Fusing Similarities and Kernels for Classification

Yihua Chen and Maya R. Gupta
Department of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
{yhchen, gupta}@ee.washington.edu

Abstract – *The problem of fusing indefinite similarity information and positive semidefinite similarity information together for classification is considered. The proposed solution jointly (i) learns a spectrum modification to make the indefinite similarity positive semidefinite, (ii) learns a conic combination of multiple given positive semidefinite kernels, and (iii) learns the parameters of a discriminative classifier. We show that the proposed fusion method can be formulated as a convex optimization problem. This work extends previous work in multiple kernel learning. Though applicable to other kernel methods, the focus is on the support vector machine. Experiments with four real data sets show that the proposed method is consistently among the best performers.*

Keywords: Similarity, indefinite kernel, multiple kernel learning, support vector machine, kernel methods, convex optimization.

1 Introduction

In some applications, there may be multiple possible descriptions of the similarities between data samples. In this paper, we consider fusing such multiple similarities for classification. An example of multiple similarity descriptions arises in the problem of protein classification in computational biology [1, 2]. Pairwise similarities between proteins can be based on protein-protein interactions, genetic interactions, co-participation in a protein complex, Smith-Waterman sequence matching algorithm [3], and other factors. In general, fusing multiple similarities with regard to a particular classification task can provide a task-specific view of the relations between samples, and the performance may be better than with any single description.

A special case is when each similarity satisfies the mathematical properties of a kernel. In that case, one can treat each similarity as a kernel, and fuse the similarities using multiple kernel learning (MKL), in which a linear combination of multiple kernels and the parameters of a discriminative classifier acting on the kernel combination are jointly learned [4]. MKL can be used to fuse heterogeneous descriptions of data samples in the form of multiple kernels. For the above example of protein function prediction, it has been shown that a classifier trained on a conic combination of all the given similarities yielded better classification results than the same classifier trained on any single type of similarities [1, 2].

However, similarities can be indefinite and thus fail to satisfy the properties of a kernel. Learning based on such indefinite similarities arises in many fields such as computational biology, computer vision, information retrieval, and natural language processing. In order to apply kernel methods with indefinite similarities, researchers have considered several ways to approximate an indefinite similarity matrix by a positive semidefinite (PSD) matrix, but the best approximation method depends on the particular problem [5].

In this paper, we investigate fusing an indefinite similarity with multiple kernels to produce a classifier with good generalization. Including indefinite similarities in the framework of kernel fusion provides a more comprehensive picture of the relations between data samples and extends the concept of MKL. We propose a method that jointly (i) learns the spectrum modification on the similarity matrix to make it PSD, (ii) learns the optimal conic combination of the modified similarity matrix and the given multiple kernel matrices, and (iii) learns the parameters of a discriminative classifier acting on this conic combination of PSD matrices. We formulate the proposed fusion method as a convex optimization problem. For this paper, we focus on the support vector machine (SVM) classifier, though the ideas presented can be generalized to other kernel methods.

The rest of the paper is organized as follows. We first review the prior art in adapting kernel methods for similarity-based classification and the literature for MKL in Section 2. Then, in Section 3 we propose a method to jointly fuse a given indefinite similarity with multiple kernels and learn a classifier with good generalization. Experimental results on four real data sets are reported in Section 4. We conclude in Section 5 with a discussion of extensions of this work and some open questions.

2 Background and Related Work

Let Ω denote the sample space, $x_i \in \Omega$, $i = 1, \ldots, n$, denote the n training samples, and y_i, $i = 1, \ldots, n$, their corresponding class labels. For the classification problem considered in this paper, we take as given an $n \times n$ indefinite matrix S of pairwise similarities between the n training samples, m kernel matrices $K_1, \ldots, K_m$ each of size $n \times n$,

and an $n \times 1$ vector y with ith element y_i. Note that we do not assume the samples $\{x_i\}_{i=1}^n$ are given – only their pairwise relationships and their class labels y.

For notational simplicity, we only consider binary classification such that $y_i \in \{\pm 1\}$, $i = 1, \ldots, n$. For a test sample $x \in \Omega$, we take as given an $n \times 1$ vector s of pairwise similarities between x and each of the n training samples, and also m vectors $k_1, \ldots, k_m$ each of size $n \times 1$, whose ith element is the value of the corresponding kernel function on the test sample and the ith training sample, that is, $K_j(x_i, x)$, $j = 1, \ldots, m$. The problem is to train a classifier based on $K_1, \ldots, K_m$, S and y, and estimate the class label $\hat{y}$ for x based on $k_1, \ldots, k_m$, and s.

We first review the approaches to applying kernel methods to indefinite similarities in Section 2.1 (see [5] for a detailed discussion). Then in Section 2.2, we give a brief review of MKL.

