Weighted Decomposition Kernels

Sauro Menchetti MENCHETT@DSI.UNIFI.IT
Fabrizio Costa COSTA@DSI.UNIFI.IT
Paolo Frasconi P-F@DSI.UNIFI.IT
Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze, via di S. Marta 3, 50139 Firenze, Italy

Abstract

We introduce a family of kernels on discrete data structures within the general class of decomposition kernels. A weighted decomposition kernel (WDK) is computed by dividing objects into substructures indexed by a selector. Two substructures are then matched if their selectors satisfy an equality predicate, while the importance of the match is determined by a probability kernel on local distributions fitted on the substructures. Under reasonable assumptions, a WDK can be computed efficiently and can avoid combinatorial explosion of the feature space. We report experimental evidence that the proposed kernel is highly competitive with respect to more complex state-of-the-art methods on a set of problems in bioinformatics.

1. Introduction

Statistical learning in structured and relational domains is rapidly becoming one of the central areas of machine learning, boosted by the increasing awareness that the traditional propositional setting lacks expressiveness for modeling many domains of interest. In this paper we focus on supervised learning of discrete data structures driven by several practical problems in bioinformatics that involve classification of sequences (e.g. protein sub-cellular localization) and graphs (e.g. prediction of toxicity or biological activity of chemical compounds).

Starting from the seminal work of Haussler (1999), several researchers have defined convolution and other decomposition kernels on various types of discrete data structures such as sequences (Lodhi et al., 2002; Leslie et al., 2002; Cortes et al., 2004), trees (Collins & Duffy, 2001), and annotated graphs (Gärtner, 2003). Thanks to its generality, decomposition is an attractive and flexible approach for constructing similarity on structured objects based on the similarity of smaller parts. Still, defining a good kernel for practical purposes may be challenging when prior knowledge about relevant features is not sufficient.

At one extreme, it may be desirable to take *all* possible subparts into account. However, in so doing, the dimension of the feature space associated with the kernel can become too large due to the combinatorial growth of the number of distinct subparts with their size. Arguably, unless an extensive use of prior knowledge guides the selection of relevant parts — e.g. as done by Cumby and Roth (2003) using description logics — most dimensions in the feature space will be poorly correlated with the target function and the explosion of features may adversely affect generalization in spite of using large margin classifiers (Ben-David et al., 2002). As observed by many researchers, the problem also manifests itself in the form of a Gram matrix having large diagonal values. Common sense remedies include down-weighting the contribution of larger fragments (Collins & Duffy, 2001) or limiting their size a priori, although in so doing, we could miss some relevant features. A remedy based on kernel transformations is described by Schölkopf et al. (2002). An alternative promising direction that can avoid dimensionality explosion is the generation of relevant features via mining frequent substructures. Methods of this family have been successfully applied to the classification of chemical compounds (Kramer et al., 2001; Deshpande et al., 2003). Other researchers have found that kernels based on *paths* can also be very effective in chemical domains. Graph kernels based on counting label paths produced by random walks have been proposed by Kashima et al. (2003) and later extended by Mahé et al. (2004) to include contextual information. Horváth et al. (2004) have proposed counting the number of common *cyclic* and *tree* patterns in a graph.

At the opposite extreme, one might flatten discrete

Appearing in *Proceedings of the 22^{nd} International Conference on Machine Learning*, Bonn, Germany, 2005.

structures into propositional representations, reducing the number of features at the (possibly severe) cost of losing valuable structural information. One example of this extreme is the use of amino acid composition for protein sequence classification (Hua & Sun, 2001). In this paper, we show how between the two above extreme approaches (taking subparts and flattening) it is possible to explore a useful class of kernels that perform well in practice for both protein sequence and molecule graph classification. A weighted decomposition kernel (WDK) focuses on relatively small parts of a structure, called *selectors*, that are matched according to an equality predicate. The importance of the match is then weighted by a factor that depends on the similarity of the *context* in which the matched selectors occur. In order to introduce a "soft" similarity notion on contexts, we extract attribute frequencies in each context and then apply a kernel on distributions. Suitable options include histogram intersection kernels (Odone et al., 2005) and probability product kernels (Jebara et al., 2004).

