Distances and (Indefinite) Kernels for Sets of Objects

Adam Woźnica, Alexandros Kalousis, Melanie Hilario
University of Geneva, Computer Science Department
Rue General Dufour 24, 1211 Geneva 4, Switzerland
{woznica, kalousis, hilario}@cui.unige.ch

Abstract

For various classification problems involving complex data, it is most natural to represent each training example as a set of vectors. While several distance measures for sets have been proposed, only a few kernels over these structures exist since it is difficult in general to design a positive semidefinite (PSD) similarity function. The main disadvantage of most existing set kernels is that they are based on averaging, which might be inappropriate for problems where only specific elements of the two sets should determine the overall similarity. In this paper we propose a class of kernels for sets of vectors directly exploiting set distance measures and, hence, incorporating various semantics into set kernels and lending the power of regularization to learning in structural domains where natural distance functions exist. These kernels belong to two groups: (i) kernels in the proximity space induced by set distances and (ii) set distance substitution kernels (non-PSD in general). We report experimental results which show that our kernels compare favorably with kernels based on averaging and achieve results similar to other state-of-the-art methods. At the same time our kernels bring systematically improvement over the naive way of exploiting distances.

1 Introduction

Various practical applications involve a comparison of two sets of vectors of possibly different cardinality. Two machine learning classifiers which can be easily adapted to work in such settings are Support Vector Machines (SVM) and k-Nearest Neighbors (kNN) since they do not require a direct access to the training examples, instead they access the data only by a kernel and a distance function, respectively.

Several distance measures for sets have been defined, imposing different semantics on what is important in determining the overall distance. On the other hand only a few kernels over these structures have been proposed since it is difficult in general to design a valid kernel function which is *positive semidefinite* (PSD). The most popular approach for building a PSD kernel over sets computes affinity measures (e.g. χ^2) between the probability density functions (PDF) (e.g. mixture of Gaussians) estimated from the corresponding two sets, [18, 13, 16, 3]. The standard Cross Product Kernel which amounts to computing the inner product between two means of the corresponding PDFs in the feature space can be seen as a simple example of such kernels. Another approach consists of computing a similarity measure between two subspaces spanned by the elements of the two sets [29].

The above kernels are similar in the sense that the resulting value can be considered as an average of all the similarities of the elements from the two sets. This feature might be inappropriate for applications where only some elements from the two sets determine of the overall similarity. An example of such application is multiple-instance (MI) learning where the task is to learn a concept given positive and negative sets of instances, [8]. In this setting a set is labeled negative if all the instances are negative and is labeled positive if at least one of the instances is positive. For this kind of problems a kernel based on specific pairs of elements from the two sets should have a better performance than a kernel based on averaging. It should be stressed that the appropriate selection of the specific pairs of elements is application dependent and ideally should be guided by domain knowledge, if such exists.

To better tackle problems like the above we propose a class of kernels for sets of vectors which are based on set distance measures. By exploiting the distances over sets we define the corresponding kernel functions which are not based on averaging, instead they only take into account similarities between specific pairs of elements from the two sets. Depending on the actual set distance we are able to impose various semantics into set kernels which can be then used within the regularization framework possibly increasing the predictive performance over methods where distances are used in a standard way (e.g. kNN). These set kernels can be divided into two groups: (i) kernels in the

proximity space induced by set distances where the mapping is defined by a given representation set [25], and (ii) set distance substitution kernels, where the set distances are substituted using a Gaussian RBF similarity measure [11]. The kernels in the second group are not PSD in general; however, encouraged by recent experimental and theoretical results on the application of Support Vector Machines with non-PSD kernels, we are able to use such kernels with the corresponding theoretical framework. We report experimental results which show that the performance of the SVM algorithm with kernels based on specific pairs of elements compare favorably to the SVM with kernels based on averaging. Additionally there is a big improvement for the former in comparison with the kNN algorithm where the set distances are used directly. Finally the performance of our kernels is similar to the state-of-the-art methods.

This paper is organized as follows. Section 2 defines various distances measures for sets and examines their formal properties. These distance measures are used in Section 3 to define (indefinite) kernels on sets: in Subsections 3.1 and 3.2 we present kernels in the proximity space and the Distance Substitution Kernels In Section 4 we review the theoretical framework allowing to use indefinite kernels in SVMs. Experimental results are reported in Sections 5 and 6 whereas in Section 7 we present the related work. We conclude with Section 8.

2 Distances on Sets

The issue that arises when working with sets of objects is how one can use the distance measures defined on $\mathcal{X}$ in order to define distance measures on the power set $2^{\mathcal{X}}$ of $\mathcal{X}$ and under what conditions the set distance measure is a distance or a metric function. A number of different measures have been proposed in the literature for defining distances between sets of objects. We will briefly present some of them.

Before going to the description of distance measures between sets of objects we will briefly review some of the terminology and the definitions used to characterize the different measures explored in this study. A function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$ is a *dissimilarity* function on a nonempty set S, iff it is *reflexive*, i.e. $\forall x \in \mathcal{X} : d(x,x) = 0$; is a *distance* function if it is a dissimilarity function and it is *symmetric*, i.e. $\forall x,y \in \mathcal{X} : d(x,y) = d(y,x)$; is a *metric* if it is a distance function and it is *strict* i.e. $\forall x,y \in \mathcal{X} : d(x,y) = 0 \Rightarrow x = y$ and satisfies the *triangle inequality* i.e. $\forall x,y,z \in \mathcal{X} : d(x,z) \leq d(x,y) + d(y,z)$. If d satisfies all but the triangle inequality it is called a *semi-metric*. It should be mentioned that any distance measure d that is not symmetric can be symmetrized by $d(x,y) := \frac{1}{2}(d(x,y) + d(y,x))$. We also call a distance function isometric to an L^2-norm iff the data can be embedded in a Hilbert space $\mathcal{H}$ by $\phi : \mathcal{X} \to \mathcal{H}$ such that $d(x,y) = \|\phi(x) - \phi(y)\|_{\mathcal{H}}$.

