neighbor design goals and methods, and (3) we present a comprehensive set of experimental results for eight similarity-based learning problems and eight different similarity-based classification approaches and their variants. First, we discuss the idea of similarities as inner products in Section 2, then the concept of treating similarities as features in Section 3. The generalizability of using similarities as features and that of using similarities as kernels are compared in Section 4. In Section 5, we propose design goals and solutions for similarity-based weighted nearest-neighbor learning. Generative similarity-based classifiers are discussed in Section 6. Then in Section 7 we describe eight similarity-based classification problems, detail our experimental setup, and discuss the results. The paper concludes with some open questions in Section 8. For the reader's reference, key notation is summarized in Table 1.

2. Similarities as Inner Products

A popular approach to similarity-based classification is to treat the given similarities as inner products in some Hilbert space or to treat dissimilarities as distances in some Euclidean space. This approach can be roughly divided into two categories: one is to explicitly embed the samples in a Euclidean space according to the given (dis)similarities using multidimensional scaling (see Borg and Groenen, 2005, for further reading); the other is to modify the similarities to be kernels and

Ω	sample space	S	$n \times n$ matrix with (i,j)-entry $\psi(x_i,x_j)$
$\mathcal{G}$	set of class labels	s_i	$n \times 1$ vector with jth element $\psi(x_i,x_j)$
n	number of training samples	s	$n \times 1$ vector with jth element $\psi(x,x_j)$
$x_i \in \Omega$	ith training sample	$\mathbf{1}$	column vector of 1's
$x \in \Omega$	test sample	I	identity matrix
$y_i \in \mathcal{G}$	class label of ith training sample	$I_{\{\cdot\}}$	indicator function
$y \in \mathcal{G}^n$	$n \times 1$ vector with ith element y_i	K	kernel matrix or kernel function
$\hat{y} \in \mathcal{G}$	estimated class label for x	k	neighborhood size
$\mathcal{D}$	n training sample pairs $\{(x_i,y_i)\}_{i=1}^n$	L	hinge loss function
$\psi : \Omega \times \Omega \to \mathbb{R}$	similarity function	$\mathrm{diag}(a)$	diagonal matrix with a as the diagonal

Table 1: Key Notation

apply kernel classifiers. We discuss different methods for modifying similarities into kernels in Section 2.1. An important technicality is how to handle test samples, which is addressed in Section 2.2.

2.1 Modify Similarities into Kernels

The power of kernel methods lies in the implicit use of a reproducing kernel Hilbert space (RKHS) induced by a positive semidefinite (PSD) kernel (Schölkopf and Smola, 2002). Although the mathematical meaning of a kernel is the inner product in some Hilbert space, a standard interpretation of a kernel is the pairwise similarity between different samples. Conversely, many researchers have suggested treating similarities as kernels, and applying any classification algorithm that only depends on inner products. Using similarities as kernels eliminates the need to explicitly embed the samples in a Euclidean space.

Here we focus on the support vector machine (SVM), which is a well-known representative of kernel methods, and thus appears to be a natural approach to similarity-based learning. All the SVM algorithms that we discuss in this paper are for binary classification[1] such that $y_i \in \{\pm 1\}$. Let y be the $n \times 1$ vector whose ith element is y_i. The SVM dual problem can be written as

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T \,\mathrm{diag}(y) K \,\mathrm{diag}(y)\alpha \\ \text{subject to} \quad & 0 \preceq \alpha \preceq C\mathbf{1}, \quad y^T\alpha = 0, \end{aligned} \tag{1}$$

with variable $\alpha \in \mathbb{R}^n$, where $C > 0$ is the hyperparameter, K is a PSD kernel matrix whose (i,j)-entry is $K(x_i,x_j)$, $\mathbf{1}$ is the column vector with all entries one, and $\preceq$ denotes component-wise inequality for vectors. The corresponding decision function is Schölkopf and Smola (2002)

$$\hat{y} = \mathrm{sgn}\left(\sum_{i=1}^n \alpha_i y_i K(x,x_i) + b\right),$$

where

$$b = y_i - \sum_{j=1}^n \alpha_j y_j K(x_i,x_j)$$

for any i that satisfies $0 < \alpha_i < C$. The theory of RKHS requires the kernel to satisfy Mercer's condition, and thus the corresponding kernel matrix K must be PSD. However, many similarity

1. We refer the reader to Hsu and Lin (2002) for multiclass SVM.

functions do not satisfy the properties of an inner product, and thus the similarity matrix S can be indefinite. In the following subsections we discuss several methods to modify similarities into kernels; a previous review can be found in Wu et al. (2005). Unless mentioned otherwise, in the following subsections we assume that S is symmetric. If not, we use its symmetric part $\frac{1}{2}\left(S+S^T\right)$ instead. Notice that the symmetrization does not affect the SVM objective function in (1) since $\alpha^T \frac{1}{2}\left(S+S^T\right)\alpha = \frac{1}{2}\alpha^T S\alpha + \frac{1}{2}\alpha^T S^T\alpha = \alpha^T S\alpha$.

2.1.1 INDEFINITE KERNELS

One approach is to simply replace K with S, and ignore the fact that S is indefinite. For example, although the SVM problem given by (1) is no longer convex when S is indefinite, Lin and Lin (2003) show that the sequential minimal optimization (SMO) (Platt, 1998) algorithm will still converge with a simple modification to the original algorithm, but the solution is a stationary point instead of a global minimum. Ong et al. (2004) interpret this as finding the stationary point in a reproducing kernel Kreĭn space (RKKS), while Haasdonk (2005) shows that this is equivalent to minimizing the distance between reduced convex hulls in a pseudo-Euclidean space. A Kreĭn space, denoted by $\mathcal{K}$, is defined to be the direct sum of two disjoint Hilbert spaces, denoted by $\mathcal{H}_+$ and $\mathcal{H}_-$, respectively. So for any $a,b \in \mathcal{K} = \mathcal{H}_+ \oplus \mathcal{H}_-$, there are unique $a_+, b_+ \in \mathcal{H}_+$ and unique $a_-, b_- \in \mathcal{H}_-$ such that $a = a_+ + a_-$ and $b = b_+ + b_-$. The "inner product" on $\mathcal{K}$ is defined as

$$\langle a,b\rangle_{\mathcal{K}} = \langle a_+, b_+\rangle_{\mathcal{H}_+} - \langle a_-, b_-\rangle_{\mathcal{H}_-},$$

which no longer has the property of positive definiteness. Pseudo-Euclidean space is a special case of Kreĭn space where $\mathcal{H}_+$ and $\mathcal{H}_-$ are two Euclidean spaces. Ong et al. (2004) provide a representer theorem for RKKS that poses learning in RKKS as a problem of finding a stationary point of the risk functional, in contrast to minimizing a risk functional in RKHS. Using indefinite kernels in empirical risk minimization (ERM) methods such as SVM can lead to a saddle point solution and thus does not ensure minimizing the risk functional, so this approach does not guarantee learning in the sense of a good function approximation. Also, the nonconvexity of the problem may require intensive computation.

2.1.2 SPECTRUM CLIP

Since S is assumed to be symmetric, it has an eigenvalue decomposition $S = U^T \Lambda U$, where U is an orthogonal matrix and Λ is a diagonal matrix of real eigenvalues, that is, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. Spectrum clip makes S PSD by clipping all the negative eigenvalues to zero. Some researchers assume that the negative eigenvalues of the similarity matrix are caused by noise and view spectrum clip as a denoising step (Wu et al., 2005). Let

$$\Lambda_{\text{clip}} = \mathrm{diag}\left(\max(\lambda_1, 0), \ldots, \max(\lambda_n, 0)\right),$$

and the modified PSD similarity matrix be $S_{\text{clip}} = U^T \Lambda_{\text{clip}} U$. Let u_i denote the ith column vector of U. Using S_{clip} as a kernel matrix for training the SVM is equivalent to implicitly using $x_i = \Lambda_{\text{clip}}^{1/2} u_i$ as the representation of the ith training sample since $\langle x_i, x_j\rangle$ is equal to the (i,j)-entry of S_{clip}. A mathematical justification for spectrum clip is that S_{clip} is the nearest PSD matrix to S in terms of the Frobenius norm (Higham, 1988), that is,

$$S_{\text{clip}} = \arg\min_{K \succeq 0} \|K - S\|_F,$$

where $\succeq$ denotes the generalized inequality with respect to the PSD cone.

Recently, Luss and d'Aspremont (2007) have proposed a robust extension of SVM for indefinite kernels. Instead of only considering the nearest PSD matrix S_{clip}, they consider all the PSD matrices within distance β to S, that is, $\{K \succeq 0 \,|\, \|K-S\|_F \leq \beta\}$, where $\beta > \min_{K\succeq 0} \|K-S\|_F$, and propose to maximize the worst case of the SVM dual objective among these matrices:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \min_{K\succeq 0, \|K-S\|_F \leq \beta} \left(\mathbf{1}^T\alpha - \frac{1}{2}\alpha^T \operatorname{diag}(y) K \operatorname{diag}(y)\alpha \right) \\ \text{subject to} \quad & 0 \preceq \alpha \preceq C\mathbf{1}, \quad y^T\alpha = 0. \end{aligned}$$

This model is more flexible in the sense that the set of possible K in the hypothesis space lies in a ball with radius β centered around S. Different draws of training samples will change the candidate set of K and could cause overfitting. In practice, they replace the hard constraint $\|K-S\|_F \leq \beta$ with a penalty term and propose the following problem as the robust SVM for S:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \min_{K\succeq 0} \left(\mathbf{1}^T\alpha - \frac{1}{2}\alpha^T \operatorname{diag}(y) K \operatorname{diag}(y)\alpha + \rho\|K-S\|_F^2 \right) \\ \text{subject to} \quad & 0 \preceq \alpha \preceq C\mathbf{1}, \quad y^T\alpha = 0, \end{aligned} \tag{2}$$

where $\rho > 0$ is the parameter to control the trade-off. They point out that the inner problem of (2) has a closed-form solution, and the outer problem is convex since its objective is a pointwise minimum of a set of concave quadratic functions of α and thus concave. A fast algorithm to solve (2) is given by Chen and Ye (2008).

2.1.3 SPECTRUM FLIP

In contrast to the interpretation that negative eigenvalues are caused by noise, Laub and Müller (2004) and Laub et al. (2006) show that the negative eigenvalues of some similarity data can code useful information about object features or categories, which agrees with some fundamental psychological studies (Tversky and Gati, 1982; Gati and Tversky, 1982). In order to use the negative eigenvalues, Graepel et al. (1998) propose an SVM in pseudo-Euclidean space, and Pekalska et al. (2001) also consider a generalized nearest mean classifier and Fisher linear discriminant classifier in the same space. Following the notation in Section 2.1.1, they assume that the samples lie in a Kreĭn space $\mathcal{K} = \mathcal{H}_+ \oplus \mathcal{H}_-$ with similarities given by $\psi(a,b) = \langle a_+, b_+\rangle_{\mathcal{H}_+} - \langle a_-, b_-\rangle_{\mathcal{H}_-}$. These proposed classifiers are their standard versions in the Hilbert space $\mathcal{H} = \mathcal{H}_+ \oplus \mathcal{H}_-$ with associated inner product $\langle a,b\rangle_{\mathcal{H}} = \langle a_+, b_+\rangle_{\mathcal{H}_+} + \langle a_-, b_-\rangle_{\mathcal{H}_-}$. This is equivalent to flipping the sign of the negative eigenvalues of the similarity matrix S: let $\Lambda_{\text{flip}} = \operatorname{diag}(|\lambda_1|, \ldots, |\lambda_n|)$, and then the similarity matrix after spectrum flip is $S_{\text{flip}} = U^T\Lambda_{\text{flip}}U$. Wu et al. (2005) note that this is the same as replacing the original eigenvalues of S with its singular values.