2.1 Modify Similarities into Kernels

To use kernel methods with indefinite similarities, one can simply replace the kernel matrix K with the similarity matrix S, and ignore the fact that S is indefinite. However, because the indefinite matrix S is not a kernel matrix, it does not actually correspond to a reproducing kernel Hilbert space (RKHS), and thus one loses the underlying theoretical support and past empirical support for such kernel methods. In practice, the associated optimization problems may become nonconvex, for example, recall the SVM dual problem:

$$\begin{array}{ll} \underset{\alpha}{\text{maximize}} & \mathbf{1}^T\alpha - \dfrac{1}{2}\alpha^T \operatorname{diag}(y) K \operatorname{diag}(y)\alpha \\ \text{subject to} & y^T\alpha = 0,\ 0 \le \alpha \le C\mathbf{1}, \end{array} \tag{1}$$

with variable $\alpha \in \mathbb{R}^n$, where $\mathbf{1}$ is a column vector of ones, and $\le$ denotes component-wise inequality for vectors. The above SVM dual (1) is no longer convex if one replaces K by S.

To retain the full theoretical and practical benefits of kernel methods, one can derive a surrogate kernel matrix K from S, ideally adapting the modification to be effective for the learning problem at hand [5]. Previous approaches have considered different spectrum modifications to make S PSD, including clipping, flipping, and shifting any negative eigenvalues. First, S is made symmetric by taking $\frac{1}{2}(S + S^T)$ if not already so, then S has eigenvalue decomposition $S = U\Lambda U^T$, where U is an orthogonal matrix and Λ is a diagonal matrix of real eigenvalues, that is, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. *Spectrum clip* makes S PSD by clipping all the negative eigenvalues to zero:

$$S_{\text{clip}} = U \operatorname{diag}\left(\max(\lambda_1, 0), \ldots, \max(\lambda_n, 0)\right) U^T.$$

Spectrum clip is mathematically satisfying in that S_{clip} is the nearest PSD matrix to S in terms of the Frobenius norm [6]:

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

where $\succeq$ denotes the generalized inequality with respect to the PSD cone for square matrices.[1] *Spectrum flip* makes S PSD by flipping the sign of the negative eigenvalues:

$$S_{\text{flip}} = U \operatorname{diag}\left(|\lambda_1|, \ldots, |\lambda_n|\right) U^T,$$

which is equivalent to replacing the original eigenvalues of S with its singular values. *Spectrum shift* makes S PSD by shifting the whole spectrum by the minimum amount needed to make it PSD:

$$S_{\text{shift}} = U\left(\Lambda + |\min(\lambda_{\min}(S), 0)|\, I\right) U^T,$$

where $\lambda_{\min}(S)$ is the minimum eigenvalue of S, and I is the identity matrix. Spectrum shift only enhances the self-similarities and does not change the similarity between any two different samples.

Rather than modifying the spectrum, a recent paper proposed to learn a K close to S within an extended SVM framework by solving [7]:

$$\begin{array}{ll} \underset{\alpha}{\text{maximize}} & \min\limits_{K \succeq 0} \left(g(\alpha, K) + \rho \|K - S\|_F^2\right) \\ \text{subject to} & y^T\alpha = 0,\ 0 \le \alpha \le C\mathbf{1}, \end{array} \tag{2}$$

where $g(\alpha, K) \triangleq \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T \operatorname{diag}(y) K \operatorname{diag}(y)\alpha$; the variables are $\alpha \in \mathbb{R}^n$ and $K \in \mathbb{R}^{n \times n}$, and $C > 0$ and $\rho > 0$ are the hyperparameters. The problem in (2) is a soft-penalty variant of maximizing the minimum of the objective function of (1) among the PSD matrices close to S.

2.2 Multiple Kernel Learning

MKL enables kernel methods to learn how to combine kernels with different parameters or kernels from different sources for a particular learning problem.

Given m kernel matrices $K_1, \ldots, K_m$, consider the set $\mathcal{K}$ of the PSD matrices that are linear combinations of these m kernel matrices with a fixed trace τ, that is,

$$\mathcal{K} = \left\{ K \succeq 0 \;\middle|\; K = \sum_{i=1}^m w_i K_i,\ \operatorname{tr}(K) = \tau \right\}.$$

Let $\psi(K)$ be the optimal value of the SVM dual problem in (1) for a specific K. Lanckriet et al. proposed to learn an optimal linear combination of $K_1, \ldots, K_m$ by solving [4]:

$$\begin{array}{ll} \underset{K}{\text{minimize}} & \psi(K) \\ \text{subject to} & K \in \mathcal{K}. \end{array} \tag{3}$$

They showed that the problem described by (3) is convex in K and formulated it as a semidefinite program (SDP). By restricting the linear combination $\sum_i w_i K_i$ to be a conic combination such that $w_i \ge 0$, $i = 1, \ldots, m$, they further

[1] For $K \in \mathbb{R}^{n \times n}$, $K \succeq 0$ means that K is PSD and thus implies that K is symmetric.