Consider two sets $A = \{a_i\} \subseteq \mathcal{X}$ and $B = \{b_j\} \subseteq \mathcal{X}$. Let $d(.,.)$ be a metric defined on $\mathcal{X}$. The set distance measure D defined on $2^{\mathcal{X}}$ as:

$$D : D(A,B) = f(\{d(a_i,b_j)|(a_i,b_j) \in A \times B\})$$

i.e. is some function of the pairwise distances, $d(a_i,b_j)$, of the set of all pairs $(a_i,b_j) \in A \times B$. A and B should be nonempty and finite sets. Within this framework we can define the following set distance measures. The *(Normalized) Average Linkage*,$\mathrm{D_{AL}}$, is defined as the (normalized) average of all pairwise distances,

$$D_{AL}(A,B) = \frac{\sum_{ij} d(a_i,b_j)}{|A||B|}$$

The *Sum of Minimum Distances*, $\mathrm{D_{SMD}}$, discussed in [7], is the sum of the minimum distances of the elements of the first set to the elements of the second set and vice versa, normalized be the sum of cardinalities of the two sets,

$$D_{SMD}(A,B) = \frac{1}{|A|+|B|}(\sum_{a_i} \min_{b_j}\{d(a_i,b_j)\} + \sum_{b_i} \min_{a_j}\{d(b_i,a_j)\})$$

The *Hausdorff* distance measure, $\mathrm{D_H}$, discussed in [7], is one of the best known distances measures between sets. By definition, two sets A and B are within the *Hausdorff* distance D of each other iff every point of A is within distance D of at least one point of B and vice versa. More formally it is defined as:

$$D_H(A,B) = \max(\max_{a_i}\{\min_{b_j}\{d(a_i,b_j)\}\}, \max_{b_i}\{\min_{a_j}\{d(b_i,a_j)\}\})$$

The *RIBL* distance, $\mathrm{D_{RIBL}}$, is the sum of the minimum distances of the elements of the smaller set to the elements of the larger, [15]. Formally it is defined as:

$$D_{RIBL}(A,B) = \begin{cases} \frac{\sum_{a_i} \min_{b_j}\{d(a_i,b_j)\}}{|B|} & |A| < |B|, \\ \frac{\sum_{b_j} \min_{a_i}\{d(a_i,b_j)\}}{|A|} & |A| \geq |B|. \end{cases}$$

The computation of the above set distance measures is straightforward if one has computed all the pairwise distances among all the pairs of elements defined from the two sets. Another family of more elaborate distance measures is based on the definition of a set of relations $R = \{R_i | R_i \subseteq A \times B\}$ between the two sets. The computation of the distance measure will be based on an $R_i \in R$ that minimizes

the sum of distances computed on the elements that are part of the relation R_i.

In the *Surjections*, $\mathrm{D_S}$, set distance measure the set of relations R consists of all the possible surjections of the larger to the smaller set [7]. In the *Linkings*, $\mathrm{D_L}$, distance measure the set of relations is the set of all possible linkings [7]. A linking is a mapping of one set to the other where all elements of each set participate in at least one pair of the mapping. For *Fair Surjections*, $\mathrm{D_{FS}}$, distance measure the set of relations is the set of all fair surjections, [7]. A surjection is fair if it maps as evenly as possible the elements of the larger set to the elements of the smaller set. For the above set distance measures the final distance of the two sets is defined as the minimum sum, over all R_i, of the distances of the pairs of elements that participate in the surjection, linkings and fair surjections, respectively:

$$D_{S\vee L\vee FS}(A,B) = \frac{\min_{R_i\in R}\sum_{(a_i,b_j)\in R_i} d(a_i,b_j)}{|R_i|}$$

In the *Matchings*, $\mathrm{D_M}$, the set of all possible matchings is considered within which the minimum distance is computed [26]. In a matching each element of the two sets is associated with *at most* one element of the other set. This distance is given by:

$$D_M(A,B) = \min_{R_i\in R}\Big(\sum_{(a_i,b_j)\in R_i} d(a_i,b_j)$$

$$+(|B - R_i(A)| + |A - R_i^{-1}(B)|)\times\frac{M}{2}\Big)$$

where M is the maximum possible distance between two elements. The second term of the sum adds an M/2 penalty for these elements of the A and B that do not participate in the relation R_i. In $\mathrm{D_S}$, $\mathrm{D_L}$ and $\mathrm{D_{FS}}$ the second term of the sum vanishes because all elements of the two sets participate in the relation R_i. $\mathrm{D_M}$ is normalized as

$$D_M(A,B) := \frac{2D_M(A,B)}{D_M(A,B) + (|A| + |B|)/2}$$

It should be noted that all of the set distance measures defined above take values between 0 and 1. Their formal properties are given in Table 1. In the rest of the paper we will focus only on the $\mathrm{D_{SMD}}$, $\mathrm{D_H}$, $\mathrm{D_{RIBL}}$, $\mathrm{D_S}$, $\mathrm{D_L}$, $\mathrm{D_{FS}}$ and $\mathrm{D_M}$ set distance measures since these are the ones based only on specific pairs of elements. The $\mathrm{D_{AL}}$ will be examined only in comparative studies.

Table 1. Characterization of the different set distance measures according to the properties they satisfy. Here we assume that distance measure d on the vector space is a metric. R stands for Reflexive, S for Symmetric, St for strict and T for Triangle inequality.

Distance Measure	R	S	St	T	Type
$\mathrm{D_{AL}}$	-	+	-	-	-
$\mathrm{D_{SMD}}$	+	+	-	-	distance
$\mathrm{D_H}$	+	+	+	+	metric
$\mathrm{D_{RIBL}}$	+	-	-	-	dissimilarity
$\mathrm{D_S}$	+	+	+	-	semi-metric
$\mathrm{D_L}$	+	+	+	-	semi-metric
$\mathrm{D_{FS}}$	+	+	+	-	semi-metric
$\mathrm{D_M}$	+	+	+	+	metric

3 (Indefinite) Kernels on Sets

A kernel is a symmetric function $k : \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ where $\mathcal{X}$ is any set. We call a kernel positive semidefinite (PSD), iff for all $x, y \in \mathcal{X}$, $k(x,y) =< \phi(x), \phi(y) >$ where ϕ is a mapping from $\mathcal{X}$ to a feature space Φ embedded with an inner product.This is equivalent to saying that, for any objects $x_1,\ldots,x_n \in \mathcal{X}$, the induced kernel matrix $\mathbf{K} = (k(x_i,x_j))_{i,j=1}^n$ is PSD, i.e. for any vector $c\in\mathbb{R}^n$ $c^T\mathbf{K}c \geq 0$ (in other words all the eigenvalues of $\mathbf{K}$ are non-negative).