2.1.4 SPECTRUM SHIFT

Spectrum shift is another popular approach to modifying a similarity matrix into a kernel matrix: since $S + \lambda I = U^T(\Lambda + \lambda I)U$, any indefinite similarity matrix can be made PSD by shifting its spectrum by the absolute value of its minimum eigenvalue $|\lambda_{\min}(S)|$. Let $\Lambda_{\text{shift}} = \Lambda + |\min(\lambda_{\min}(S), 0)|\, I$, which is used to form the modified similarity matrix $S_{\text{shift}} = U^T\Lambda_{\text{shift}}U$. Compared with spectrum clip and flip, spectrum shift only enhances all the self-similarities by the amount of $|\lambda_{\min}(S)|$ and

does not change the similarity between any two different samples. Roth et al. (2003) propose spectrum shift for clustering nonmetric proximity data and show that S_{shift} preserves the group structure of the original data represented by S. Let $\mathcal{X}$ be the set of samples to cluster, and $\{\mathcal{X}_\ell\}_{\ell=1}^N$ be a partition of $\mathcal{X}$ into N sets. Specifically, they consider minimizing the clustering cost function[2]

$$f\left(\{\mathcal{X}_\ell\}_{\ell=1}^N\right) = -\sum_{\ell=1}^{N} \sum_{\substack{i,j\in\mathcal{X}_\ell \\ i\neq j}} \frac{\psi(x_i,x_j)}{|\mathcal{X}_\ell|}, \tag{3}$$

where $|\mathcal{X}_\ell|$ denotes the cardinality of set $\mathcal{X}_\ell$. It is easy to see that (3) is invariant under spectrum shift.

Recently, Zhang et al. (2006) proposed training an SVM only on the k-nearest neighbors of each test sample, called SVM-KNN. They used spectrum shift to produce a kernel from the similarity data. Their experimental results on image classification demonstrated that SVM-KNN performs comparably to a standard SVM classifier, but with significant reduction in training time.

2.1.5 SPECTRUM SQUARE

The fact that $SS^T \succeq 0$ for any $S \in \mathbb{R}^{n\times n}$ led us to consider using SS^T as a kernel, which is valid even when S is not symmetric. For symmetric S, this is equivalent to squaring its spectrum since $SS^T = U^T\Lambda^2 U$. It is also true that using SS^T is the same as defining a new similarity function $\tilde{\psi}$ for any $a,b \in \Omega$ as

$$\tilde{\psi}(a,b) = \sum_{i=1}^{n} \psi(a,x_i)\psi(x_i,b).$$

We note that for symmetric S, treating SS^T as a kernel matrix K is equivalent to representing each x_i by its similarity feature vector $s_i = \begin{bmatrix}\psi(x_i,x_1) & \dots & \psi(x_i,x_n)\end{bmatrix}^T$ since $K_{ij} = \langle s_i, s_j\rangle$. The concept of treating similarities as features is discussed in more detail in Section 3.

2.2 Consistent Treatment of Training and Test Samples

Consider a test sample x that is the same as a training sample x_i. Then if one uses an ERM classifier trained with modified similarities $\tilde{S}$, but uses the unmodified test similarities, represented by vector $s = \begin{bmatrix}\psi(x,x_1) & \dots & \psi(x,x_n)\end{bmatrix}^T$, the same sample will be treated inconsistently. In general, one would like to modify the training and test similarities in a consistent fashion, that is, to modify the underlying similarity function rather than only modifying the S. In this context, given S and $\tilde{S}$, we term a transformation T on test samples *consistent* if $T(s_i)$ is equal to the ith row of $\tilde{S}$ for $i = 1,\dots,n$.

One solution is to modify the training and test samples all at once. However, when test samples are not known beforehand, this may not be possible. For such cases, Wu et al. (2005) proposed to first modify S and train the classifier using the modified $n\times n$ similarity matrix $\tilde{S}$, and then for each test sample modify its s in an effort to be consistent with the modified similarities used to train the model. Their approach is to re-compute the same modification on the augmented $(n+1)\times(n+1)$ similarity matrix

$$S' = \begin{bmatrix} S & s \\ s^T & \psi(x,x)\end{bmatrix}$$

2. They originally use dissimilarities in their cost function, and we reformulate it into similarities with the assumption that the relationship between dissimilarities and similarities is affine.

to form $\tilde{S}'$, and then let the modified test similarities $\tilde{s}$ be the first n elements of the last column of $\tilde{S}'$. The classifier that was trained on $\tilde{S}$ is then applied on $\tilde{s}$. To implement this approach, Wu et al. (2005) propose a fast algorithm to perform eigenvalue decomposition of S' by using the results of the eigenvalue decomposition of S. However, this approach does not guarantee consistency.

To attain consistency, we note that both the spectrum clip and flip modifications can be represented by linear transformations, that is, $\tilde{S} = PS$, where P is the corresponding transformation matrix, and we propose to apply the same linear transformation P on s such that $\tilde{s} = Ps$. For spectrum flip, the linear transformation is $P_{\text{flip}} = U^T M_{\text{flip}} U$, where

$$M_{\text{flip}} = \text{diag}\left(\text{sgn}(\lambda_1), \ldots, \text{sgn}(\lambda_n)\right).$$

For spectrum clip, the linear transformation is $P_{\text{clip}} = U^T M_{\text{clip}} U$, where

$$M_{\text{clip}} = \text{diag}\left(I_{\{\lambda_1 \geq 0\}}, \ldots, I_{\{\lambda_n \geq 0\}}\right),$$

and $I_{\{\cdot\}}$ is the indicator function. Recall that using $\tilde{S}$ implies embedding the training samples in a Euclidean space. For spectrum clip, this linear transformation is equivalent to embedding the test sample as a feature vector into the same Euclidean space of the embedded training samples:

Proposition 1 *Let S_{clip} be the Gram matrix of the column vectors of $X \in \mathbb{R}^{m \times n}$, where $\text{rank}(X) = m$. For a given s, let $x = \arg\min_{z \in \mathbb{R}^m} \|X^T z - s\|_2$, then $X^T x = P_{\text{clip}} s$.*

The proof is in the appendix.

Proposition 1 states that if the n training samples are embedded in $\mathbb{R}^m$ with S_{clip} as the Gram matrix, and we embed the test sample in $\mathbb{R}^m$ by finding the feature vector whose inner products with the embedded training samples are closest to the given s, then the inner products between the embedded test sample and the embedded training samples are indeed $\tilde{s} = P_{\text{clip}} s$.

On the other hand, there is no linear transformation to ensure consistency for spectrum shift. For our experiments using spectrum shift, we adopt the approach of Wu et al. (2005), which for this case is to let $\tilde{s} = s$, because spectrum shift only affects self-similarities.

3. Similarities as Features

Similarity-based classification problems can be formulated into standard learning problems in Euclidean space by treating the similarities between a sample x and the n training samples as features (Graepel et al., 1998, 1999; Pekalska et al., 2001; Pekalska and Duin, 2002; Liao and Noble, 2003). That is, represent sample x by the similarity feature vector s. As detailed in Section 4, the generalizability analysis yields different results for using similarities as features and using similarities as inner products.

Graepel et al. (1998) consider applying a linear SVM on similarity feature vectors by solving the following problem:

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} L(w^T s_i + b, y_i) \tag{4}$$

with variables $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ and hyperparameter $C > 0$, where $L(\alpha, \beta) \triangleq \max(1 - \alpha\beta, 0)$ is the hinge loss function. Liao and Noble (2003) also propose to apply an SVM on similarity feature vectors; they use a Gaussian radial basis function (RBF) kernel.

In order to make the solution w sparser, which helps ease the computation of the discriminant function $f(s) = w^T s + b$, Graepel et al. (1999) substitute the ℓ_1-norm regularization for the squared ℓ_2-norm regularization in (4), and propose a linear programming (LP) machine:

$$\underset{w,b}{\text{minimize}} \quad \|w\|_1 + C\sum_{i=1}^{n} L(w^T s_i + b, y_i). \tag{5}$$

Balcan et al. (2008a) provide a theoretical analysis for using similarities as features, and show that if a similarity is good in the sense that the expected intraclass similarity is sufficiently large compared to the expected interclass similarity, then given n training samples, there exists a linear separator on the similarities as features that has a specifiable maximum error at a margin that depends on n. Specifically, Theorem 4 in Balcan et al. (2008a) gives a sufficient condition on the similarity function ψ for (4) to achieve good generalization. Their latest results for ℓ_1-margin (inversely proportional to $\|w\|_1$) provide similar theoretical guarantees for (5) (Balcan et al., 2008b, Theorem 11). Wang et al. (2007) show that under slightly less restrictive assumptions on the similarity function there exists with high probability a convex combination of simple classifiers on the similarities as features which has a maximum specifiable error.

Another approach is the potential support vector machine (P-SVM) (Hochreiter and Obermayer, 2006; Knebel et al., 2008), which solves

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & \frac{1}{2}\|y - S\alpha\|_2^2 + \varepsilon\|\alpha\|_1 \\ \text{subject to} \quad & \|\alpha\|_\infty \leq C, \end{aligned} \tag{6}$$

where $C > 0$ and $\varepsilon > 0$ are two hyperparameters. We note that by strong duality (6) is equivalent to

$$\underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\|y - S\alpha\|_2^2 + \varepsilon\|\alpha\|_1 + \gamma\|\alpha\|_\infty \tag{7}$$

for some $\gamma > 0$. One can see from (7) that P-SVM is equivalent to the lasso regression (Tibshirani, 1996) with an extra ℓ_∞-norm regularization term. The use of multiple regularization terms in P-SVM is similar to the elastic net (Zou and Hastie, 2005), which uses ℓ_1 and squared ℓ_2 regularization together.

The algorithms above minimize the empirical risk with regularization. In addition, Pekalska et al. consider generative classifiers for similarity feature vectors; they propose a regularized Fisher linear discriminant classifier (Pekalska et al., 2001) and a regularized quadratic discriminant classifier (Pekalska and Duin, 2002).

We note that treating similarities as features may not capture discriminative information if there is a large intraclass variance compared to the interclass variance, even if the classes are well-separated. A simple example is if the two classes are generated by Gaussian distributions with highly-ellipsoidal covariances, and the similarity function is taken to be a negative linear function of the distance.

4. Generalization Bounds of Similarity SVM Classifiers

To investigate the generalizability of SVM classifiers using similarities, we analyze two forms of SVMs: using similarity as a kernel as discussed in Section 2, and a linear SVM using the similarities as features as given by (4). When similarities are used as features, we show that good generalization

performance can be achieved by training the SVM on a small subset of m ($< n$) randomly selected training examples, and we compare this to the established analysis for the kernelized SVM.