simplified the problem to the following quadratically constrained quadratic program (QCQP):

$$\begin{aligned} \underset{\alpha,t}{\text{maximize}} \quad & \mathbf{1}^T\alpha - \frac{1}{2}\tau t \\ \text{subject to} \quad & y^T\alpha = 0,\ 0 \le \alpha \le C\mathbf{1}, \\ & \alpha^T \operatorname{diag}(y) K_i \operatorname{diag}(y)\alpha \le \operatorname{tr}(K_i)t \\ & \text{for } i = 1, \ldots, m, \end{aligned} \tag{4}$$

with variables $\alpha \in \mathbb{R}^n$ and $t \in \mathbb{R}$. To make (4) look more like (1), one can re-express (4) by moving the m quadratic inequality constraints into the objective so that the problem becomes to maximize over α the following objective

$$\mathbf{1}^T\alpha - \frac{1}{2}\max_i \left(\frac{\tau}{\operatorname{tr}(K_i)} \alpha^T \operatorname{diag}(y) K_i \operatorname{diag}(y)\alpha \right), \tag{5}$$

with the same constraints as in (1). Although the function in (5) is concave, it is not differentiable, and thus the sequential minimal optimization (SMO) algorithm [8] (an efficient algorithm used to solve (1) for large-scale problems) cannot be applied. Bach et al. showed that (4) is in fact the dual of the following primal problem [9]:

$$\begin{aligned} \underset{\{f_i\}_{i=1}^m, b, \xi}{\text{minimize}} \quad & \frac{1}{2}\left(\sum_{i=1}^m \nu_i \|f_i\|_{\mathcal{H}_i} \right)^2 + C\mathbf{1}^T\xi \\ \text{subject to} \quad & y_j \left(\sum_{i=1}^m f_i(x_j) + b \right) \ge 1 - \xi_j,\ j = 1, \ldots, n, \\ & \xi \ge 0, \end{aligned}$$

where $\nu_i = \sqrt{\operatorname{tr}(K_i)/\tau}$, and $\mathcal{H}_i$ denotes the RKHS associated with K_i, which is the hypothesis space for f_i, where f_i, by the representer theorem [10] takes the form

$$f_i(x) = \sum_{j=1}^n c_{ij} K_i(x_j, x).$$

By adding additional regularization terms to the primal, they derived a dual with a smooth objective and proposed an SMO-like algorithm to solve it efficiently [9].

Recently, two fast algorithms designed for large-scale MKL problems have been proposed in [11] and [12] for a slightly different case where $\mathcal{K}$ is replaced by the convex combination of $K_1, \ldots, K_m$, that is,

$$\mathcal{K}' = \left\{ K = \sum_{i=1}^m w_i K_i \;\middle|\; \mathbf{1}^T w = 1,\ w \ge 0 \right\}.$$

3 Proposed Method for Fusing Indefinite and Positive Semidefinite Similarities

We address the theoretical motivation for the proposed method in Section 3.1. In Section 3.2, we propose to use linear transformation to find a surrogate kernel matrix for the indefinite similarity matrix S. Section 3.3 details the proposed method to fuse an indefinite similarity with multiple kernels for classification, including how to formulate it as a convex optimization problem.

3.1 Theoretical Motivation

The m given kernel matrices $K_1, \ldots, K_m$ already have associated RKHS's $\mathcal{H}_1, \ldots, \mathcal{H}_m$. For the given indefinite similarity matrix S, we would like to find a surrogate kernel matrix K_0 corresponding to an RKHS denoted by $\mathcal{H}_0$. Then we can define a new RKHS $\mathcal{H}$ as the direct sum of $\mathcal{H}_i$, $i = 0, \ldots, m$, that is,

$$\mathcal{H} = \bigoplus_{i=0}^m \mathcal{H}_i,$$

whose associated inner product is defined for some $w_i \ge 0$, $i = 0, \ldots, m$, as

$$\langle a, b \rangle_{\mathcal{H}} = \sum_{i=0}^m w_i \langle a_i, b_i \rangle_{\mathcal{H}_i}$$

for any $a, b \in \mathcal{H}$, where a_i and b_i are the unique components of a and b in $\mathcal{H}_i$, respectively. The goal is to learn a classifier in $\mathcal{H}$ that can generalize better than one trained in any single $\mathcal{H}_i$, $i = 0, \ldots, m$. To achieve this goal, we need to find an effective $\mathcal{H}_0$ and equip the inner product of $\mathcal{H}$ with an optimal set of weights w_i, $i = 0, \ldots, m$.

3.2 Learning the Spectrum Modification

Next we discuss how to effectively find a surrogate kernel matrix K_0 for S. As noted in [5], both spectrum clip and flip can be represented by a linear transformation on S, that is, the modified similarity matrix $\tilde{S}$ can be obtained by letting $\tilde{S} = AS$, where $A = U \operatorname{diag}(a) U^T$ (recall $S = U\Lambda U^T$). For spectrum clip,

$$a_{\text{clip}} = \begin{bmatrix} I_{\{\lambda_1 \ge 0\}} & \cdots & I_{\{\lambda_n \ge 0\}} \end{bmatrix}^T,$$

where $I_{\{\cdot\}}$ is the indicator function, and for spectrum flip,

$$a_{\text{flip}} = \begin{bmatrix} \operatorname{sgn}(\lambda_1) & \cdots & \operatorname{sgn}(\lambda_n) \end{bmatrix}^T.$$

We propose to let the surrogate kernel matrix K_0 be a linear transformation of S such that

$$K_0 = AS = U \operatorname{diag}(a) U^T S = U \operatorname{diag}(a) \Lambda U^T,$$

where the spectrum modification vector a is a variable that is learned from the training data, as detailed in the following subsection.