3.1 Set Kernels in the Proximity Space

In the first method the learning instances (sets of objects in our case) are represented in a proximity space [25, 9] [1]. This space is defined by a given set distance measure and a representation set (set of prototypes) of learning instances. More precisely, given a representation set $S = \{s_1,\ldots,s_n\}\subseteq 2^{\mathcal{X}}$ and a set distance measure $D : 2^{\mathcal{X}}\times 2^{\mathcal{X}}\to\mathbb{R}_0^+$ we define a mapping $D(z,S) : 2^{\mathcal{X}}\to\mathbb{R}^n$ as

$$D(z,S) = [D(z,s_1),\ldots,D(z,s_n)]^T$$

where $z\in 2^{\mathcal{X}}$. Since distance measures are nonnegative, all the data examples are projected as points to a nonnegative orthotope of that vector space. The dimensionality of this space is controlled by the size of the set S (usually the full training set).

The construction of the proximity space is justified by the fact that for an object s_i belonging to the same class as z, $D(z,s_i)$ should be small while for an object s_j of different classes $D(z,s_j)$ should be large, resulting in a set of features with possibly high discrimination power [25]. On the other hand if s_i is a characteristic object of a particular class, then the feature $D(x,s_i)$ has a large discrimination power while for s_j being an outlier $D(x,s_j)$ may discriminate poorly. It should be noted that the mapping $D(z,S)$

[1] The proximity space is sometimes classed the "empirical kernel map".

transforms instances to a vector space where any traditional machine learning algorithm (e.g. kNN) may be used.

A direct approach to dissimilarity information between sets given by D leads to the k-nearest neighbor (kNN) method. For a given test instance z this rule is applied to $D(z, S)$ such that z is classified to the most frequent class occurring among the k neighbors in S. For distance measures which are metrics this classifier has a good asymptotic behavior in the Bayes sense [6]. The main disadvantages of this algorithm are large storage and computational requirements as well as sensitivity to outliers. Besides that, if the data is sparsely sampled and the underlying distance measure is not a metric, the classification performance of the kNN algorithm may significantly differ from its asymptotic behavior. It has been argued [25] that the above problems could be alleviated precisely by representing the data in the proximity space. It means that the classification performance in the proximity space is expected to be higher than the one in the "initial" space of sets.

The definition of a set kernel in the proximity space amounts to choosing a set distance measure, D, and a vectorial kernel, k, in the induced space. The resulting Gram matrix of the set kernel $\mathrm{K_P}$ consists of the elements:

$$(\mathbf{K}_{P_D})_{ij} = k(D(x_i, S), D(x_j, S)) \quad (1)$$

It should be stressed that the kernel from equation 1 is PSD iff k is PSD, independently of the characteristics of the corresponding set distance measure, D.

The method for constructing the kernel from equation 1 is similar to the one of kernels based on similarity measures between corresponding distributions and linear subspaces [18, 29], in the sense that sets are first transformed to some other spaces. For kernels based on distributions the "elementary" kernel is defined in a space of (parametric) distributions whereas for the latter all the operations take place in a space of linear subspaces. In our case the sets are mapped to a vectorial space whose dimensionality is given by the cardinality of the representation set.

3.2 Set Distance Substitution Kernels

The next approach exploits the set distance measures by plugging them into the Distance Substitution (DS) Kernels. These kernels were echoed in the literature for some time [2, 1, 12] but have only recently been introduced in a general form in [11].

For any kernel of the form $K(\|x - y\|)$ and for any distance measure D, a DS Kernel can be written in a form $K(x, y) = K(D(x, y))$. Here we will focus on the generalized Gaussian RBF kernel

$$K^D_{DS}(x, y) = e^{-\gamma D(x,y)^2} \quad (2)$$

for $\gamma \in \mathbb{R}^+$ and where $D : 2^S \times 2^S \to \mathbb{R}^+_0$ is at least a distance function, i.e. it is nonnegative, zero-diagonal and symmetric.

Some statements about the PSD-ness of the kernel from equation 2 can be made. According to [11] the following proposition is true:

Proposition 3.1 *For any set distance measure D which is at least a distance function (i.e. it is nonnegative, has zero diagonal and is symmetric) the DS kernel from the equation 2 is PSD for any $\gamma \in \mathbb{R}^+$ if and only if D is isometric to an L^2-norm.*

As an intermediate result of Proposition 3.1 we can state that when D is not a metric (L^2-norms are in particular metrics) the kernel from the equation 2 is not PSD. In particular, from the results presented in Table 1, we can see that for D being a $\mathrm{D_{SMD}}$, $\mathrm{D_{RIBL}}$ (after converting it to be symmetric), $\mathrm{D_S}$, $\mathrm{D_L}$ and $\mathrm{D_{FS}}$ set distance measures the resulting kernel $K_{DS}(x, y)$ is not PSD. For $\mathrm{D_H}$, $\mathrm{D_M}$, even though these distance measures are metrics, the resulting kernels are not PSD since counterexamples can be found.

The discussion of the application of the above non-PSD kernels with the SVM algorithm will be given in Section 4. The kernel from equation 2 can be used to compute a distance measure by $d^2(x, y) = k(x, x) + k(y, y) - 2k(x, y)$. This is possible since $K^D_{DS}(x, x) = 1$ and $0 < K^D_{DS}(x, y) < 1 \; \forall x, y \in \mathcal{X}$, where D is at least a distance function. This distance measure can be then used in the kNN algorithm, however the performance of the resulting classifier is expected to be similar to the performance of the kNN working directly with distances on sets. This is because a given distance measure D is transformed to a kernel using a monotonic function form equation 2 which means that relative similarities of learning instances are not changed.

4 SVMs with Indefinite Kernels

Traditionally the SVM algorithm was applied only if the underlying kernel function is PSD, yielding a convex, local-optimum free optimization problem which is amenable to various efficient optimization algorithms. In the case when the feature space has a finite number of dimensions, the solution to the above optimization problem corresponds to finding a decision boundary of the form of a hyperplane $\{x \in \mathbb{R}^n :< w, x > +b = 0\}$ where w (a normal vector to the hyperplane) and b (the bias) are chosen such that the distance from this decision hyperplane to the nearest training point(s) is maximal.