The remainder of the paper is organized as follows. In Section 2 we review Haussler's decomposition kernels giving a slightly more flexible definition. In Section 3 we introduce a general class of weighted decomposition kernels and in Section 4 we discuss efficient algorithmic implementations. Finally, in Section 5 we validate the method on several problems in bioinformatics, involving classification of protein sequences and classification of molecules represented as graphs.

2. Decomposition Kernels

We start from some of the definitions and results in (Haussler, 1999) (see also Shawe-Taylor and Cristianini (2004)). An R-decomposition structure on a set X is a triple $\mathcal{R} = \langle \vec{X}, R, \vec{k} \rangle$ where $\vec{X} = (X_1, \ldots, X_D)$ is a D–tuple of non–empty subsets of X, R is a finite parthood relation on $X_1 \times \cdots \times X_D \times X$, and $\vec{k} = (k_1, \ldots, k_D)$ is a D–tuple of positive definite kernel functions $k_d : X_d \times X_d \mapsto \mathbb{R}$. $R(\vec{x}, x)$ is true iff $\vec{x}$ is a tuple of parts for x — i.e. $\vec{x}$ is a decomposition of x. For any $x \in X$, let $R^{-1}(x) = \{\vec{x} \in \vec{X} : R(\vec{x}, x)\}$ denote the multiset of all possible decompositions of x. A decomposition kernel is then defined as the *multiset kernel* between the decompositions:

$$K_{\mathcal{R}}(x, x') \doteq \sum_{\vec{x} \in R^{-1}(x)} \sum_{\vec{x}' \in R^{-1}(x')} \kappa(\vec{x}, \vec{x}') \qquad (1)$$

where we adopt the convention that summations over the elements of a multiset take into account their multiplicity. To compute $\kappa(\vec{x}, \vec{x}')$, kernels on parts are combined by means of operators that need to be closed with respect to kernel positive definiteness. Haussler (1999) proved that combinations based on tensor product (R-convolution kernels) and direct sum are positive definite.

Since decomposition kernels form a rather vast class, the relation R needs to be carefully tuned to different applications in order to characterize a suitable kernel. One commonly used family consists of *all-substructures kernels*, which count the number of common substructures in two decomposable objects. In this case $D = 1$ and $\mathcal{R} = \langle X, R, \delta \rangle$, where $R(x_1, x)$ if x_1 is a substructure of x and δ is the *exact matching kernel*: $\delta(x_1, x_1') = 1$ if $x_1 = x_1'$ and 0 otherwise. Note that in general, computing the equality predicate between x_1 and x_1' may not be computationally efficient as it might require solving a subgraph isomorphism problem (Gärtner et al., 2003). Known kernels that can be reduced to the above form include the spectrum kernel on strings (Leslie et al., 2002), the basic version (with no down-weighting) of co-rooted subtree kernel on trees (Collins & Duffy, 2001) and kernels counting common walks on graphs (Gärtner, 2003).

3. Weighted Decomposition Kernels

3.1. Data Types

We focus on instances from a wide class of annotated graphs. This includes sequences and trees as special cases. No particular restriction needs to be assumed about graph topologies. In particular, we allow the presence of cycles and we can use directed or undirected edges and ordered, unordered or positional adjacency lists. For simplicity, we assume that labels associated with vertices and edges are tuples of atomic attributes. Attributes are organized into classes and can be instantiated for each vertex or edge. So, for example, in a chemical domain we may introduce the class *AtomType* for vertex attributes and write $\mathit{AtomType}(3) = C$ to indicate that vertex 3 in a graph molecule is a carbon atom. In the following we will denote by ξ a generic vertex attribute class and by $\xi(v)$ its value at vertex v. Similarly, we denote by $\xi(u, v)$ the value of an edge attribute of class ξ at edge (u, v). Finally, if x is a graph, we denote by $\xi(x)$ the *value multiset* associated with attribute ξ. In the case of vertex attributes, $\xi(x) = \{\xi(v) : v \in V(x)\}$ where $V(x)$ is the vertex set of x. Similarly, in the case of edge attributes, $\xi(x) = \{\xi(u, v) : (u, v) \in E(x)\}$ where $E(x)$ is the edge set of x.