Expressing the modification of S as a linear transformation helps achieve a *consistent* treatment of training and test samples [5]. Since the classifier is trained with K_0 instead of S, to estimate the class label for a test sample x, we would

like to apply the trained classifier on a modification of s, denoted by an $n \times 1$ vector k_0 that is derived from s in a consistent way as K_0 from S. To this end, we propose to apply the same linear transformation A on s such that $k_0 = As$. This method for modifying the test similarities is consistent in the sense that if any training sample is taken as a test sample, its similarities will be modified in the same way during training and test, in line with the spirit of empirical risk minimization.

3.3 Fusing Similarities and Kernels Using Convex Optimization

The proposed method for fusing an indefinite similarity with multiple kernels extends the primal form of the SVM:

$$\begin{aligned} \underset{c,b,\xi}{\text{minimize}} \quad & \frac{1}{n}\mathbf{1}^T\xi + \eta c^T K c \\ \text{subject to} \quad & \operatorname{diag}(y)(Kc + b\mathbf{1}) \geq \mathbf{1} - \xi,\ \xi \geq 0, \end{aligned} \tag{6}$$

with variables $c \in \mathbb{R}^n$, $b \in \mathbb{R}$ and $\xi \in \mathbb{R}^n$, and regularization parameter $\eta > 0$. We favor framing our proposed method in terms of the primal form of the SVM given by (6) because of its clear interpretation as empirical risk minimization with regularization.

We propose to minimize the empirical risk with regularization simultaneously over the spectrum modification vector $a \in \mathbb{R}^n$, the kernel conic combination weights $w \in \mathbb{R}^m$, and the original SVM variables. To begin with, let

$$\kappa(a, w) = K_0 + \sum_{i=1}^{m} w_i K_i = U \operatorname{diag}(a) \Lambda U^T + \sum_{i=1}^{m} w_i K_i.$$

Then we extend the SVM primal (6) to:

$$\begin{aligned} \underset{c,b,\xi,a,w}{\text{minimize}} \quad & \frac{1}{n}\mathbf{1}^T\xi + \eta c^T \kappa(a, w) c + \gamma h(a) \\ \text{subject to} \quad & \operatorname{diag}(y)\left(\kappa(a, w)c + b\mathbf{1}\right) \geq \mathbf{1} - \xi, \\ & \xi \geq 0,\ \Lambda a \geq 0,\ w \geq 0, \\ & \sum_{i=1}^{m} w_i \operatorname{tr}(K_i) \leq \tau, \end{aligned} \tag{7}$$

where the regularizer $h(a)$ is a convex function of a with regularization parameter γ, and τ is a constant. Besides the empirical risk term, we have two regularizers in the objective of (7). The first regularizer $c^T\kappa(a, w)c$ increases the smoothness of the decision function in $\mathcal{H}$, and the second regularizer $h(a)$ focuses the search area for a. For example, one can choose $h(a) = \|a - a_{\text{clip}}\|_2$ to make a behave more like spectrum clip, or $h(a) = \|a - a_{\text{flip}}\|_2$ to make a behave more like spectrum flip. The trace constraint is a linear inequality constraint on w, which prevents w from growing unbounded. This also guarantees that the inner product in $\mathcal{H}$ is bounded, which is crucial to proving a generalization bound [5, Theorem 1].

It is not trivial to compute the solution to (7). We show that (7) can in fact be formulated as a convex optimization problem. First, let

$$\tilde{c} = U^T c,$$

and

$$\tilde{\kappa}(a, w) = \operatorname{diag}(a)\Lambda + \sum_{i=1}^{m} w_i U^T K_i U,$$

then we can rewrite (7) as

$$\begin{aligned} \underset{\tilde{c},b,\xi,a,w}{\text{minimize}} \quad & \frac{1}{n}\mathbf{1}^T\xi + \eta \tilde{c}^T \tilde{\kappa}(a, w) \tilde{c} + \gamma h(a) \\ \text{subject to} \quad & \operatorname{diag}(y)\left(U\tilde{\kappa}(a, w)\tilde{c} + b\mathbf{1}\right) \geq \mathbf{1} - \xi, \\ & \xi \geq 0,\ \Lambda a \geq 0,\ w \geq 0, \\ & \sum_{i=1}^{m} w_i \operatorname{tr}(K_i) \leq \tau. \end{aligned} \tag{8}$$

Next let

$$z = \tilde{\kappa}(a, w)\tilde{c},$$

and notice that

$$\tilde{c}^T \tilde{\kappa}(a, w) \tilde{c} = z^T \left(\tilde{\kappa}(a, w)\right)^\dagger z \tag{9}$$

due to the fact that $\tilde{\kappa}(a, w)\left(\tilde{\kappa}(a, w)\right)^\dagger \tilde{\kappa}(a, w) = \tilde{\kappa}(a, w)$, where $\left(\tilde{\kappa}(a, w)\right)^\dagger$ is the Moore-Penrose pseudoinverse of $\tilde{\kappa}(a, w)$. Next, we use the following lemma to finish the derivation.