However, recent experimental [11, 17, 1, 12, 2] and theoretical [10, 23, 17] results state that even a kernel which is non-PSD can be plugged into the SVMs. The theoretical argument supporting the use of indefinite kernels in SVMs

is three-fold. First, the indefinite kernels can be interpreted as inner products in pseudo-Euclidean [10, 25] or more general Krein spaces [23]. Second, SVMs with indefinite kernels can be interpreted as optimal hyperplanes in these indefinite spaces [10]. Third, the uniqueness of the solution of the SVMs can be guaranteed [10].

Several criteria can be examined of how suitable a given non-PSD kernel is for SVM [10]. One of the most useful tests of how difficult it is to obtain a suitable solution with SVMs is to examine the spectrum of a non-PSD kernel matrix. The more negative eigenvalues it has, the more difficult is to obtain good generalization with SVMs. We will use this test to characterize the (indefinite) kernels over sets presented in the Section 3.2.

The other approach to deal with non-PSD kernels is to regularize the kernel matrix by eliminating its negative eigenvalues. Possible approaches to address this problem include removing negative eigenvalues by thresholding, reflecting negative eigenvalues by taking their absolute values or adding a suitable constant to the off-diagonal elements of the squared distance matrix, which results in a Euclidean distance matrix [11]. The main problem with the above schema it that the kernel function is no longer given in analytic form and the testing data should be known beforehand so that it can used for computing the modified kernel matrix.

5 Experiments

In the experiments we want to perform several comparisons of the SVM and kNN algorithms with different kernels and distances over sets on a number of relational benchmark datasets [2]. First, for the various distance measures we want to explore the relative performance of the linear kernel defined in the corresponding proximity spaces and the set DS kernels from Equation 2. These kernels will be used with the SVM method, resulting in the $\mathrm{SVM_P}$ and $\mathrm{SVM_{DS}}$ algorithms, respectively. Second, we want to see how the performance of the SVM with the above kernels compares with the SVM with the following two kernels based on averaging: (i) Bhattacharrya kernel (Bhatta) [16] with the linear kernel as an elementary kernel (ii) the Cross Product Kernel (CPK) with the linear kernel and (iii) the linear kernel in the proximity space induced by the $\mathrm{D_{AL}}$ distance measure; kernels based on averaging are a standard way of tackling classification problems where instances are represented as sets. Third, we are going to examine how the $\mathrm{SVM_P}$ algorithm compares with the kNN algorithm where these distances are used directly. By doing this we establish whether $\mathrm{SVM_P}$ indeed provides an improvement over the simple kNN. Finally, we will try to gain more insight into the proximity space by examining the relative performance of the $\mathrm{SVM_P}$ and a kNN algorithm operating in the same feature space. The reason we use the linear kernel in the experimental setup is to make a fair comparison between the algorithms and to avoid the situation where an implicit mapping given by a nonlinear kernel will influence the results.

For $\mathrm{SVM_P}$ and the SVM with the CPK the regularization parameter C was optimized in an inner 10-fold cross validation loop over the set $C = \{0.1, 1, 10, 50\}$. For $\mathrm{SVM_{DS}}$ the same procedure was used to optimize the width γ and the C parameter over the grid of $\gamma = \{0.1, 1, 5, 10, 20, 30, 40, 50, 60\}$ and $C = \{0.1, 1, 10, 50\}$. In all the kNN algorithms the number of nearest neighbors was optimized over the set $k = \{1, 3, 9\}$.

We will experiment on a number of relational problems: musk, mutagenesis and carcinogenecity. The musk dataset was described in [5] and is a standard MI benchmark dataset; here the goal is to predict the strength of synthetic musk molecules. We worked with both versions (1 and 2) of the dataset. The Mutagenesis dataset was introduced in [27]. The application task is the prediction of mutagenicity of a set of 230 aromatic and heteroaromatic nitrocompounds which constitute the "regression friendly" version of this dataset. Here we do not consider global molecular features and represent each molecule as a set of bonds together with the two adjacent atoms, hence this problem can be considered in a MI setting. This non-global representation of chemical molecules was used e.g. in [21] and it is motivated by the fact that most chemical properties are captured well by local features like atom labels, bond types and functional groups. The last classification problem comes from the Predictive Toxicology Challenge (PTC) and is defined over carcinogenecity properties of chemical compounds [14]. This dataset lists the bioassays of 417 chemical compounds for four type of rodents. Here we present results only for the FR (female rats) version of the problem. We transformed the original dataset (with eight classes) into a binary problem by ignoring EE (equivocal evidence), E (equivocal) and IS (inadequate study) classes, grouping SE (some evidence), CE (clear evidence) and P (positive) in the positive class and N (negative) and NE (no evidence) in the negative one. This dataset is similar to the mutagenesis dataset in the sense that each molecule is represented as a set of bonds with atoms at both ends.

We estimate accuracy using stratified ten-fold cross-validation and control for the statistical significance of observed differences using McNemar's test [20] (sig. level=0.05). The results (with the significance test results in parenthesis) are presented in table 2.

6 Results

In order to compare $\mathrm{SVM_P}$ with $\mathrm{SVM_{DS}}$ we fix a dataset and for each distance measure we compute the significance

[2] A Java implementation can be downloaded from http://cui.unige.ch/~woznica/rel_weka/

Table 2. Accuracy and significance test results on the musk, mutagenesis and carcinogenecity dataset (+ stands for a significant win of the first algorithm in the pair, - for a significant loss and = for no significant difference). The first sign in parethesis corresponds to the comparison of $\mathrm{SVM_P}$ vs. $\mathrm{SVM_{DS}}$, the second to $\mathrm{SVM_P}$ vs. Bhattacharrya kernel from [16], the third to $\mathrm{SVM_P}$ vs. SVM with Cross Product Kernel and the last one to $\mathrm{SVM_P}$ vs. $\mathrm{SVM_P}$ with $\mathrm{K_{P_{AL}}}$.