The SVM learns a discriminant function $f(s) = w^T s + b$ by minimizing the empirical risk

$$\hat{R}_{\mathcal{D}}(f, L) = \frac{1}{n} \sum_{i=1}^{n} L(f(s_i), y_i),$$

where $\mathcal{D}$ denotes the training set, subject to some smoothness constraint $\mathcal{N}$, that is,

$$\underset{f}{\text{minimize}} \quad \hat{R}_{\mathcal{D}}(f, L) + \lambda_n \mathcal{N}(f),$$

where $\lambda_n = \frac{1}{2nC}$. We note that using (arbitrary) similarities as features corresponds to setting $\mathcal{N}(f) = w^T w$, while using (PSD) similarities as a kernel changes the smoothness constraint to $\mathcal{N}(f) = w^T S w$ (Rifkin, 2002, Appendix B), and in fact, this change of regularizer is the only difference between these two similarity-based SVM approaches. In this section, we examine how this small change in regularization affects the generalization ability of the classifiers.

To simplify the following analysis, we do not directly investigate the SVM classifier as presented; instead, as is standard in SVM learning theory, we investigate the following constrained version of the problem:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \hat{R}_{\mathcal{D}}(f, L_t) \\ \text{subject to} \quad & \mathcal{N}(f) \leq \beta^2, \end{aligned} \tag{8}$$

with truncated hinge loss $L_t \triangleq \min(L, 1) \in [0, 1]$ and $f(s) = w^T s$ stripped of the intercept b.

The following generalization bound for the SVM using (PSD) similarities[3] as a kernel follows directly from the results in Bartlett and Mendelson (2002).

Theorem 1 (Generalization Bound of Similarities as Kernel) *Suppose* (x, y) *and* $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ *are drawn i.i.d. from a distribution on* $\Omega \times \{\pm 1\}$. *Let* ψ *be a positive definite similarity such that* $\psi(a, a) \leq \kappa^2$ *for some* $\kappa > 0$ *and all* $a \in \Omega$. *Let* S *be the* $n \times n$ *matrix with* (i, j)*-entry* $\psi(x_i, x_j)$ *and* s *be the* $n \times 1$ *vector with* i*th element* $\psi(x, x_i)$. *Define* F_S *to be the set of real-valued functions* $\left\{ f(s) = w^T s \,\middle|\, w^T S w \leq \beta^2 \right\}$ *for a finite* β. *Then with probability at least* $1 - \delta$ *with respect to* $\mathcal{D}$, *every function* f *in* F_S *satisfies*

$$P(y f(s) \leq 0) \leq \hat{R}_{\mathcal{D}}(f, L_t) + 4\beta\kappa \sqrt{\frac{1}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}}.$$

The proof is in the appendix.

Theorem 1 says that with high probability, as $n \to \infty$, the misclassification rate is tightly bounded by the empirical risk $\hat{R}_{\mathcal{D}}(f, L_t)$, implying that a discriminant function trained by (8) with $\mathcal{N}(f) = w^T S w$ generalizes well to unseen data.

Next, we state a weaker result in Theorem 2 for the SVM using (arbitrary) similarities as features. Let the features be the similarities to m ($< n$) prototypes $\{(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_m, \tilde{y}_m)\} \subseteq \mathcal{D}$ randomly chosen from the training set so that $\tilde{s}$ is the $m \times 1$ vector with ith element $\psi(x, \tilde{x}_i)$. Results will be obtained on the remaining $n - m$ training data $\widetilde{\mathcal{D}} = \mathcal{D} \backslash \{(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_m, \tilde{y}_m)\}$.

3. If the original similarities are not PSD, then they must be modified to be PSD before this result applies; see Section 2 for a discussion of common PSD modifications.

Theorem 2 (Generalization Bound of Similarities as Features) *Suppose* (x,y) *and* $\mathcal{D} = \{(x_i,y_i)\}_{i=1}^n$ *are drawn i.i.d. from a distribution on* $\Omega \times \{\pm 1\}$. *Let* ψ *be a similarity such that* $\psi(a,b) \leq \kappa^2$ *for some* $\kappa > 0$ *and all* $a,b \in \Omega$. *Let* $\{(\tilde{x}_1,\tilde{y}_1),\ldots,(\tilde{x}_m,\tilde{y}_m)\} \subseteq \mathcal{D}$ *be a set of randomly chosen prototypes, and denote* $\widetilde{\mathcal{D}} = \mathcal{D}\backslash\{(\tilde{x}_1,\tilde{y}_1),\ldots,(\tilde{x}_m,\tilde{y}_m)\}$. *Let* $\tilde{s}$ *be the* $m \times 1$ *vector with ith element* $\psi(x,\tilde{x}_i)$. *Define* F_I *to be the set of real-valued functions* $\left\{ f(\tilde{s}) = w^T\tilde{s} \,\middle|\, w^T w \leq \beta^2 \right\}$ *for a finite* β. *Then with probability at least* $1-\delta$ *with respect to* $\widetilde{\mathcal{D}}$, *every function* f *in* F_I *satisfies*

$$P(yf(s) \leq 0) \leq \hat{R}_{\widetilde{\mathcal{D}}}(f,L_t) + 4\beta\kappa^2\sqrt{\frac{m}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}}.$$

The proof is in the appendix.

Theorem 2 only differs significantly from Theorem 1 in the term $\sqrt{m/n}$, which means that if m, the number of prototypes used, grows no faster than $o(n)$, then with high probability, as $n \to \infty$, the misclassification rate is tightly bounded by the empirical risk on the remaining training set $\hat{R}_{\widetilde{\mathcal{D}}}(f,L_t)$. Note that Theorem 2 is unable to claim anything about the generalization when $m = n$, that is, the entire training set is chosen as prototypes. For a further discussion, see the appendix.

5. Similarity-based Weighted Nearest-Neighbors

In this section, we consider design goals and propose solutions for weighted nearest-neighbors for similarity-based classification. Nearest-neighbor learning is the algorithmic parallel of the *exemplar* model of human learning (Goldstone and Kersten, 2003). Weighted nearest-neighbor algorithms are task-flexible because the weights on the neighbors can be used as probabilities as long as they are non-negative and sum to one. For classification, such weights can be summed for each class to form posteriors, which is helpful for use with asymmetric misclassification costs and when the similarity-based classifier is a component of a larger decision-making system. As a lazy learning method, weighted nearest-neighbor classifiers do not require training before the arrival of test samples. This can be advantageous to certain applications where the amount of training data is huge, or there are a large number of classes, or the training data is constantly evolving.

5.1 Design Goals for Similarity-based Weighted k-NN

In this section, for a test sample x, we use x_i to denote its ith nearest neighbor from the training set as defined by the similarity function ψ for $i = 1,\ldots,k$, and y_i to denote the label of x_i. Also, we redefine S as the $k \times k$ matrix of the similarities between the k-nearest neighbors and s the $k \times 1$ vector of the similarities between the test sample x and its k-nearest neighbors. For each test sample, weighted k-NN assigns weight w_i to the ith nearest neighbor for $i = 1,\ldots,k$. Weighted k-NN classifies the test sample x as the class $\hat{y}$ that is assigned the most weight,

$$\hat{y} = \arg\max_{g \in \mathcal{G}} \sum_{i=1}^{k} w_i I_{\{y_i = g\}}. \tag{9}$$

It is common to additionally require that the weights be nonnegative and normalized such that the weights form a posterior distribution over the set of classes $\mathcal{G}$. Then the estimated probability for class g is $\sum_{i=1}^{k} w_i I_{\{y_i=g\}}$, which can be used with asymmetric misclassification costs.

An intuitive and standard approach to weighting nearest neighbors is to give larger weight to neighbors that are more similar to the test sample. Formally, we state:

Design Goal 1 (Affinity): w_i should be an increasing function of $\psi(x, x_i)$.

In addition, we propose a second design goal. In practice, some samples in the training set are often very similar, for example, a random sampling of the emails by one person may include many emails from the same thread that contain repeated text due to replies and forwarding. Such similar training samples provide highly-correlated information to the classifier. In fact, many of the nearest neighbors may provide very similar information which can bias the classifier. Moreover, we consider those training samples that are similar to many other training samples less valuable based on the same motivation for tf-idf. To address this problem, one can choose weights to down-weight highly similar samples and ensure that a diverse set of the neighbors has a voice in the classification decision. We formalize this goal as:

Design Goal 2 (Diversity): w_i should be a decreasing function of $\psi(x_i, x_j)$.

Next we propose two approaches to weighting neighbors for similarity-based classification that aim to satisfy these goals.

5.2 Kernel Ridge Interpolation Weights

First, we describe kernel regularized linear interpolation, and we show that it leads to weights that satisfy the design goals. Gupta et al. (2006) proposed weights for k-NN in Euclidean space that satisfy a linear interpolation with maximum entropy (LIME) objective:

$$\begin{array}{ll} \underset{w}{\text{minimize}} & \left\| \sum_{i=1}^{k} w_i x_i - x \right\|_2^2 - \lambda H(w) \\ \text{subject to} & \sum_{i=1}^{k} w_i = 1,\ w_i \geq 0,\ i = 1, \ldots, k, \end{array} \tag{10}$$

with variable $w \in \mathbb{R}^k$, where $H(w) = -\sum_{i=1}^{k} w_i \log w_i$ is the entropy of the weights and $\lambda > 0$ is a regularization parameter. The first term of the convex objective in (10) tries to solve the linear interpolation equations, which balances the weights so that the test point is best approximated by a convex combination of the training samples. Additionally, the entropy maximization pushes the LIME weights toward the uniform weights.

We simplify (10) to a quadratic programming (QP) problem by replacing the negative entropy regularization with a ridge regularizer[4] $w^T w$, and we rewrite (10) in matrix form:

$$\begin{array}{ll} \underset{w}{\text{minimize}} & \frac{1}{2} w^T X^T X w - x^T X w + \frac{\lambda}{2} w^T w \\ \text{subject to} & w \succeq 0, \quad \mathbf{1}^T w = 1, \end{array} \tag{11}$$

where $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \end{bmatrix}$. Note that (11) is completely specified in terms of the inner products of the feature vectors: $\langle x_i, x_j \rangle$ and $\langle x, x_i \rangle$, and thus we term the solution to (11) as *kernel ridge*

4. Due to the constraint $\mathbf{1}^T w = 1$, the ridge regularizer actually regularizes the variance of the weights and thus has similar effect as the negative entropy regularizer.

interpolation (KRI) weights. Generalizing from inner products to similarities, we form the KRI similarity-based weights:

$$\begin{array}{ll} \underset{w}{\text{minimize}} & \frac{1}{2} w^T S w - s^T w + \frac{\lambda}{2} w^T w \\ \text{subject to} & w \succeq 0, \quad \mathbf{1}^T w = 1. \end{array} \tag{12}$$

There are three terms in the objective function of (12). Acting alone, the linear term $-s^T w$ would give all the weight to the 1-nearest neighbor. This is prevented by the ridge regularization term $\frac{1}{2}\lambda w^T w$, which regularizes the variance of w and hence pushes the weights toward the uniform weights. These two terms work together to give more weight to the training samples that are more similar to the test sample, and thus help the resulting weights satisfy the first design goal of rewarding neighbors with high affinity to the test sample. The quadratic term in (12) can be expanded as follows,

$$\frac{1}{2} w^T S w = \frac{1}{2} \sum_{i,j} \psi(x_i, x_j) w_i w_j.$$

From the above expansion, one sees that holding all else constant in (12), the bigger $\psi(x_i, x_j)$ and $\psi(x_j, x_i)$ are, the smaller the chosen w_i and w_j will be. Thus the quadratic term tends to down-weight the neighbors that are similar to each other and acts to achieve the second design goal of spreading the weight among a diverse set of neighbors.