Lemma 1. *Let $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$. Then*

$$\begin{bmatrix} A & b \\ b^T & c \end{bmatrix} \succeq 0$$

if and only if $A \succeq 0$, b is in the range (column space) of A, and $c - b^T A^\dagger b \geq 0$, where $A^\dagger$ is the Moore-Penrose pseudoinverse of A.

Lemma 1 follows directly from [13, p. 44, Theorem 1.20], which states a basic property of the generalized Schur complement.

Lastly, by introducing slack variables u and v, and applying Lemma 1 with (9), we can express (8) as

$$\begin{aligned} \underset{z,b,\xi,a,w,u,v}{\text{minimize}} \quad & \frac{1}{n}\mathbf{1}^T\xi + \eta u + \gamma v \\ \text{subject to} \quad & \operatorname{diag}(y)(Uz + b\mathbf{1}) \geq \mathbf{1} - \xi,\ \xi \geq 0, \\ & \Lambda a \geq 0,\ w \geq 0,\ \sum_{i=1}^{m} w_i \operatorname{tr}(K_i) \leq \tau, \\ & \begin{bmatrix} \tilde{\kappa}(a, w) & z \\ z^T & u \end{bmatrix} \succeq 0,\ h(a) \leq v, \end{aligned} \tag{10}$$

with variables $z \in \mathbb{R}^n$, $b \in \mathbb{R}$, $\xi \in \mathbb{R}^n$, $a \in \mathbb{R}^n$, $w \in \mathbb{R}^m$, $u \in \mathbb{R}$ and $v \in \mathbb{R}$. One can recognize that (10) is a convex optimization problem since it has a linear objective, a set of affine constraints, a linear matrix inequality (LMI)

constraint, and a convex inequality constraint depending on $h(a)$. For regularizers like $\|a - a_{\text{clip}}\|_2$ or $\|a - a_{\text{flip}}\|_2$, the last constraint becomes a second-order cone constraint, and the problem in (10) becomes a convex conic program, which can be efficiently solved by a general-purpose convex conic optimizer such as SeDuMi [14] and SDPT3 [15].

Let $z^\star$, $b^\star$, $a^\star$ and $w^\star$ denote the optimal solution to (10); the resulting linear transformation on S is

$$A = U \operatorname{diag}(a^\star) U^T,$$

and we can recover the optimal $c^\star$ by

$$c^\star = U \left(\tilde{\kappa}(a^\star, w^\star)\right)^\dagger z^\star.$$

For a test sample x, given its similarities $s, k_1, \ldots, k_m$, we classify x as

$$\hat{y} = \operatorname{sgn}\left((c^\star)^T \left(U \operatorname{diag}(a^\star) U^T s + \sum_{i=1}^{m} w_i^\star k_i \right) + b^\star \right).$$

4 Experiments

In this section, we compare the proposed SVM for data fusion with SVMs that use a surrogate kernel for the indefinite similarity formed by spectrum clip, flip or shift, and with SVMs trained individually on one of the given kernel matrices.

4.1 Data Sets

Four real data sets[2] were used for experiments. We ran one experiment each with the Amazon, Aural Sonar, and Protein data sets, and two experiments with the Yeast data set, as detailed below. Figure 1 shows the similarity and kernel matrices for all the samples for each data set. The eigenvalue spectra of the indefinite similarity matrices are shown in Figure 2.

The *Amazon* data set consists of 96 fiction and nonfiction books by 23 different authors, including some authors who write both fiction and nonfiction. The problem is to correctly classify each book based on its similarities to the books in the training set as one of the 36 nonfiction books or one of the 60 fiction books. The similarity between book A and book B is $\frac{1}{2}\left(P(A,B) + P(B,A)\right)$, where $P(A,B)$ is the percentage of customers who bought book A after viewing book B, as reported by `amazon.com`. We created the first two kernels by treating the similarities as features. Let S_{*j} denote the jth column of the similarity matrix S. Kernel 1 is a linear kernel on similarity features such that

$$K_1(x_i, x_j) = \left(S_{*i}\right)^T S_{*j}.$$

Kernel 2 is a Gaussian radial basis function (RBF) kernel on similarity features such that

$$K_2(x_i, x_j) = \exp\left(-\beta \|S_{*i} - S_{*j}\|_2^2\right),$$

with $\beta = 0.1$. One can observe from Figure 1 that the original similarity matrix of this data set is very sparse. We created a third kernel by treating S as the adjacency matrix of a graph and generated a diffusion kernel [16, 17] using the normalized graph Laplacian with parameter $\sigma^2 = 20$.