	musk (version 1)		*musk (version 2)*	
Set Distance	$\mathrm{SVM_P}$	$\mathrm{SVM_{DS}}$	$\mathrm{SVM_P}$	$\mathrm{SVM_{DS}}$
$\mathrm{D_{SMD}}$	96.74 (+)(+)(+)(+)	92.39	92.16 (+)(+)(+)(=)	84.31
$\mathrm{D_H}$	88.04 (=)(+)(=)(=)	88.04	91.18 (=)(+)(=)(=)	84.31
$\mathrm{D_{RIBL}}$	82.61 (+)(+)(=)(=)	52.17	75.49 (+)(+)(=)(-)	60.78
$\mathrm{D_S}$	86.96 (=)(+)(=)(=)	86.96	87.25 (=)(+)(=)(=)	86.27
$\mathrm{D_L}$	90.22 (=)(+)(=)(=)	91.30	87.25 (=)(+)(=)(=)	80.39
$\mathrm{D_{FS}}$	88.04 (=)(+)(=)(=)	89.13	81.37 (=)(+)(=)(=)	81.37
$\mathrm{D_M}$	83.70 (+)(+)(=)(=)	55.43	80.39 (+)(+)(=)(=)	60.78
Bhatta	*51.09*		*61.76*	
CPK	*84.78*		*82.35*	
$\mathrm{K_{P_{AL}}}$	*85.87*		*89.22*	
Def. Accuracy	*51.09*		*61.76*	
	mutagenesis		*carcinogenecity (FR)*	
Set Distance	$\mathrm{SVM_P}$	$\mathrm{SVM_{DS}}$	$\mathrm{SVM_P}$	$\mathrm{SVM_{DS}}$
$\mathrm{D_{SMD}}$	83.51 (+)(+)(=)(=)	69.68	67.52 (=)(+)(=)(=)	65.53
$\mathrm{D_H}$	75.00 (=)(=)(-)(=)	72.87	65.53 (=)(+)(=)(-)	62.40
$\mathrm{D_{RIBL}}$	80.85 (+)(+)(=)(=)	66.49	65.53 (=)(+)(=)(-)	65.53
$\mathrm{D_S}$	89.89 (+)(+)(+)(+)	64.36	65.53 (=)(+)(-)(-)	65.53
$\mathrm{D_L}$	90.42 (+)(+)(+)(+)	73.40	67.81 (=)(+)(=)(=)	65.24
$\mathrm{D_{FS}}$	87.77 (+)(+)(=)(+)	66.49	64.39 (=)(=)(-)(-)	64.96
$\mathrm{D_M}$	89.36 (+)(+)(+)(+)	66.49	64.10 (=)(=)(-)(-)	64.39
Bhatta	*72.34*		*60.11*	
CPK	*84.57*		*67.52*	
$\mathrm{K_{P_{AL}}}$	*79.79*		*68.09*	
Def. Accuracy	*66.49*		*65.53*	

of the observed accuracy differences. The comparison results of the two kernels provide strong evidence that in terms of the predictive accuracy there is an advantage of the linear kernels in the proximity spaces over the set distance substitution kernels. Indeed, we observed that $\mathrm{SVM_{DS}}$ was never significantly better than $\mathrm{SVM_P}$ and it was significantly worse for three, three and six set distance measures for musk 1, 2 and mutagenesis, respectively. In carcinogenecity the differences were not significant. The better performance of $\mathrm{SVM_P}$ could be explained by the fact that the set distance substitution kernels for our distances are not PSD, which means that it is harder for the SVM to find an optimal solution.

To further investigate the above issue we examined the spectra of the corresponding kernel matrices for musk 1, musk 2, mutagenesis and carcinogenecity datasets. The results are visualized in Figure 1. In each of the graphs the x-axis corresponds to different set distance measures which are associated with two characteristics of the kernel matrices (indicated be different colors, where available): smallest negative eigenvalues and proportion of negative eigenvalues to the total number of eigenvalues. Additionally the estimated log accuracies the $\mathrm{SVM_{DS}}$ algorithm using the corresponding set kernels are presented. By looking at the figure 1 several findings can be observed: (i) for all the datasets and for all the set distance measures (with the exception of $FairSurjections$) the negative eigenvalues were very small in magnitude (in the range of 0.001), (ii) low proportion of negative eigenvalues occur for the $Hausdorff$ metric and the AL distance measure [3] and (iii) no clear correlation between the spectra of a given kernel matrix and the estimated accuracy of the corresponding SVM algorithm can be observed. As a result of the above analysis we can state that the worse performance of the $\mathrm{SVM_{DS}}$ algorithm is not necessary caused by the lack of PSD-ness of the distance substitution kernels but that the kernels in the proximity space are better suited for the datasets we experimented

[3] The distance substitution kernel obtained from the AL distance measure ($\mathrm{K_{DS}^{AL}}$) has a similar semantics to the standard Cross Product Kernel, and hence, it is expected that $\mathrm{K_{DS}^{AL}}$ is almost PSD.

with.

It should be mentioned that in musk 1 by using $\mathrm{SVM_P}$ with $\mathrm{D_{SMD}}$ we obtained 96.74 % of accuracy which is as good as the best result reported in the literature.

The next dimension of comparison is the relative performance between the $\mathrm{SVM_P}$ and SVM with kernels based on averaging (the latter, as already mentioned, is a standard approach to tackle set problems). We experimented with the following kernels based on averaging: the Bhattacharrya kernel (Bhatta) [16], the Cross Product Kernel (CPK) and the linear kernel in the proximity space induced by the $\mathrm{D_{AL}}$ distance measure ($\mathrm{K_{P_{AL}}}$). The main point of this comparison is to examine whether there are cases in which different ways of matching the elements of two sets can be more beneficial than the standard averaging which matches everything with everything. From the results it is clear that the relative performance of kernels based on specific pairs of elements and kernels based on averaging depends on the actual application. The strongest advantage of the former is in mutagenesis whereas the opposite trend holds for carcinogenecity. For musk 1 five different instantiations of $\mathrm{SVM_P}$ achieve a higher accuracy than SVM with both CPK and $\mathrm{K_{P_{AL}}}$, nevertheless only for $\mathrm{D_{SMD}}$ the difference is significant. In musk 2 no conclusions can be drawn. It should be noted that the state-of-the-art Bhattacharrya set kernel from [16] performs poorly for all the examined datasets. Overall the choice of the appropriate way of matching the elements of two sets depends on the application and ideally should be guided by domain knowledge, if such exists. Nevertheless, the relative performance of the different kernels provides valuable information about the type of problem we are facing. For example examining mutagenesis and carcinogenecity we see that although they correspond to the same type of classification problem, i.e. classification of graphs, in the latter averaging works better, hinting that the global structure of the molecules is important, whereas in the former averaging performs poorly, indicating that matching specific components of the molecules is more informative. Finally for the musk dataset we see that there is an advantage of the kernels based on matchings of specific elements of two sets over kernels that match everything with everything.