A sensitivity analysis further verifies the above observations. Let $g(w; S, s)$ be the objective function of (12), and $w^\star$ denote the optimal solution. To simplify the analysis, we only consider $w^\star$ in the interior of the probability simplex and thus $\nabla g(w^\star; S, s) = 0$. We first perturb s by adding $\delta > 0$ to its ith element, that is, $\tilde{s} = s + \delta e_i$, where e_i denotes the standard basis vector whose ith element is 1 and 0 elsewhere. Then

$$\nabla g(w^\star; S, \tilde{s}) = (S + \lambda I) w^\star - \tilde{s} = -\delta e_i,$$

whose projection on the probability simplex is

$$\nabla g(w^\star; S, \tilde{s}) - \frac{1}{k} \left(\mathbf{1}^T \nabla g(w^\star; S, \tilde{s}) \right) \mathbf{1} = \delta \left(\frac{1}{k} \mathbf{1} - e_i \right). \tag{13}$$

The direction of the steepest descent given by the negative of the projected gradient in (13) indicates that the new optimal solution will have an increased w_i, which satisfies the first design goal.

Similarly, if we instead perturb S by adding $\delta > 0$ to its (i, j)-entry ($i \neq j$), that is, $\tilde{S} = S + \delta E_{ij}$, where E_{ij} denotes the matrix with (i, j)-entry 1 and 0 elsewhere, then

$$\nabla g(w^\star; \tilde{S}, s) = (\tilde{S} + \lambda I) w^\star - s = \delta w_j^\star e_i,$$

whose projection on the probability simplex is

$$\nabla g(w^\star; \tilde{S}, s) - \frac{1}{k} \left(\mathbf{1}^T \nabla g(w^\star; \tilde{S}, s) \right) \mathbf{1} = \delta w_j^\star \left(e_i - \frac{1}{k} \mathbf{1} \right). \tag{14}$$

The direction of the steepest descent given by the negative of the projected gradient in (14) indicates that the optimal solution will have a decreased w_i, which satisfies the second design goal.

Experimentally, we found little statistically significant difference between using negative entropy or ridge regularization for the KRI weights. Analytically, entropy regularization leads to an

exponential form for the weights that can be used to prove consistency (Friedlander and Gupta, 2006). Computationally, the ridge regularizer is more practical because it results in a QP with box constraints and an equality constraint if the S matrix is PSD or approximated by a PSD matrix, and can thus be solved by the SMO algorithm (Platt, 1998).

5.2.1 KERNEL RIDGE REGRESSION WEIGHTS

A closed-form solution to (12) is possible if one relaxes the problem by removing the constraints $w_i \in [0,1]$ and $\sum_i w_i = 1$ that ensure the weights form a probability mass function. Then for PSD S, the objective $\frac{1}{2}w^T Sw - s^T w + \frac{1}{2}\lambda w^T w$ is solved by

$$w = (S + \lambda I)^{-1} s. \tag{15}$$

The k-NN decision rule (9) using these weights is equivalent to classifying by maximizing the discriminant of a local kernel ridge regression. For each class $g \in \mathcal{G}$, local kernel ridge regression (without intercept) solves

$$\underset{\beta_g}{\text{minimize}} \quad \sum_{i=1}^{k} \left(I_{\{y_i = g\}} - \langle \beta_g, \phi(x_i) \rangle \right)^2 + \lambda \langle \beta_g, \beta_g \rangle, \tag{16}$$

where ϕ denotes the mapping from the sample space Ω to a Hilbert space with inner product $\langle \phi(x_i), \phi(x_j) \rangle = \psi(x_i, x_j)$. Each solution to (16) yields the discriminant $f_g(x) = \langle \beta_g, \phi(x) \rangle = \nu_g^T (S + \lambda I)^{-1} s$ for class g (Cristianini and Shawe-Taylor, 2000), where $\nu_g = \left[I_{\{y_1 = g\}} \quad \dots \quad I_{\{y_k = g\}} \right]^T$. Maximizing $f_g(x)$ over $g \in \mathcal{G}$ produces the same estimated class label as (9) using the weights given in (15), thus we refer to these weights as *kernel ridge regression* (KRR) weights.

For a non-PSD S, it is possible that $S + \lambda I$ is singular. In the experiments, we compare handling non-PSD S by clip, flip, shift, or taking the pseudo-inverse (pinv) $(S + \lambda I)^\dagger$.

5.2.2 ILLUSTRATIVE EXAMPLES

We illustrate the KRI and KRR[5] weights with three toy examples shown on the next page. For each example, there are $k = 4$ nearest-neighbors, and the KRI and KRR weights are shown for a range of regularization parameter λ.

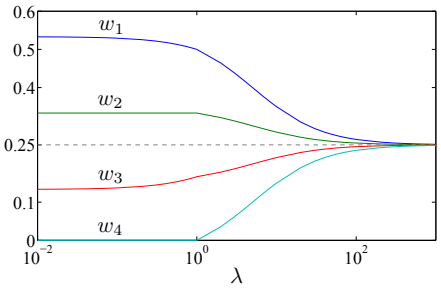
In Example 1, affinity is illustrated. One sees from s that the four distinct training samples are not equally similar to the test sample, and from S that the training samples have zero similarity to each other. Both KRI and KRR give more weight to training samples that are more similar to the test sample, illustrating that the weighting methods achieve the design goal of affinity.

In Example 2, diversity is illustrated. The four training samples all have similarity 3 to the test sample, but x_2 and x_3 are very similar to each other, and are thus weighted down as prescribed the by the design goal of diversity. Because of the symmetry of the similarities, the weights for x_2 and x_3 are exactly the same for both KRI and KRR.

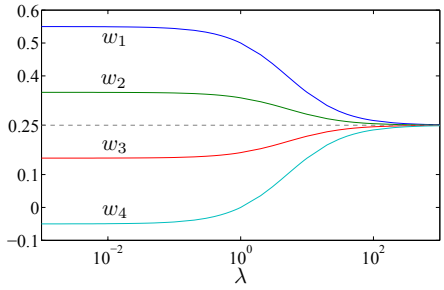
In Example 3, the interaction between the two design goals is illustrated. The S matrix is the same as in Example 2, but s is no longer uniform. In fact, although x_1 is less similar to the test sample than x_3, x_1 receives more weight because it is less similar to other training samples. The

5. For the purpose of comparison, we show the normalized KRR weights $\tilde{w}$ where $\tilde{w} = \left(I - \frac{1}{k}\mathbf{1}\mathbf{1}^T\right) w + \frac{1}{k}\mathbf{1}$. This does not affect the result of classification since each weight is shifted by the same constant.

Example 1: $s^T = \begin{bmatrix} 4 & 3 & 2 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$

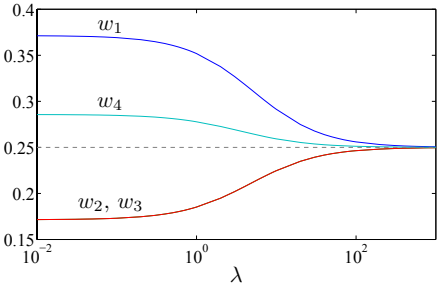


KRI weights

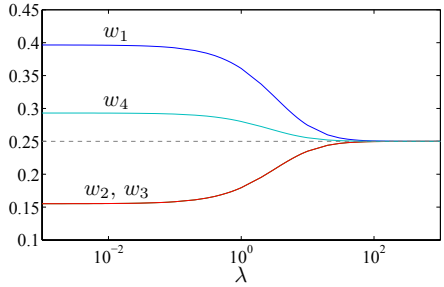


KRR weights

Example 2: $s^T = \begin{bmatrix} 3 & 3 & 3 & 3 \end{bmatrix}, \quad S = \begin{bmatrix} 5 & 1 & 1 & 1 \\ 1 & 5 & 4 & 2 \\ 1 & 4 & 5 & 2 \\ 1 & 2 & 2 & 5 \end{bmatrix}$

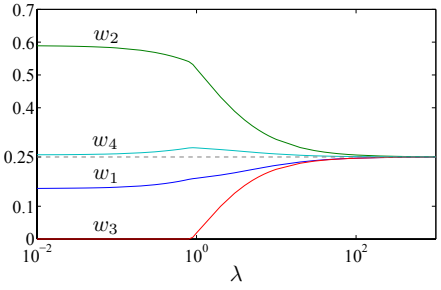


KRI weights

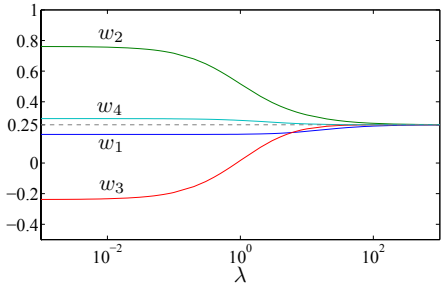


KRR weights

Example 3: $s^T = \begin{bmatrix} 2 & 4 & 3 & 3 \end{bmatrix}, \quad S = \begin{bmatrix} 5 & 1 & 1 & 1 \\ 1 & 5 & 4 & 2 \\ 1 & 4 & 5 & 2 \\ 1 & 2 & 2 & 5 \end{bmatrix}$



KRI weights



KRR weights

affinity goal allots the largest weight to the most similar neighbor x_2, but because x_2 and x_3 are highly similar, the diversity goal forces them to share weight, resulting in x_3 receiving little weight.

One observes from these examples that the KRR weights tend to be smoother than the KRI weights, because the KRI weights are constrained to a probability simplex, while the KRR weights are unconstrained.

6. Generative Classifiers

Generative classifiers provide probabilistic outputs that can be easily fused with other probabilistic information or used to minimize expected misclassification costs. One approach to generative classification given similarities is using the similarities as features to define an n-dimensional feature space, and then fitting standard generative models in that space, as discussed in Section 3. However, this requires fitting generative models with n data points (or less for class-conditional models) in an n-dimensional space. Another generative framework for similarity-based classification termed *similarity discriminant analysis* (SDA) has been proposed that models the class-conditional distributions of similarity statistics (Cazzanti et al., 2008). First we review the basic SDA model, then consider a local variant (Cazzanti and Gupta, 2007) and a mixture model variant that were both designed to reduce bias.

Let X denote a random test sample and x denote a realization of X. Assume that the relevant information about the class label of X is captured by a finite set $\mathcal{T}(X)$ of M descriptive statistics, where the mth descriptive statistic is denoted $\mathcal{T}_m(X)$. In this paper, we take the set of descriptive statistics to be the similarities to class centroids:

$$\mathcal{T}(x) = \{\psi(x,\mu_1), \psi(x,\mu_2), \ldots, \psi(x,\mu_G)\}, \tag{17}$$

where $\mu_g \in \Omega$ is a centroid for the gth class, and G is the number of classes. Although there are many possible definitions of a centroid given similarities, in this paper a class centroid is defined to be the training sample that has maximum sum-similarity to the other training samples of its class. The centroid-based descriptive statistics given by (17) were shown to be overall more effective than other considered descriptive statistics on simulations and a small set of real-data experiments (Cazzanti, 2007).