The *Aural Sonar* data set was developed to investigate the human ability to distinguish different types of sonar signals by ear [18]. Test subjects listened to pairs of sonar returns from a broadband active sonar system and rated the similarity of each pair on a scale of 1 to 5. Each pairwise similarity is the sum of the similarity score of two humans for that pair, producing similarities with integer values in the range 2 to 10. The problem is to classify among the total 100 samples the 50 target-of-interest signals from the 50 clutter signals. We created three kernels. The first one is a linear kernel on similarity features, and the second and third are Gaussian RBF kernels on similarity features with $\beta = 0.01$ and $\beta = 0.1$, respectively.

The *Protein* data set has sequence-alignment similarities for 226 proteins from 9 classes [19]. Here we treat the problem as classifying the two most confusable classes, each of which has 72 samples. Again, we created three kernels. The first one is a linear kernel on similarity features, and the second and third are Gaussian RBF kernels on similarity features with $\beta = 0.1$ and $\beta = 0.05$, respectively.

The *Yeast* data set is taken from [1], where the problem is to predict the functions of yeast proteins. The original data set contains 3588 samples and each sample is a yeast protein sequence. There are 13 classes and some samples belong to several classes due to their multiple functions. To simplify the problem, we choose a subset of 200 samples by selecting the first 100 samples that exclusively belong to class 7 and the first 100 samples that exclusively belong to class 12. The similarity here is the Smith-Waterman E-value. We created three kernels out of the similarity. One is a linear kernel on similarity features and the other two are Gaussian RBF kernels[3] on similarity features with $\beta = 0.01$ and $\beta = 0.001$, respectively. We added a fourth kernel by using the Pfam kernel in [1], which measures the similarities between these yeast proteins in a different way than the Smith-Waterman algorithm. We ran two sets of experiments on this data set. One was to fuse the Smith-Waterman similarity and the three kernels created out of it, and the other was to fuse the Smith-Waterman similarity and the Pfam kernel.

4.2 Experimental Setup

We normalized all the similarity and kernel matrices to the range of $[0, 1]$. For each data set, we randomly partitioned the data 20 times into 20% test and 80% training. For each of the 20 partitions, the parameters of the classifiers were selected by a 10-fold cross-validation on the training set.

[2] These data sets are available at `http://idl.ee.washington.edu/similaritylearning/`.

[3] Since the two Gaussian RBF kernels look highly similar to each other, only one of them is shown in Figure 1.