We also compared the performance of a standard kNN algorithm on the standard set distance measures with that of $\mathrm{SVM_P}$ in order to establish whether the latter indeed brings some improvement over the naive way of exploiting distances (these results are listed in table 3). Indeed kNN was never significantly better than $\mathrm{SVM_P}$ and it was significantly worse for two, six, three and two set distance measures for musk 1, 2, mutagenesis and carcinogenecity, respectively. Finally, to exclude the possibility that the improvement comes simply from the use of SVM, we compared $\mathrm{SVM_P}$ with kNN applied in the same proximity spaces ($\mathrm{kNN_P}$). In musk 1 $\mathrm{SVM_P}$ was significantly better (worse) then $\mathrm{kNN_P}$ for one (one) set distance measure; in musk 2 $\mathrm{SVM_P}$ significantly outperformed $\mathrm{kNN_P}$ in three cases whereas in mutagenesis these was no significant difference for all the set distances. In the carcinogenicity dataset $\mathrm{SVM_P}$ is significantly better than $\mathrm{kNN_P}$ in three cases. The better results $\mathrm{SVM_P}$ in musk 2 and carcinogenicity may indicate that for these datasets the bias introduced by SVM is important.

The better performance of $\mathrm{SVM_P}$ and $\mathrm{kNN_P}$ in the proximity space in comparison with the "standard" kNN can be also explained by the fact that working in the proximity space gives a more global view to the data. More precisely the kNN algorithm in a non proximity space examines a single neighborhood of a given instance which is to be classified. On the other hand $\mathrm{SVM_P}$ and $\mathrm{kNN_P}$ in the proximity space have access to all the neighborhoods of points in the representation set. These neighborhoods are precisely given by the instances in the new space. In particular it means that kNN in the proximity space has access to more training points, hence kNN in the initial space would probably archive better results for higher values of the parameter k. To justify the above hypothesis we checked the performance of the standard kNN (where $k = 9$) algorithm in the musk 1 dataset for two distances $\mathrm{D_{RIBL}}$ and $\mathrm{D_M}$, both achieving the poor accuracy of 50 % for $k = 1$. For $k = 9$ the performance increased by more than 25 % reaching 75% and 76.09 %, respectively.

To sum up, in our experiments we observed several findings: (i) kernels in the proximity space ($\mathrm{K_P}$) outperform distance substitution kernels ($\mathrm{K_{DS}}$) – this is due to better suitability of the $\mathrm{K_P}$ (or non-PSD-ness of the $\mathrm{K_{DS}}$), (ii) the appropriate way of matching the elements of two sets depends on the actual application (kernels based on specific pairs vs. kernels based on averaging), (iii) $\mathrm{SVM_P}$ and kNN in the proximity space perform better than kNN working directly on sets (in the proximity space algorithms have a more global view to the data, hence separability is increased).

To situate the performance of our relational learner to other relational learning systems we give the best results reported in the literature on the same benchmark datasets. All the results denote the accuracy and all have been estimated with ten fold cross-validation. The best result for the musk 1 (musk 2) dataset is 96.7 % (96 %) (EM–DD) [30]. In comparison, for $\mathrm{SVM_P}$ with $\mathrm{D_{SMD}}$ we obtained 96.74 % and 92.16 % accuracy, respectively. It should be noted that in [8], using the special purpose MI kernel, the authors achieved 86.4 % and 88 % for musk 1 and musk 2. For the mutagenesis dataset we obtained 87.80 % accuracy while the best result from the literature was 90.4 % [19]. The results for the carcinogenicity dataset are not directly comparable with other results from the literature since differ-