The classification rule for SDA to minimize the expected misclassification cost is: classify x as the class

$$\hat{y} = \arg\min_{g\in\mathcal{G}} \sum_{h\in\mathcal{G}} C(g,h) P(\mathcal{T}(x)\,|\,Y=h) P(Y=h), \tag{18}$$

where $C(g,h)$ is the cost of classifying a sample as class g if the truth is class h.

To estimate the class-conditional distributions $\{P(\mathcal{T}(x)\,|\,Y=g)\}_{g=1}^{G}$ the SDA model estimates the expected value of the mth descriptive statistic $\mathcal{T}_m(X)$ with respect to the class conditional distribution $P(\mathcal{T}(x)\,|\,Y=g)$ to be the average of the training sample data for each class g:

$$E_{P(\mathcal{T}(x)\,|\,g)}\left(\mathcal{T}_m(X)\right) = \frac{1}{|\mathcal{X}_g|} \sum_{z\in\mathcal{X}_g} \mathcal{T}_m(z), \tag{19}$$

where $\mathcal{X}_g$ is the set of training samples from class g. Given the $G \times G$ constraints specified by (19), the SDA model estimates each class-conditional distribution as the solution to (19) with maximum

entropy, which is the exponential (Jaynes, 1982):

$$\hat{P}(\mathcal{T}(x)\,|\,g) = \prod_{m=1}^{G} \gamma_{gm} e^{\lambda_{gm}\mathcal{T}_m(x)}. \tag{20}$$

Substituting the maximum entropy solution (20) into (18) yields the SDA classification rule: classify x as the class

$$\hat{y} = \arg\min_{g\in\mathcal{G}} \sum_{h\in\mathcal{G}} C(g,h)P(Y=h) \prod_{m=1}^{G} \gamma_{hm} e^{\lambda_{hm}\mathcal{T}_m(x)}.$$

Each pair of parameters $(\lambda_{gm},\gamma_{gm})$ can be separately calculated from the constraints given in (19) by one-dimensional optimization and normalization.

6.1 Reducing SDA Model Bias

The SDA model may not be flexible enough to capture a complicated decision boundary. To address this model bias issue, one could apply SDA locally to a neighborhood of k nearest neighbors for each test point, or learn a mixture SDA model.

In this paper we experimentally compare to *local SDA* (Cazzanti and Gupta, 2007) with local centroid-similarity descriptive statistics given by (17), in which SDA is applied to the k nearest neighbors of a test point, where the parameter k is trained by cross-validation. If any class in the neighborhood has fewer than three samples, there are not enough data samples to fit distributions of similarities, so every λ is assumed to be zero, and the local SDA model is reduced to a simple local centroid classifier. Given a discrete set of possible similarities, local SDA has been shown to be a consistent classifier in the sense that its error converges to the Bayes error under the usual asymptotic assumption that when the number of training samples $n \to \infty$, the neighborhood size $k \to \infty$ but k grows relatively slowly such that $k/n \to 0$ (Cazzanti and Gupta, 2007).

Cazzanti (2007) explored mixture SDA models analogous to Gaussian mixture models (GMM). He fits SDA mixture models, producing the following decision rule:

$$\hat{y} = \arg\min_{g\in\mathcal{G}} \sum_{h\in\mathcal{G}} C(g,h)P(Y=h) \left(\prod_{m=1}^{G} \sum_{l=1}^{c_m} w_{gml}\gamma_{gml} e^{\lambda_{gml}\psi(x,\mu_{ml})} \right),$$

where $\sum_{l=1}^{c_m} w_{ml} = 1$, and $w_{ml} > 0$. The number of components c_m are determined by cross-validation. The component weights $\{w_{gml}\}$ and the component SDA parameters $\{(\lambda_{gml},\gamma_{gml})\}$ are estimated by an expectation-maximization (EM) algorithm, analogous to the EM-fitting of a GMM, except that the centroids are calculated only once (rather than iteratively) at the beginning using a k-medoids algorithm (Hastie et al., 2001). Simulations and a small set of experiments showed that this mixture SDA performed similarly to local SDA, but the model training for mixture SDA was much more computationally intensive.

7. Experiments

We compare eight similarity-based classification approaches: a linear and an RBF SVM using similarities as features, the P-SVM (Hochreiter and Obermayer, 2006), a local SVM (SVM-KNN) (Zhang et al., 2006) and a global SVM using the given similarities as a kernel, local SDA, k-NN, and the three weighted k-NN methods discussed in Section 5: the proposed KRR and KRI weights,

and affinity weights as a control, defined by $w_i = a\psi(x, x_i)$, $i = 1, \ldots, k$, where a is a normalization constant. Results are shown in Table 3 and 4.

For algorithms that require a PSD S, we make S PSD by clip, flip or shift as discussed in Section 2, and pinv for KRR weights. The results in Table 3 are for clip; the experimental differences between clip, flip and shift, and pinv are shown in Table 4 and discussed in Section 7.4.

7.1 Data Sets

We tested the proposed classifiers on eight real data sets[6] representing a diverse set of similarities ranging from the human judgement of audio signals to sequence alignment of proteins.

The *Amazon-47* data set, created for this paper, consists of 204 books written by 47 authors. Each book listed on `amazon.com` links to the top four books that customers bought after viewing it, along with the percentage of customers who did so. We take the similarity of book A to book B to be the percentage of customers who bought B after viewing A, and the classification problem is to determine the author of the book.

The *Aural Sonar* data set is from a recent paper which investigated the human ability to distinguish different types of sonar signals by ear (Philips et al., 2006). The signals were returns from a broadband active sonar system, with 50 target-of-interest signals and 50 clutter signals. Every pair of signals was assigned a similarity score from 1 to 5 by two randomly chosen human subjects unaware of the true labels, and these scores were added to produce a 100×100 similarity matrix with integer values from 2 to 10.

The *Caltech-101* data set (Fei-Fei et al., 2004) is an object recognition benchmark data set consisting of 8677 images from 101 object categories. Similarities between images were computed using the pyramid match kernel (Grauman and Darrell, 2007) on SIFT features (Lowe, 2004). Here, the similarity is PSD.

The *Face Rec* data set consists of 945 sample faces of 139 people from the NIST Face Recognition Grand Challenge data set.[7] There are 139 classes, one for each person. Similarities for pairs of the original three-dimensional face data were computed as the cosine similarity between integral invariant signatures based on surface curves of the face (Feng et al., 2007). The original paper demonstrated comparable results to the state-of-the-art using these similarities with a 1-NN classifier.

The *Mirex07* data set was obtained from the human-rated, fine-scale audio similarity data used in the MIREX 2007 Audio Music Similarity and Retrieval[8] task. Mirex07 consists of 3090 samples, divided roughly evenly among 10 classes that correspond to different music genres. Humans judged how similar two songs are on a 0–10 scale with 0.1 increments. Each song pair was evaluated by three people, and the three similarity values were averaged. Self-similarity was assumed to be 10, the maximum similarity. The classification task is to correctly label each song with its genre.

The *Patrol* data set was collected by Driskell and McDonald (2008). Members of seven patrol units were asked to name five members of their unit; in some cases the respondents inaccurately named people who were not in their unit, including people who did not belong to any unit. Of the original 385 respondents and named people, only the ones that were named at least once were

6. The data sets along with the randomized partitions are available at `http://idl.ee.washington.edu/SimilarityLearning/`.
7. For more information, see `http://face.nist.gov/frgc/`.
8. For more information, see `http://www.music-ir.org/mirex/2007/index.php/Audio_Music_Similarity_and_Retrieval`.

kept, reducing the data set to 241 samples. The similarity between any two people a and b is $(N(a,b)+N(b,a))/2$, where $N(a,b)$ is the number of times person a names person b. Thus, this similarity ϕ has a range $\{0, 0.5, 1\}$. The classification problem is to estimate to which of the seven patrol units a person belongs, or to correctly place them in an eighth class that corresponds to "not in any of the units."

The *Protein* data set has sequence-alignment similarities for 213 proteins from 4 classes,[9] where class one through four contains 72, 72, 39, and 30 samples, respectively (Hofmann and Buhmann, 1997). As further discussed in the results, we define an additional similarity termed *RBF-sim* for the Protein data set: $\psi_{\text{RBF}}(x_i, x_j) = e^{-\|s(x_i)-s(x_j)\|_2}$, where $s(x)$ is the 213×1 vector of similarities with mth component $\psi(x, x_m)$.

The *Voting* data set comes from the UCI Repository (Asuncion and Newman, 2007). It is a two-class classification problem with 435 samples, where each sample is a categorical feature vector with 16 components and three possibilities for each component. We compute the value difference metric (Stanfill and Waltz, 1986) from the categorical data, which is a dissimilarity that uses the training class labels to weight different components differently so as to achieve maximum probability of class separation.

Shown in Figure 1 are the similarity matrices of all the data sets. The rows and columns are ordered by class label; on many of the data sets, particularly those with a fewer number of classes, a block-diagonal structure is visible along the class boundaries, indicated by tick marks. Note that a purely block-diagonal similarity matrix would indicate a particularly easy classification problem, as objects have nonzero similarity only to objects of the same class.

7.2 Other Experimental Details

For each data set, we randomly selected 20% of the data for testing and used the remaining 80% for training. We chose the classifier parameters such as C for the SVM, λ for the KRI and KRR weights, and k for local classifiers by 10-fold cross-validation on the training set, and then used them to classify the held out test data. This process was repeated for 20 random partitions of test and training data, and the statistical significance of the classification error was computed by a one-sided Wilcoxon signed-rank test. Multiclass implementations of the SVM classifiers used the "one-vs-one" scheme (Hsu and Lin, 2002).

Nearest neighbors for local methods were determined using symmetrized similarities[10] $\frac{1}{2}(\psi(x, x_i) + \psi(x_i, x))$. Cross-validation choices are listed in Table 2. These choices were based on recommendations and usage in previous literature, and on preliminary experiments we conducted with a larger range of cross-validation parameters on the Voting and Protein data sets.

7.3 Results

The mean and standard deviation (in parentheses) of the error across the 20 randomized test/training partitions are shown in Table 3. The bold results in each column indicate the classifier with lowest average error; also bolded are any classifiers that were not statistically significantly worse than the classifier with lowest average error.

9. The original data set has 226 samples with 9 classes. As is standard practice with this data set, we removed those classes which contain less than 7 samples.
10. Only Amazon-47 and Patrol are natively asymmetric.