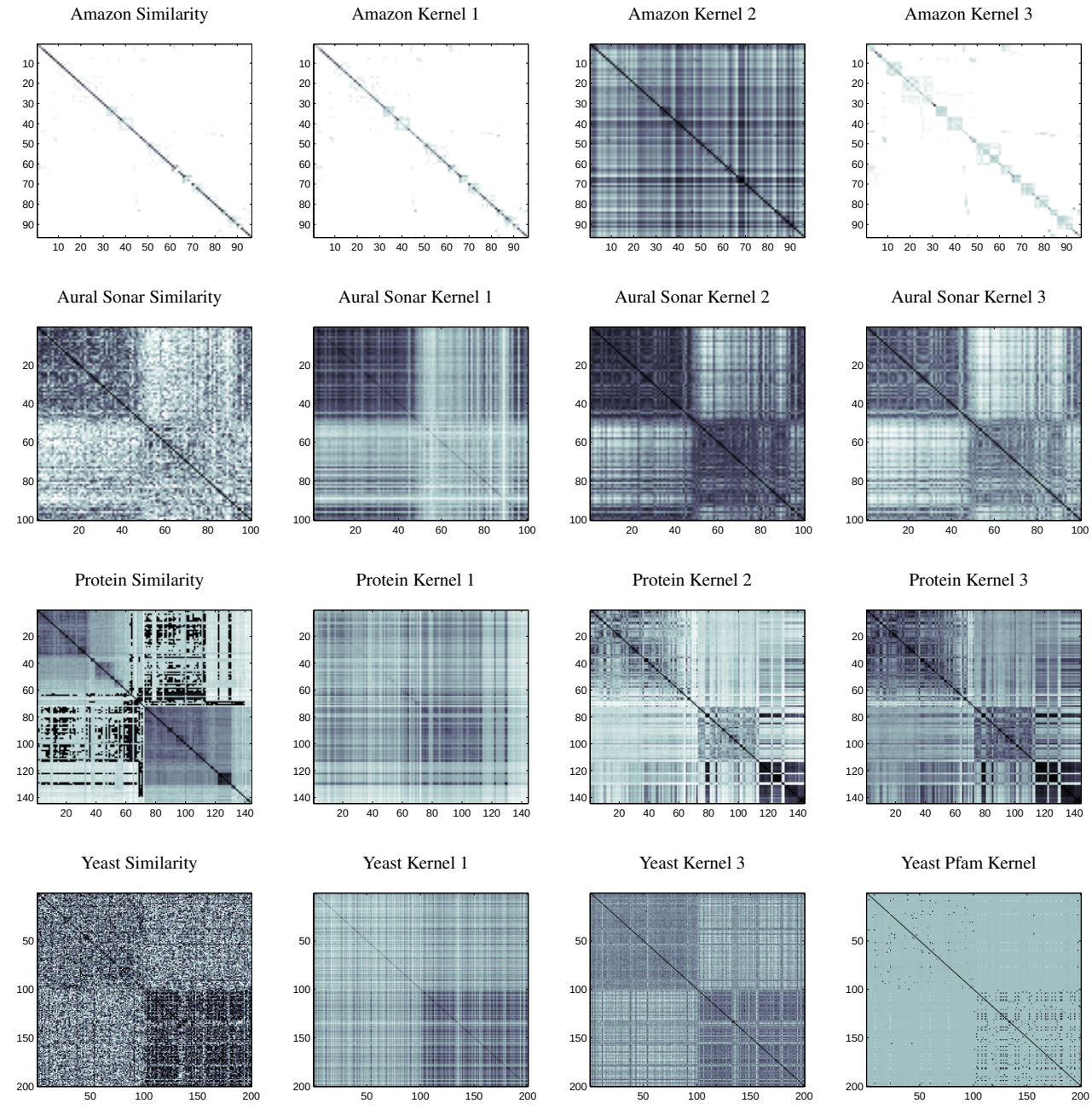

Figure 1: Similarity and kernel matrices of the four data sets. Black corresponds to maximum similarity and white to zero.

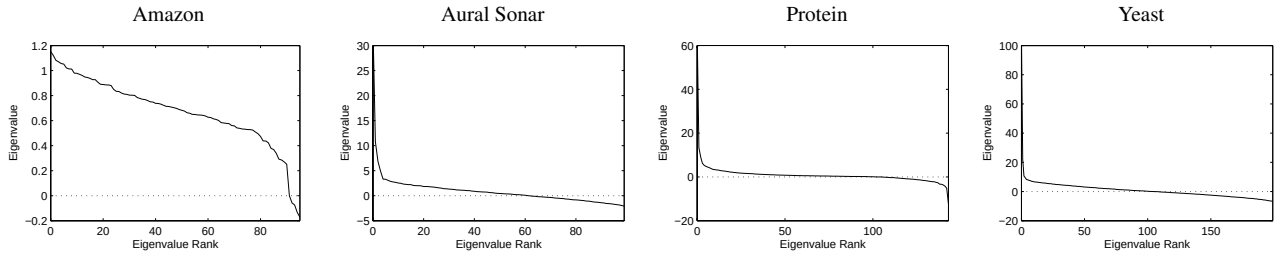

Figure 2: Eigenvalue spectra of the similarity matrices of the four data sets.

For the proposed method, we choose the regularizer of the spectrum modification vector a to be

$$h(a) = \|a - a_{\text{clip}}\|_2,$$

so (10) becomes a convex conic program and we solve it by the semidefinite-quadratic-linear program solver SDPT3 [15]. For all the other SVMs used for comparison, we use the traditional C-SVM, whose dual problem is given in (1).

The regularization parameters η and γ for the proposed method, and the hyperparameter C for C-SVM were cross-validated from the following sets:

$$\begin{aligned}\eta &\in \left\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\right\},\\ \gamma &\in \left\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\right\},\\ C &\in \left\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\right\}.\end{aligned}$$

4.3 Results

The test errors averaged over the 20 randomized test/training partitions are shown in Table 1. For each data set, the lowest average error is boldfaced. Also boldfaced are the results that are not statistically significantly worse than the lowest one based on a one-sided Wilcoxon signed-rank test at the 5% significance level.[4] The results show that the proposed method achieves the lowest average error in three out of the five experiments, and unlike any of the other methods, in all the experiments the proposed method is either the best or not statistically significantly different from the best. Interestingly, one can see from the last column of Table 1 that even when the average error of the SVM using the Pfam kernel alone is twice as high as that of the SVM using spectrum clip on the Smith-Waterman similarity, fusing these two descriptions can in fact improve the classification performance.

We illustrate the fused similarity $\kappa(a^\star, w^\star)$ learned by the proposed method for the Amazon and Yeast (Smith-Waterman and Pfam fusion) data sets in Figure 3, where for each data set we trained the proposed SVM on all the samples using the parameters selected most frequently over the 20 random partitions. For the Amazon data set, the intra-class similarities have been enhanced, as seen by the more obvious block-diagonal structure. For the Smith-Waterman and Pfam fusion, the fused kernel matrix appears to be a less noisy version of the original indefinite similarity matrix shown in Figure 1 with some scattered contributions from the Pfam kernel.

[4] A statistical significance test decides whether a classifier performs consistently better or worse than another classifier, and the result may differ from that indicated by the average performance. For example, the results on the Aural Sonar data set show that the SVM with spectrum flip is statistically significantly worse than the proposed method but the SVM trained on kernel 1 is not, although the average error of the former is less than that of the latter.