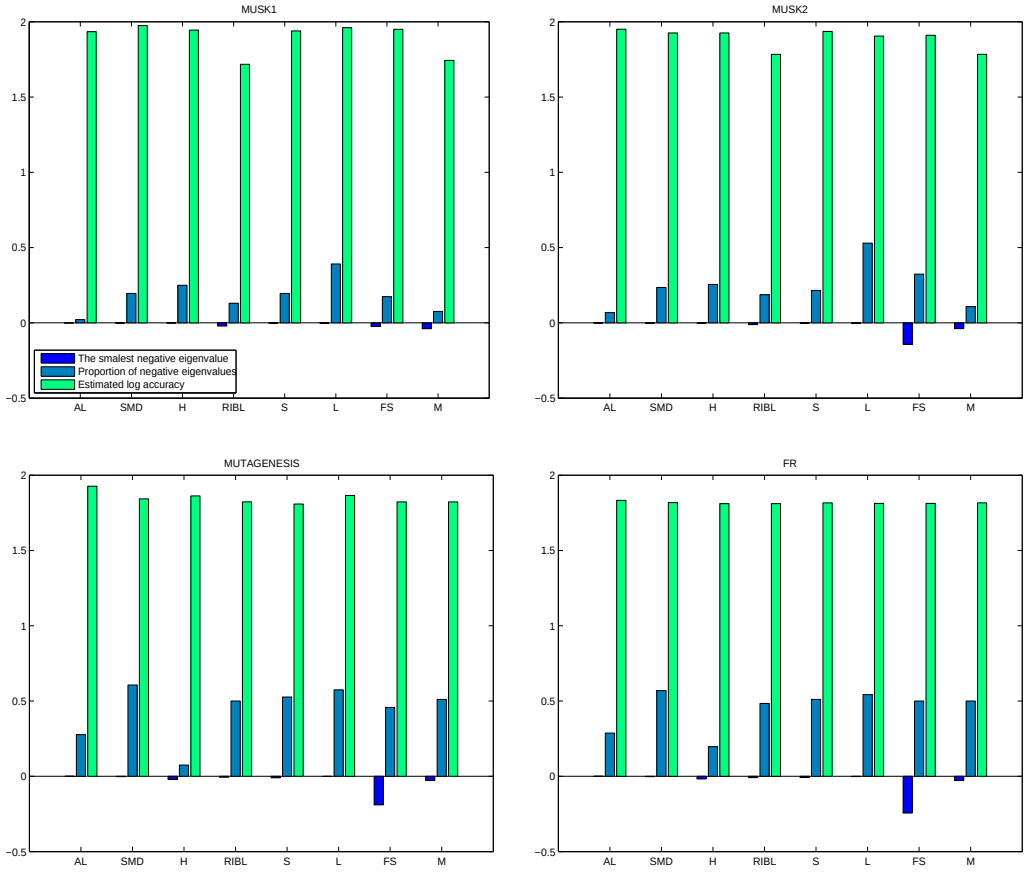

Figure 1. Smallest negative eigenvalues (first bar) and proportion of negative eigenvalues to the total number of eigenvalues (second bar) for musk 1, musk 2, mutagenesis and carcinogenecity datasets for distance substitution kernel matrices which correspond to different set distance measures. Additionally the estimated log accuracies (third bar) of the $\mathrm{SVM}_{\mathrm{DS}}$ algorithm using the set kernels are presented.

ent evaluation metric was used (accuracy instead area under ROC curve). From the results reported above we can see that our kernel-based learner compares favorably with the results achieved by special-purpose algorithms applied to structured data.

7 Related work

As already mentioned the most popular approach for building kernels over sets amounts to computing affinity measures between the PDFs estimated from the corresponding sets. Apart from the Cross Product Kernel, the two most relevant examples are the ones proposed in [16] and [18] where the PDFs are Gaussians and mixture of Gaussians, respectively. A simplified solution was presented in [13] where the corresponding distributions were estimated by histograms and the kernels were build from similarity measures based on histograms. This approach for building set kernels suffers however from several drawbacks. First, it is in general difficult to estimate PDFs in a high-dimensional space [6]. Second, if the underlying PDFs can be estimated well enough, a Bayesian framework would be more appropriate [4]. Last but not least, as we showed in Section 6, the "averaging" property might be inadequate for some applications.

A more general approach for building kernels over sets was recently proposed in [3] where a kernel between (molecular) measures or densities on the space of elements of the two sets is considered. The final kernel is defined as a measure of dispersion of the sum of the corresponding measures. The authors prove that several function that quantify the dispersion of measures through their entropy or through their generalized variance result in a valid PSD kernel. The above kernels, however, are also based on averaging.

The geometrical approach for building set kernels was presented in [29] where the concepts of principal angles be-

Table 3. Accuracy and significance test results on the musk, mutagenesis and carcinogenecity dataset (+ stands for a significant win of the first algorithm in the pair, - for a significant loss and = for no significant difference). The first parenthesis corresponds to the comparison of $\mathrm{SVM_P}$ vs. kNN and the second to $\mathrm{SVM_P}$ vs. $\mathrm{kNN_P}$.

	musk (version 1)			*musk (version 2)*		
Set Distance	$\mathrm{SVM_P}$	kNN	$\mathrm{kNN_P}$	$\mathrm{SVM_P}$	kNN	$\mathrm{kNN_P}$
$\mathrm{D_{SMD}}$	96.74 (+)(+)	80.43	90.22	92.16 (+)(=)	74.51	87.25
$\mathrm{D_H}$	88.04 (=)(=)	81.52	85.87	91.18 (+)(+)	76.47	82.35
$\mathrm{D_{RIBL}}$	82.61 (+)(=)	65.22	84.78	75.49 (+)(=)	60.78	71.57
$\mathrm{D_S}$	86.96 (=)(=)	83.70	90.22	87.25 (+)(=)	73.53	87.25
$\mathrm{D_L}$	90.22 (=)(=)	84.78	83.70	87.25 (+)(+)	73.53	77.45
$\mathrm{D_{FS}}$	88.04 (=)(-)	82.61	96.74	81.37 (=)(=)	77.45	72.55
$\mathrm{D_M}$	83.70 (=)(=)	70.65	81.52	80.39 (+)(+)	60.78	69.61
	mutagenesis			*carcinogenecity (FR)*		
Set Distance	$\mathrm{SVM_P}$	kNN	$\mathrm{kNN_P}$	$\mathrm{SVM_P}$	kNN	$\mathrm{kNN_P}$
$\mathrm{D_{SMD}}$	83.51 (=)(=)	84.04	82.45	67.52 (+)(=)	60.40	62.96
$\mathrm{D_H}$	75.00 (=)(=)	75.53	75.53	65.53 (=)(+)	64.67	57.26
$\mathrm{D_{RIBL}}$	80.85 (=)(=)	76.06	84.57	65.53 (=)(+)	64.96	58.97
$\mathrm{D_S}$	89.89 (+)(=)	80.32	87.23	65.53 (+)(+)	60.11	56.69
$\mathrm{D_L}$	90.42 (+)(=)	82.98	85.64	67.81 (=)(=)	64.10	62.68
$\mathrm{D_{FS}}$	87.77 (=)(=)	83.51	90.42	64.39 (=)(=)	62.39	61.82
$\mathrm{D_M}$	89.36 (+)(=)	71.28	87.77	64.10 (=)(=)	62.96	59.54

tween two linear subspaces are used. Again this kernel can be seen as an averaged similarity of the elements of the two sets.

A specialized kernel for MI problems was proposed in [8] which, in the case of a Gaussian RBF elementary kernel, amounts to the standard Cross Product Kernel.

The most relevant work in the context of DS kernels from Section 3.2 was presented in [4], where set distance measures based on level sets of corresponding PDFs, easier to estimate than the PDFs themselves, are "substituted". In these kernels the "averaging" mechanism is also present. Recently DS kernels based on tangent-distance [12], dynamic-time-wrapping [1] ,Kullback-Leibler divergence [22] and χ^2-distance on histograms [2] have been proposed. On the other hand, [11] presents a general framework for DS kernels, examines their formal properties and shows that better performance of SVMs with these kernels over standard kNN algorithms can be achieved.