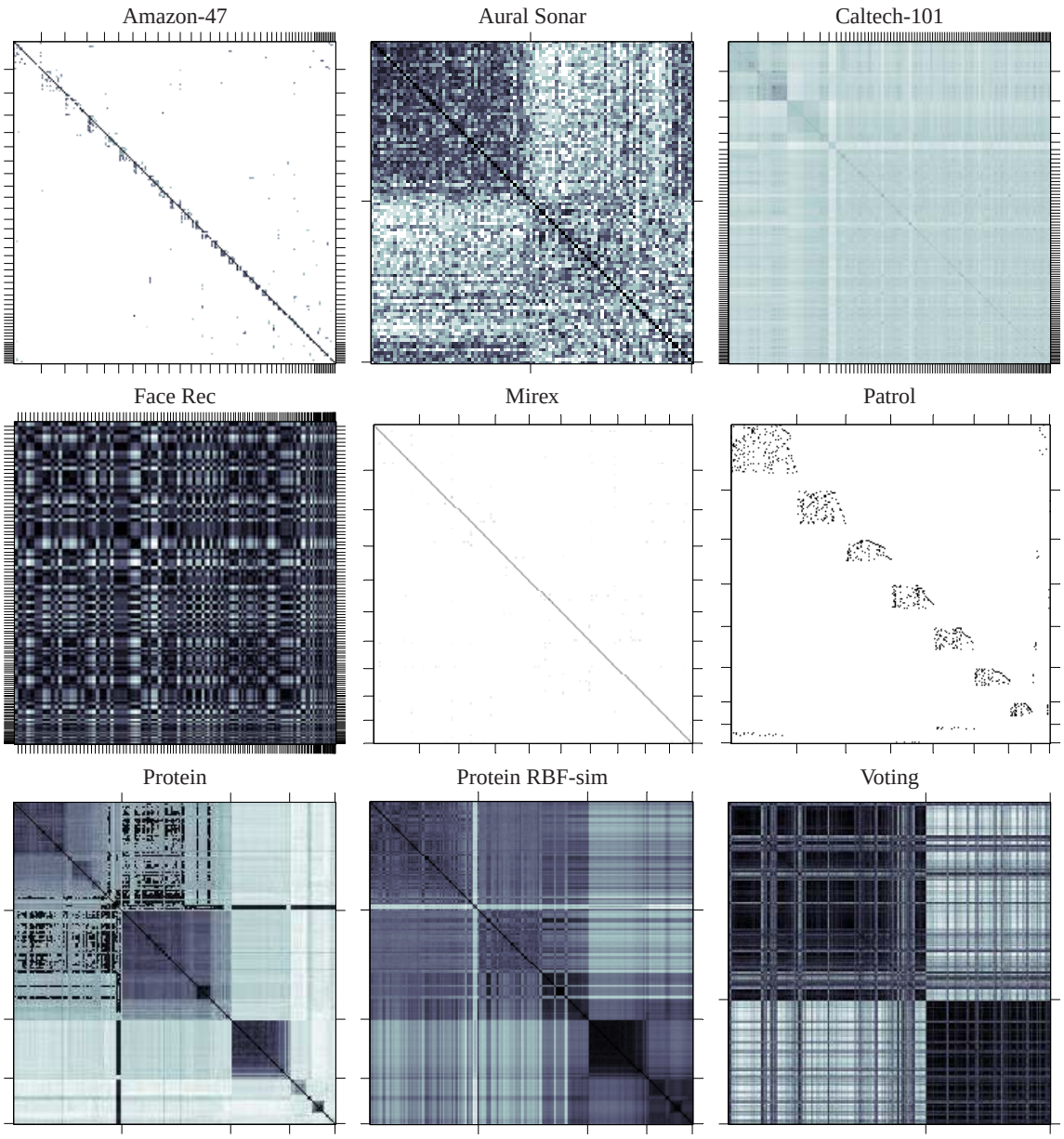


Figure 1: Similarity matrices of each data set with class divisions indicated by tick marks; black corresponds to maximum similarity and white to zero.

The similarity matrices in Figure 1 show that Aural Sonar and Voting exhibit fairly nice block-diagonal structures, indicating that these are somewhat easy classification problems. This is reflected in the relatively low errors across the board and few statistically significant differences in performance. More interesting results can be observed on the more difficult classification problems posed by the other data sets.

All local methods	k:	1, 2, 3, ..., 16, 32, 64, 128
KRR	λ:	$10^{-3}, 10^{-2}, \ldots, 10$
KRI	λ:	$10^{-6}, 10^{-5}, \ldots, 10,$ and 10^{6}
PSVM	ε:	$10^{-4}, 10^{-3}, \ldots, 10$
PSVM	C:	$10^{0}, 10^{1}, \ldots, 10^{4}$
SVM-KNN	C:	$10^{-3}, 10^{-2}, \ldots, 10^{5}$
SVM (linear, clip, flip, shift)	C:	$10^{-3}, 10^{-2}, \ldots, 10^{5}$
SVM (RBF)	C:	$10^{-3}, \ldots, 10$
SVM (RBF)	γ:	$10^{-5}, 10^{-4}, \ldots, 10$

Table 2: Cross-validation Parameter Choices

The Amazon-47 data set is very sparse with at most four non-zero similarities per row. With such sparse data, one might expect a 1-NN classifier to perform well; indeed for the uniform k-NN classifier, the cross-validation chose $k = 1$ on all 20 of the randomized partitions. For all of the local classifiers, the k chosen by cross-validation on this data set was never larger than $k = 3$, and out of the 20 randomized partitions, $k = 1$ was chosen the majority of the time for all the local classifiers. In contrast, the global SVM classifiers perform poorly on this sparse data set. The Patrol data set is the next sparsest data set, and the results show a similar pattern. However, the Mirex data set is also relatively sparse, and yet the global classifiers do well, in particular the SVMs that use similarities as features. The Amazon-47 and Patrol data sets do differ from the Mirex data set in that the self-similarities are not always maximal. Whether (and how) this difference causes or correlates the relative differences in performance is an open question.

The Protein data set exhibits large differences in performance, with the statistically significantly best performances achieved by two of the three SVMs that use similarity as features. The reason that similarities on features performs so well while others do so poorly can be seen in the Protein similarity matrix in Figure 1. The first and second classes (roughly the first and second thirds of the matrix) exhibit a strong interclass similarity, and rows belonging to the same class exhibit very similar patterns, thus treating the rows of the similarity matrix as feature vectors provides good discrimination of classes. To investigate this effect, we transformed the entire data set using a radial basis function (RBF) to create a similarity based on the Euclidean distance between rows of the original similarity matrix, yielding the 213×213 Protein RBF similarity matrix. One sees from Table 3 that this transformation increases the performance of classifiers across the board, indicating that this is indeed a better measure of similarity for this data set. Furthermore, after this transformation we see a complete turnaround in performance: for Protein RBF, the SVMs that use similarities as features are among the worst performers (with P-SVM still performing decently), and the basic k-NN does better than the best classifier given the original Protein similarities.

The Caltech-101 data set is the largest data set, and with 8677 samples in 101 classes, an analysis of the structure of this similarity matrix is difficult. Here we see the most dramatic enhancement in performance (25% lower error) by using the KRR and KRI weights rather than k-NN or the affinity

	Amazon-47		Aural Sonar		Caltech-101	
k-NN	16.95	(4.85)	17.00	(7.65)	41.55	(0.95)
affinity *k*-NN	**15.00**	(4.77)	**15.00**	(6.12)	39.20	(0.86)
KRI *k*-NN (clip)	17.68	(4.75)	**14.00**	(6.82)	30.13	(0.42)
KRR *k*-NN (pinv)	**16.10**	(4.90)	15.25	(6.22)	**29.90**	(0.44)
Local SDA	16.83	(5.11)	17.75	(7.66)	41.99	(0.52)
SVM-KNN (clip)	17.56	(4.60)	**13.75**	(7.40)	36.82	(0.60)
SVM-similarities as kernel (clip)	81.34	(4.77)	**13.00**	(5.34)	33.49	(0.78)
SVM-similarities as features (linear)	76.10	(6.92)	**14.25**	(6.94)	38.18	(0.78)
SVM-similarities as features (RBF)	75.98	(7.33)	**14.25**	(7.46)	38.16	(0.75)
P-SVM	70.12	(8.82)	**14.25**	(5.97)	34.23	(0.95)
	Face Rec		Mirex		Patrol	
k-NN	**4.23**	(1.43)	61.21	(1.97)	**11.88**	(4.42)
affinity *k*-NN	**4.23**	(1.48)	61.15	(1.90)	**11.67**	(4.08)
KRI *k*-NN (clip)	**4.15**	(1.32)	61.20	(2.03)	**11.56**	(4.54)
KRR *k*-NN (pinv)	4.31	(1.86)	61.18	(1.96)	12.81	(4.62)
Local SDA	4.55	(1.67)	60.94	(1.94)	**11.77**	(4.62)
SVM-KNN (clip)	4.23	(1.25)	61.25	(1.95)	**11.98**	(4.36)
SVM-similarities as kernel (clip)	4.18	(1.25)	57.83	(2.05)	38.75	(4.81)
SVM-similarities as features (linear)	4.29	(1.36)	**55.54**	(2.52)	42.19	(5.85)
SVM-similarities as features (RBF)	**3.92**	(1.29)	**55.72**	(2.06)	40.73	(5.95)
P-SVM	**4.05**	(1.44)	63.81	(2.70)	40.42	(5.94)
	Protein		Protein RBF		Voting	
k-NN	29.88	(9.96)	**0.93**	(1.71)	5.80	(1.83)
affinity *k*-NN	30.81	(6.61)	**0.93**	(1.71)	5.86	(1.78)
KRI *k*-NN (clip)	30.35	(9.71)	**1.05**	(1.72)	**5.29**	(1.80)
KRR *k*-NN (pinv)	9.53	(5.04)	**1.05**	(1.72)	**5.52**	(1.69)
Local SDA	17.44	(6.52)	**0.93**	(1.71)	6.38	(2.07)
SVM-KNN (clip)	11.86	(5.50)	**1.16**	(1.72)	**5.23**	(2.25)
SVM-similarities as kernel (clip)	5.35	(4.60)	**1.16**	(1.72)	**4.89**	(2.05)
SVM-similarities as features (linear)	3.02	(2.76)	2.67	(2.12)	5.40	(2.03)
SVM-similarities as features (RBF)	**2.67**	(2.97)	2.44	(2.60)	5.52	(1.77)
P-SVM	**1.86**	(1.89)	**1.05**	(1.56)	**5.34**	(1.72)

Table 3: % Test misclassified averaged over 20 randomized test/training partitions.

k-NN, suggesting that there are highly correlated samples that bias the classification. In contrast, for the Amazon-47, Aural Sonar, Face Rec, Mirex, and Patrol data sets there is only a small win by using the KRI or KRR weights, or a statistically insignificant small decline in performance (we hypothesize this occurs because of overfitting due to the additional parameter λ). On Protein the KRR error is 1/3 the error of the other weighted methods; this is a consequence of using the pinv rather than other types of spectrum modification, as can be seen from Table 4. The other significant difference between the weighting methods is a roughly 10% improvement in average error on Voting by using the KRR or KRI weights. In conclusion, the use of diverse weights may not matter on some data sets, but can be very helpful on certain data sets.

SVM-KNN was proposed by Zhang et al. (2006) in part as a way to reduce the computations required to train a global SVM using similarities as a kernel, and their results showed that it performed similarly to a global SVM. That is somewhat true here, but some differences emerge. For the Amazon-47 and Patrol data sets the local methods all do well including SVM-KNN, but the global methods do poorly, including the global SVM using similarities as a kernel. On the other hand, the global SVM using similarities as a kernel is statistically significantly better than SVM-KNN on Caltech-101, even though the best performance on that data set is achieved by a local method (KRR). From this sampling of data sets, we conclude that applying the SVM locally or globally can in fact make a difference, but whether it is a positive or negative difference depends on the application.