5 Discussion and Conclusions

For learning from heterogeneous data, we considered the problem of fusing an indefinite similarity with multiple kernels for classification. The work in this paper extends previous research in MKL. The proposed method is based on empirical risk minimization with regularization, and provides a unified framework to find a surrogate kernel for the indefinite similarity, an optimal set of weights to combine the multiple kernels, and the parameters of the classifier. Experimental evidence suggests that the proposed method can be effective in fusing information and providing a holistic view of the data samples. We consider it worthwhile to investigate regularizers of the spectrum modification vector other than the one used in this paper to possibly find a more effective surrogate kernel for the indefinite similarity.

We formulated the proposed method as a convex optimization problem, which can be efficiently solved by a general-purpose convex conic optimizer. As future work, we would like to find fast algorithms to solve the problem in (10) more efficiently so that for large-scale problems the proposed SVM with similarity fusion can be trained within reasonable time. A possible starting point is to exploit the special structure of the LMI constraint in (10).

The proposed framework can be extended to fusing multiple indefinite similarities by using multiple spectrum modification vectors. However, to introduce weights to these indefinite similarities yet still keep the problem convex remains an open question.

References

[1] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble, "Kernel-based data fusion and its application to protein function prediction in yeast," in *Proc. Pacific Symposium Biocomputing*, 2004.

[2] G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble, "A statistical framework for genomic data fusion," *Bioinformatics*, vol. 20, no. 16, pp. 2626–2635, Nov. 2004.

[3] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J. Molecular Biology*, vol. 147, no. 1, pp. 195–197, March 1981.

[4] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learning Res.*, vol. 5, pp. 27–72, Jan. 2004.

[5] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti, "Similarity-based classification: Concepts and algorithms," *J. Mach. Learning Res.*, vol. 10, pp. 747–776, March 2009.

[6] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear Algebra and its Applications*, vol. 103, pp. 103–118, May 1988.

Table 1: Test errors (in percentage) averaged over 20 test/training partitions. For each data set, the lowest error rate and those not statistically significantly worse are boldfaced.

	AMAZON	AURAL SONAR	PROTEIN	YEAST SW	YEAST SW-PFAM
SVM W/ SIMILARITY FUSION	**9.74**	**12.00**	**1.38**	**7.38**	**6.13**
SVM W/ SPECTRUM CLIP	11.84	**13.00**	8.79	**7.25**	**7.25**
SVM W/ SPECTRUM FLIP	20.00	13.25	4.83	**8.00**	8.00
SVM W/ SPECTRUM SHIFT	17.89	32.75	26.55	43.50	43.50
SVM W/ KERNEL 1	11.05	**13.50**	3.45	**8.38**	14.88
SVM W/ KERNEL 2	12.89	13.75	**1.55**	**8.00**	–
SVM W/ KERNEL 3	**9.21**	13.75	**1.90**	**7.63**	–

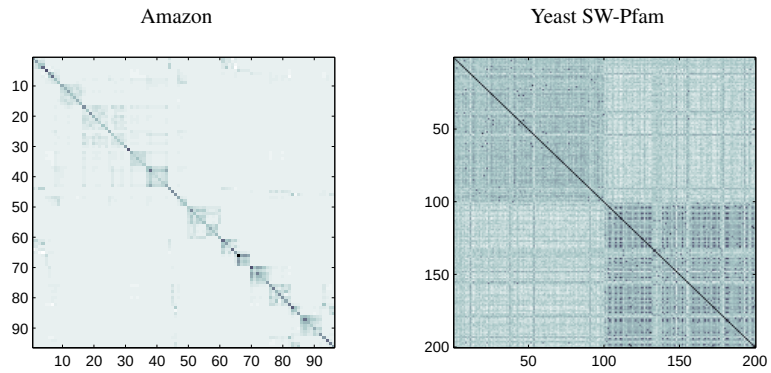

Figure 3: Fused kernel matrices learned from the Amazon and Yeast data sets.

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

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

[9] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. Intl. Conf. Mach. Learning*, 2004.

[10] G. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *J. Math. Anal. and Applications*, vol. 33, no. 1, pp. 82–95, Jan. 1971.

[11] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *J. Mach. Learning Res.*, vol. 7, pp. 1531–1565, July 2006.

[12] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learning Res.*, vol. 9, pp. 2491–2521, Nov. 2008.

[13] R. A. Horn and F. Zhang, "Basic properties of the Schur complement," in *The Schur Complement and Its Applications*. Springer, 2005.

[14] J. F. Strum, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, pp. 625–653, 1999.

[15] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Math. Programming*, vol. 95, no. 2, pp. 189–217, Feb. 2003.

[16] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *Proc. Intl. Conf. Mach. Learning*, 2002.

[17] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Proc. Ann. Conf. Learning Theory*, 2003.

[18] S. Philips, J. Pitton, and L. Atlas, "Perceptual feature identification for active sonar echoes," in *Proc. IEEE OCEANS Conf.*, 2006.

[19] T. Hoffmann and J. M. Buhmann, "Pairwise data clustering by deterministic annealing," *IEEE Trans. Pattern Anal. and Mach. Intel.*, vol. 19, no. 1, pp. 1–14, Jan. 1997.