Proximity space was first proposed in [28]. However, in [28] the space is induced by means of an asymmetric kernel function. The proximity space defined by dissimilarity measures was considered among others in [9, 25]. Several experimental results were reported for algorithms in the proximity space: SVM was considered in [9, 25], LP machines were examined in [25] whereas [25] proposed Fisher Linear Discriminant to be applied in the proximity space.

8 Conclusions and Future Work

In this paper we defined a class of kernels over sets which directly exploit the set distance measures such that the computation is based only on specific pairs of elements. These set kernels belong to two groups: (i) kernels in the proximity space induced by set distances and (ii) set distance substitution kernels. We reported experimental results which show that the kernels in the proximity space perform significantly better than the set distance substitution kernels. We also argue that when dealing with set problems the standard approaches based on averaging do not necessarily provide the best performance. We can get significant gains in classification performance by focusing on specific types of matchings between elements of the two sets. Finally we show that SVMs with our kernels systematically outperform kNN, where distances are used directly.

In the future we would like to define a new class of kernels over sets which will be directly based on specific pairs of elements from the two sets. This new kernels will be given in a general form as:

$$K \ : \ K(A,B) = g(\{k(a_i,b_j)|(a_i,b_j) \in A \times B\})$$

i.e. it is some function of the pairwise elementary kernels, $k(a_i,b_j)$, of the set of all pairs $(a_i,b_j) \in A \times B$. Our preliminary results show that these kernels perform favorably with the other set kernels proposed in this paper.

Also, in the case of non-PSD kernels, we would like to examine whether regularization of the kernel matrix as sug-

gested in Section 4 will improve the predictive performance of SVMs. In particular by looking at the figure 1 we expect that the regularization will improve the performance of the $FairSurjection$ substitution kernel which (in theory) was the most unsuitable to use within SVMs.

The other research direction is to examine more informative methods of selecting sets of prototypes based on which the proximity space is defined. First, we would like to examine the feature (prototype) weighting methods based on sparse linear hyperplane, [24], and linear SVM applied in the proximity space. Second, a feature prototype selection method based on support vectors of any SVM applied in the same space will be evaluated.

References

[1] C. Bahlmann, B. Haasdonk, and H. Burkhardt. On-line handwriting recognition with support vector machines—a kernel approach. In *Proc. 8th Int. Workshop Front. Handwriting Recognition (IWFHR)*, pages 49–54, 2002.

[2] O. Chapelle, P. Haffner, and V. Vapnik. Svms for histogram-based image classification. *IEEE Transactions on Neural Networks, special issue on Support Vectors*, pages 1055–1064, September 1999.

[3] M. Cuturi, K. Fukumizu, and J.-P. Vert. Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:1169–1198, 2005.

[4] F. Desobry, M. Davy, and W. Fitzgerald. A class of kernels for sets of vectors. In *ESANN 2005*, 2005.

[5] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[6] R. Duda, P. Hart, and D. Stork. *Pattern Classification and Scene Analysis*. John Willey and Sons, 2001.

[7] T. Eiter and H. Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34(2):109–133, 1997.

[8] T. Gärtner, P. Flach, A. Kowalczyk, and A. Smola. Multi-instance kernels. In C. Sammut, editor, *ICML02*. Morgan Kaufmann, July 2002.

[9] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, , and K. Obermayer. Classification on pairwise proximity data. In *In Advances in Neural Information Processing Systems 11*, pages 438–444, 1999.

[10] B. Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492, April 2005.

[11] B. Haasdonk and C. Bahlmann. Learning with distance substitution kernels. In *26th Pattern Recognition Symposium of the German Association for Pattern Recognition (DAGM 2004)*, Tübingen, Germany, 2004. Springer Verlag.

[12] B. Haasdonk and D. Keysers. Tangent distance kernels for support vector machines. In *16th ICPR*, volume 2, pages 864–868. Springer Verlag, 2002.

[13] M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In *Proceedings of AISTATS 2005*, 2005.

[14] C. Helma, R. D. King, S. Kramer, and A. Srinivasan. The predictive toxicology challenge 2000–2001. *Bioinformatics*, 17:107–108, 2001.

[15] T. Horvath, S. Wrobel, and U. Bohnebeck. Relational instance-based learning with lists and terms. *Machine Learning*, 43(1/2):53–80, 2001.

[16] R. Kondor and T. Jebara. A kernel between sets of vectors. In *Proceedings of 20th International Conference on Machine Learning (ICML-2003)*, Washington, DC, 2003.

[17] 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.

[18] S. Lyu. Kernels for unordered sets: the gaussian mixture approach. In *Proceedings of European Conference on Machine Learning (ECML)*, Porto, Portugal, 2005.

[19] P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. Extensions of marginalized graph kernels. In *ICML 2004*, 2004.

[20] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12:153–157, 1947.

[21] S. Menchetti, F. Costa, and P. Frasconi. Weighted decomposition kernels. In *Proceedings of 22nd International Conference on Machine Learning (ICML-2005)*, 2005.

[22] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A kullback-leibler divergence based kernel for svm classification in multimedia applications. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[23] C. S. Ong, X. Mary, S. Canu, and A. J. Smola. Learning with non-positive kernels. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 81, New York, NY, USA, 2004. ACM Press.

[24] E. Pekalska, R. P. Duin, and P. Paclík. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39:189–208, 2006.

[25] E. Pekalska, P. Paclík, and R. P. Duin. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, 2:175–211, 2001.

[26] J. Ramon and M. Bruynooghe. A polynomial time computable metric between point sets. *Acta Informatica*, 37(10):765–780, 2001.

[27] A. Srinivasan, S. Muggleton, R. King, and M. Sternberg. Mutagenesis: ILP experiments in a non-determinate biological domain. In S. Wrobel, editor, *Proceedings of the 4th International Workshop on Inductive Logic Programming*, volume 237, pages 217–232, 1994.

[28] K. Tsuda. Support vector classifier with asymmetric kernel functions. In *ESANN'1999: European Symposium on Artificial Neural Networks*, pages 183–188, 1999.

[29] L. Wolf and A. Shashua. Learning over sets using kernel principal angles. *Journal of Machine Learning Research*, 4:913–931, October 2003.

[30] Q. Zhang and S. A. Goldman. Em-dd: An improved multiple-instance learning technique. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.