Among the four global SVMs (including P-SVM), it is hard to draw conclusions about the performance of the one that uses similarities as a kernel versus the three that use similarities as features. In terms of statistical significance, the SVM using similarities as a kernel outperforms the others on Patrol and Caltech-101 whereas it is the worst on Amazon-47 and Protein, and there is no clear division on the remaining data sets. Lastly, the results do not show statistically significant differences between using the linear or RBF version of the SVM with similarities as features except for small differences on Face Rec and Patrol.

7.4 Clip, Flip, or Shift?

Different approaches to modifying similarities to form a kernel were discussed in Section 2.1. We experimentally compared clip, flip, and shift for the KRI weights, SVM-KNN, and SVM, and flip, clip, shift and pinv for the KRR weights on the nine data sets. Table 4 shows the five data sets for which at least one method showed statistically significantly different results depending on the choice of spectrum modification.

For KRR weights, one sees that the pinv solution is never statistically significantly worse than clip, flip, or shift, which are worse than pinv at least once. For KRI weights, the differences are negligible, but based on average error we recommend clip.

Flip takes the absolute value of the eigenvalues, which is similar to the effect of using SS^T (as discussed in Section 2.1.5), which for an SVM is equivalent to using the SVM on similarities-as-features. Thus it is not surprising that for the Protein data set, which we have seen in Table 3 works best with similarities as features, flip makes a large positive difference for SVM-KNN and SVM. One sees different effects of the spectrum modification on the local methods versus the global SVMs because for the local methods the modification is only done locally.

	KRI					KRR				
	Amazon	Mirex	Patrol	Protein	Voting	Amazon	Mirex	Patrol	Protein	Voting
clip	17.68	61.20	11.56	30.35	5.34	16.22	*61.22*	11.67	*30.35*	5.34
flip	17.56	61.17	11.67	31.28	5.34	16.22	61.12	*12.08*	*30.47*	5.29
shift	17.68	61.25	*13.23*	30.35	5.29	16.34	*61.25*	11.88	*30.35*	5.52
pinv	-	-	-	-	-	16.10	*61.18*	*12.81*	9.53	5.52

	SVM-KNN					SVM				
	Amazon	Mirex	Patrol	Protein	Voting	Amazon	Mirex	Patrol	Protein	Voting
clip	17.56	61.25	11.98	11.86	5.23	81.34	57.83	38.75	5.35	4.89
flip	17.56	61.25	11.88	1.74	5.23	*84.27*	56.34	*47.29*	1.51	4.94
shift	17.56	61.25	11.88	*30.23*	5.34	77.68	*85.29*	40.83	*23.49*	5.17

Table 4: Clip, flip, shift, and pinv comparison. Table shows % test misclassified averaged over 20 randomized test/training partitions for the five data sets that exhibit statistically significant differences between these spectrum modifications. If there are statistically significant differences for a given algorithm and a given data set, then the worst score, and scores not statistically better, are shown in italics.

8. Conclusions and Some Open Questions

Similarity-based learning is a practical learning framework for many problems in bioinformatics, computer vision, and those regarding human similarity judgement. Kernel methods can be applied in this framework, but similarity-based learning creates a richer set of challenges because the data may not be natively PSD.

In this paper we explored four different approximations of similarities: clipping, flipping, and shifting the spectrum, and in some cases a pseudoinverse solution. Experimental results show small but sometimes statistically significant differences. Based on the theoretical justification and results, we suggest practitioners clip. Flipping the spectrum does create significantly better performance for the original Protein problem because, as we noted earlier, flipping the spectrum has a similar effect to using the similarities as features, which works favorably for the original Protein problem. However, it should be easy to recognize when flip will be advantageous, modify the similarity as we did for the Protein RBF problem, and possibly achieve even better results. Concerning approximating similarities, we addressed the issue of consistent treatment of training and test samples when approximating the similarities to be PSD. Although our linear solution is consistent, we do not argue it is optimal, and consider this issue still open.

A fundamental question is whether it is more advisable to use the similarities as kernels or features. We showed that the difference for SVMs is in the regularization, and that for similarities-as-kernels generalization bounds can be proven using standard learning theory machinery. However, it is not straightforward to apply standard learning theory machinery to similarities-as-features

because the normal Rademacher complexity bounds do not hold with the resulting adaptive non-independent functions. To address this, we considered splitting the training set into prototypes for similarities-as-features and a separate set to evaluate the empirical risk. Even then, we were only able to show generalization bounds if the number of prototypes grows slowly. Complementary results have been shown for similarities-as-features by Balcan et al. (2008a) and Wang et al. (2007), but further analysis would be valuable. Experimentally, it would be interesting to investigate methods using prototypes and performance as the number of training samples increases, ideally with larger data sets than those used in our experiments.

We proposed design goals of affinity and diversity for weighting nearest neighbors, and suggested two methods for constructing weights that satisfy these design goals. Experimental results on eight diverse data sets demonstrated that the proposed weighted k-NN methods can significantly reduce error compared to standard k-NN. In particular, on the largest data set Caltech-101, the proposed KRI and KRR weights provided a roughly 25% improvement over k-NN and affinity-weighted k-NN. The Caltech-101 similarities are PSD, and it may be that the KRI and KRR methods are sensitive to approximations of the matrix S. Preliminary experiments using the unmodified local S and solving KRI or KRR objective functions using a global optimizer showed an increase in performance but at the price of greatly increased computational cost. Compared to the local SVM (SVM-KNN), the proposed KRR weights were statistically significantly worse on the two-class data sets Aural Sonar and Patrol, but statistically significantly better on both of the highly multi-class data sets, Amazon-47 and Caltech-101.

Overall, the results show that local methods are effective for similarity-based learning. It is tempting from the obvious discrepancy between the performance of local and global methods on Amazon and Patrol to argue that local methods will work best for sparse similarities. However, Mirex is also relatively sparse, and the best methods are global. The Amazon and Patrol data sets are different from the other data sets in that they are the only two data sets with non-maximal self-similarities, and this issue may be the root of the discrepancy. For local methods an open question not addressed here is how to efficiently find nearest-neighbors given only similarities. Some research has been done in this area, for example Goyal et al. (2008) developed methods for fast search with logarithmic query times that depend on how "disordered" the similarity is, where they measure a disorder constant D of a set of samples as the D that ensures that if x_i is the kth most similar sample to x, and x_j is the qth most similar sample to x, then x is among the $D(q+k)$ most similar samples to x_j.

Lastly, we note that similarity-based learning can be considered a special case of graph-based learning (see, for example, Zhu, 2005), where the graph is fully-connected. However, most graph-based learning literature addresses the problem of semi-supervised learning, while the similarity-based learning algorithms discussed in this paper are mainly for supervised learning. We have seen no cross-referential literature between these two fields, and it remains an open question which techniques developed for one problem will be useful for the other problem and vice versa.

Acknowledgments

This work was funded by the United States Office of Naval Research and Intel.

Appendix A.

Proof of Proposition 1: Recall the eigenvalue decomposition $S_{\text{clip}} = U^T \Lambda_{\text{clip}} U$. After removing the zero eigenvalues in Λ_{clip} and their corresponding eigenvectors in U and U^T, one can express $S_{\text{clip}} = \check{U}^T \check{\Lambda}_{\text{clip}} \check{U}$, where $\check{\Lambda}_{\text{clip}}$ is an $m \times m$ diagonal matrix with m the number of nonzero eigenvalues and $\check{U}$ an $m \times n$ matrix satisfying $\check{U}\check{U}^T = I$. The vector representation of the training samples implicitly used via S_{clip} is $X = \check{\Lambda}_{\text{clip}}^{1/2} \check{U}$. Given test similarity vector s, the least-squares solution to the equation $X^T x = s$ is $x = \left(XX^T\right)^{-1} Xs$. Let $\tilde{s}$ be the vector of the inner products between the embedded test sample x and the embedded training samples X, then

$$\tilde{s} = X^T x = X^T \left(XX^T\right)^{-1} Xs = \check{U}^T \check{U} s = U^T M_{\text{clip}} U s = P_{\text{clip}} s. \qquad \blacksquare$$

The proofs of the generalization bounds of Theorem 1 and Theorem 2 rely on bounding the Rademacher complexity of the function classes F_S and F_I, respectively. We provide the definition of Rademacher complexity here for convenience.

Definition 1 (Rademacher Complexity) *Suppose $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ are samples drawn independently from a distribution on Ω, and let F be a class of functions mapping from Ω to $\mathbb{R}$. Then, the Rademacher complexity of F is*

$$\mathcal{R}_{\mathcal{X}}(F) = E_{\sigma,\mathcal{X}} \left(\sup_{f \in F} \left| \frac{2}{n} \sum_{i=1}^{n} \sigma_i f(X_i) \right| \right),$$

where $\sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ is a set of independent uniform $\{\pm 1\}$-valued random variables[11] *such that $P(\sigma_i = 1) = P(\sigma_i = -1) = 1/2$ for all i.*

The following lemma establishes that for a class of bounded functions F, the generalization error for any $f \in F$ is bounded above by a function of $\hat{R}_{\mathcal{D}}(f, L)$ and $\mathcal{R}_{\mathcal{D}}(F)$.

Lemma 1 (Bartlett and Mendelson, 2002, Theorem 7) *Suppose (X, Y) and the elements of $\mathcal{D}$ are drawn i.i.d. from a distribution on $\Omega \times \{\pm 1\}$. Let F be a class of bounded real-valued functions defined on Ω such that $\sup_{f \in F} \sup_{x \in \Omega} |f(x)| < \infty$. Suppose $L : \mathbb{R} \to [0,1]$ is Lipschitz with constant C and satisfies $L(a) \geq I_{\{a \leq 0\}}$. Then with probability at least $1 - \delta$ with respect to $\mathcal{D}$, every function in F satisfies*

$$P(Y f(X) \leq 0) \leq \hat{R}_{\mathcal{D}}(f, L) + 2C\mathcal{R}_{\mathcal{D}}(F) + \sqrt{\frac{\ln(2/\delta)}{2n}}.$$

For the proof of Theorem 1, we also require the following bound on the Rademacher complexity of kernel methods.

Lemma 2 (Bartlett and Mendelson, 2002, Lemma 22) *Suppose the elements of $\mathcal{D}$ are drawn i.i.d. from a distribution on $\Omega \times \{\pm 1\}$. Let F_K denote the set of functions $\left\{ f(x) = \sum_i \alpha_i K(x, X_i) \,\middle|\, \sum_{i,j} \alpha_i \alpha_j K(X_i, X_j) \leq \beta^2 \right\}$, then by Jensen's inequality,*

$$\mathcal{R}_{\mathcal{D}}(F_K) \leq 2\beta \sqrt{\frac{E\left(K(X,X)\right)}{n}}.$$

11. Such random variables are called *Rademacher random variables* and their distribution is called the *Rademacher distribution.*

Proof of Theorem 1: Theorem 1 is an application of Lemma 1 and 2 for the function class $F_S = \left\{ f(s) = w^T s \,\middle|\, w^T S w \leq \beta^2 \right\}$. By replacing the kernel function K in Lemma 2 with ψ, and noticing $E\left(\psi(X,X)\right) \leq \kappa^2$ since $\psi(a,a) \leq \kappa^2$ for all $a \in \Omega$, we have $\mathcal{R}_{\mathcal{D}}(F_S) \leq 2\beta\kappa\sqrt{1/n}$. It can be verified that the function class F_S is bounded as $|f(s)| \leq \beta\kappa$ for all $f \in F_S$, and thus we can apply Lemma 1. Noting that L_t is Lipschitz with $C = 1$ completes the proof. ■

Proof of Theorem 2: Recall that $\tilde{s} = \begin{bmatrix} \psi(x,\tilde{x}_1) & \dots & \psi(x,\tilde{x}_m) \end{bmatrix}^T$ where $\{(\tilde{x}_1,\tilde{y}_1),\dots,(\tilde{x}_m,\tilde{y}_m)\} \subseteq \mathcal{D}$ is a subset of the training data and $\widetilde{\mathcal{D}} = \mathcal{D} \backslash \{(\tilde{x}_1,\tilde{y}_1),\dots,(\tilde{x}_m,\tilde{y}_m)\}$ is the remaining training set. It is tempting to apply Lemma 2 with the linear kernel, but in this case, this does not satisfy the definition of the function class $F_I = \left\{ f(\tilde{s}) = w^T \tilde{s} \,\middle|\, w^T w \leq \beta^2 \right\}$ defined on these prototypes. The following bound on the Rademacher complexity mirrors that of Bartlett and Mendelson (2002, Lemma 22), but requires an important modification noted below:

$$
\begin{aligned}
\mathcal{R}_{\widetilde{\mathcal{D}}}(F_I) &= E_{\widetilde{\mathcal{D}},\sigma}\left(\sup_{f \in F_I} \left| \frac{2}{n} \sum_{i=1}^{n-m} \sigma_i f(\tilde{s}_i) \right| \right) \\
&\leq \frac{2}{n} E_{\widetilde{\mathcal{D}},\sigma}\left(\sup_{\|w\|_2 \leq \beta} \left| w^T \left(\sum_{i=1}^{n-m} \sigma_i \tilde{s}_i \right) \right| \right) \\
&\overset{(a)}{\leq} \frac{2}{n} E_{\widetilde{\mathcal{D}},\sigma}\left(\beta \left\| \sum_{i=1}^{n-m} \sigma_i \tilde{s}_i \right\|_2 \right) \\
&= \frac{2\beta}{n} E_{\widetilde{\mathcal{D}},\sigma}\left(\sqrt{\sum_{i,j} \sigma_i \sigma_j \tilde{s}_i^T \tilde{s}_j} \right) \\
&\overset{(b)}{\leq} \frac{2\beta}{n} \sqrt{\sum_{i,j} E_{\widetilde{\mathcal{D}},\sigma}\left(\sigma_i \sigma_j \tilde{s}_i^T \tilde{s}_j \right)} \\
&= \frac{2\beta}{n} \sqrt{\sum_{i=1}^{n-m} E_{\widetilde{\mathcal{D}}}\left(\tilde{s}_i^T \tilde{s}_i \right)} \\
&= \frac{2\beta}{n} \sqrt{\sum_{i=1}^{n-m} \sum_{j=1}^{m} E_{\widetilde{\mathcal{D}}} \psi^2(x_i,\tilde{x}_j)} \\
&\leq 2\beta\kappa^2 \sqrt{\frac{m}{n} - \frac{m^2}{n^2}} \\
&\leq 2\beta\kappa^2 \sqrt{\frac{m}{n}}
\end{aligned}
$$

where (a) follows from the Cauchy-Schwarz inequality[12] and (b) follows from Jensen's inequality.

It can be verified that the function class is bounded as $|f(\tilde{s})| \leq \beta\kappa^2\sqrt{m}$ for all $f \in F_I$, and thus we can apply Lemma 1. As before, noting that L_t is Lipschitz with $C = 1$ completes the proof. ■

The proof of Theorem 2 illustrates why using similarities as features has a poorer guarantee on the generalization than using similarities as a kernel. Specifically, the function class corresponding

12. Note that in the proof of Theorem 1 and in Bartlett and Mendelson (2002, Lemma 22) the Cauchy-Schwartz inequality is applied in the RKHS space whereas here it is applied in $\mathbb{R}^m$.

to regularization on $w^T w$ is too large. Of course, this flexibility can be mitigated by using only a set of m prototypes whose size grows as $o(n)$, which can be seen as an additional form of capacity control.

References

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Molecular Biology*, 215(3):403–410, Oct. 1990.

A. Asuncion and D. J. Newman. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

M.-F. Balcan, A. Blum, and N. Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1–2):89–112, Aug. 2008a.

M.-F. Balcan, A. Blum, and N. Srebro. Improved guarantees for learning via similarity functions. In *Proc. Ann. Conf. Learning Theory*, 2008b.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *J. Machine Learning Research*, 3:463–482, Nov. 2002.

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. and Machine Intel.*, 24(4):509–522, April 2002.

I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York, 2nd edition, 2005.

L. Cazzanti. *Generative Models for Similarity-based Classification*. PhD thesis, Dept. of Electrical Engineerng, University of Washington, 2007.

L. Cazzanti and M. R. Gupta. Local similarity discriminant analysis. In *Proc. Intl. Conf. Machine Learning*, 2007.

L. Cazzanti, M. R. Gupta, and A. J. Koppal. Generative models for similarity-based classification. *Pattern Recognition*, 41(7):2289–2297, July 2008.

J. Chen and J. Ye. Training SVM with indefinite kernels. In *Proc. Intl. Conf. Machine Learning*, 2008.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

J. E. Driskell and T. McDonald. Identification of incomplete networks. Technical report, Florida Maxima Corporation, 2008.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2nd edition, 2001.

L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, 2004.

S. Feng, H. Krim, and I. A. Kogan. 3D face recognition using Euclidean integral invariants signature. In *Proc. IEEE Workshop Statistical Signal Processing*, 2007.

M. P. Friedlander and M. R. Gupta. On minimizing distortion and relative entropy. *IEEE Trans. Information Theory*, 52(1):238–245, Jan. 2006.

I. Gati and A. Tversky. Representations of qualitative and quantitative dimensions. *J. Experimental Psychology: Human Perception & Performance*, 8(2):325–340, April 1982.

I. Gati and A. Tversky. Weighting common and distinctive features in perceptual and conceptual judgments. *Cognitive Psychology*, 16(3):341–370, July 1984.

R. L. Goldstone and A. Kersten. *Comprehensive Handbook of Psychology*, volume 4, chapter 22: Concepts and Categorization, pages 599–621. Wiley, New Jersey, 2003.

N. Goyal, Y. Lifshits, and H. Schütze. Disorder inequality: A combinatorial approach to nearest neighbor search. In *Proc. ACM Symposium Web Search and Data Mining*, 2008.

T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. In *Advances in Neural Information Processing Systems*, 1998.

T. Graepel, R. Herbrich, B. Schölkopf, A. Smola, P. Bartlett, K.-R. Müller, K. Obermayer, and R. Williamson. Classification on proximity data with LP–machines. In *Proc. Intl. Conf. Artificial Neural Networks*, 1999.

K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *J. Machine Learning Research*, 8:725–760, April 2007.

M. R. Gupta, R. M. Gray, and R. A. Olshen. Nonparametric supervised learning by linear interpolation with maximum entropy. *IEEE Trans. Pattern Anal. and Machine Intel.*, 28(5):766–781, May 2006.

B. Haasdonk. Feature space interpretation of SVMs with indefinite kernels. *IEEE Trans. Pattern Anal. and Machine Intel.*, 27(4):482–492, April 2005.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2001.

N. J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, May 1988.

S. Hochreiter and K. Obermayer. Support vector machines for dyadic data. *Neural Computation*, 18(6):1472–1510, June 2006.

T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Trans. Pattern Anal. and Machine Intel.*, 19(1):1–14, Jan. 1997.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Networks*, 13(2):415–425, March 2002.

E. T. Jaynes. On the rationale for maximum entropy methods. *Proc. IEEE*, 70(9):939–952, Sept. 1982.

T. Knebel, S. Hochreiter, and K. Obermayer. An SMO algorithm for the potential support vector-machine. *Neural Computation*, 20(1):271–287, Jan. 2008.

J. Laub and K.-R. Müller. Feature discovery in non-metric pairwise data. *J. Machine Learning Research*, 5:801–808, July 2004.

J. Laub, V. Roth, J. M. Buhmann, and K.-R. Müller. On the information and representation of non-Euclidean pairwise data. *Pattern Recognition*, 39(10):1815–1826, Oct. 2006.

L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *J. Computational Biology*, 10 (6):857–868, 2003.

H.-T. Lin and C.-J. Lin. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Technical report, National Taiwan University, March 2003.

D. J. Lipman and W. R. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227 (4693):1435–1441, March 1985.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. Computer Vision*, 60 (2):91–110, 2004.

R. Luss and A. d'Aspremont. Support vector machine classification with indefinite kernels. In *Advances in Neural Information Processing Systems*, 2007.

C. S. Ong, X. Mary, S. Canu, and A. J. Smola. Learning with non-positive kernels. In *Proc. Intl. Conf. Machine Learning*, 2004.

E. Pekalska and R. P. W. Duin. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23(8):943–956, June 2002.

E. Pekalska, P. Paclík, and R. P. W. Duin. A generalized kernel approach to dissimilarity-based classification. *J. Machine Learning Research*, 2:175–211, Dec. 2001.

S. Philips, J. Pitton, and L. Atlas. Perceptual feature identification for active sonar echoes. In *Proc. IEEE OCEANS Conf.*, 2006.

J. C. Platt. Using analytic QP and sparseness to speed training of support vector machines. In *Advances in Neural Information Processing Systems*, 1998.

R. M. Rifkin. *Everything Old is New Again: A Fresh Look at Historical Approaches in Machine Learning*. PhD thesis, MIT, 2002.

V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Trans. Pattern Anal. and Machine Intel.*, 25(12):1540–1551, Dec. 2003.

Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Intl. J. Computer Vision*, 40(2):99–121, Nov. 2000.

S. Santini and R. Jain. Similarity measures. *IEEE Trans. Pattern Anal. and Machine Intel.*, 21(9): 871–883, Sept. 1999.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002.

T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Molecular Biology*, 147(1):195–197, March 1981.

C. Stanfill and D. Waltz. Toward memory-based reasoning. *Comm. ACM*, 29(12):1213–1228, Dec. 1986.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Statistical Society, Series B (Statistical Methodology)*, 58(1):267–288, 1996.

A. Tversky. Features of similarity. *Psychological Review*, 84(2):327–352, July 1977.

A. Tversky and I. Gati. Similarity, separability, and the triangle inequality. *Psychological Review*, 89(2):123–154, March 1982.

L. Wang, C. Yang, and J. Feng. On learning with dissimilarity functions. In *Proc. Intl. Conf. Machine Learning*, 2007.

G. Wu, E. Y. Chang, and Z. Zhang. An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines. Technical report, University of California, Santa Barbara, March 2005.

H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, 2006.

X. Zhu. *Semi-supervised Learning with Graphs*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2005.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, April 2005.