3.2. Graph Probability Distribution Kernels

In a probability product kernel, a simple generative model is fitted to each example and the kernel between two examples is evaluated by integrating the product

of the two corresponding distributions (Jebara et al., 2004). In this paper we use a discrete version of these kernels, based on multinomial frequencies. Given a graph x and an attribute ξ_i, let $p_i(j)$ be the observed frequency of value j in $\xi_i(x)$. A first type of kernel is defined as

$$k_i(x,x') = \sum_{j=1}^{m_i} p_i(j)^\rho p_i'(j)^\rho \tag{2}$$

where m_i is the number of distinct values for ξ_i. Setting $\rho = 1/2$ we obtain a discrete version of the Bhattacharyya kernel. As an interesting alternative, we may use histogram intersection kernels (Odone et al., 2005) defined as:

$$k_i(x,x') = \sum_{j=1}^{m_i} \min\{p_i(j), p_i'(j)\}. \tag{3}$$

The contributions of multiple attributes can be summed or multiplied, yielding kernels of the form:

$$\kappa(x,x') = \prod_{i=1}^{n}(1+k_i(x,x')) \tag{4}$$

$$\kappa(x,x') = \sum_{i=1}^{n} k_i(x,x') \tag{5}$$

where $k_i(x,x')$ is one of Eq. (2) or Eq. (3). In the case of continuous attributes (no experimentation reported in the current paper) one could fit appropriate continuous distributions and apply kernels defined in (Jebara et al., 2004).

3.3. General Form

A weighted decomposition kernel (WDK) is characterized by the following decomposition structure:

$$\mathcal{R} = \langle \vec{X}, R, (\delta, \kappa_1, \ldots, \kappa_D)\rangle$$

where $\vec{X} = (S, Z_1, \ldots, Z_D)$, $R(s, z_1, \ldots, z_D, x)$ is true iff $s \in S$ is a subgraph of x called the *selector* and $\vec{z} = (z_1, \ldots, z_D) \in Z_1 \times \cdots \times Z_D$ is a tuple of subgraphs of x called the *contexts* of occurrence of s in x (precise definitions of s and $\vec{z}$ are domain-dependent as shown in Section 3.4 and 3.5). In order to ensure an efficient computation of the kernel, some restrictions have to be placed on the sizes of the above entities. First we assume that $|R^{-1}(x)| = O(|V(x)| + |E(x)|)$, i.e. the number of ways a graph can be decomposed grows at most linearly with its size. Second, we assume that selectors have constant size with respect to x, i.e. $R(s,\vec{z},x) \Rightarrow |V(s)| + |E(s)| = O(1)$. The definition is completed by the kernels on parts: δ is an exact matching kernel on $S \times S$ and κ_d is a graph probability distribution kernel on $Z_d \times Z_d$. This setting results in the following general form of the kernel:

$$K(x,x') = \sum_{\substack{(s,\vec{z}) \in R^{-1}(x) \\ (s',\vec{z}') \in R^{-1}(x')}} \delta(s,s') \sum_{d=1}^{D} \kappa_d(z_d, z_d') \tag{6}$$

where the direct sum between kernels over parts κ_d can be replaced by the tensor product. Compared to kernels that simply count the number of substructures, the above function weights different matches between selectors according to contextual information. The kernel can be afterwards normalized. In the following subsections we specialize this general form to practical cases of interest.

3.4. A WDK for Biological Sequences

Biological sequences are finite length strings on a finite alphabet Σ (for example Σ consists of the 20 amino acid letters in the case of proteins) and therefore $X = \Sigma^*$. Given a string $x \in \Sigma^*$, two integers $e \geq 0$ and $e \leq t \leq |x| - e$, let $x(t,e)$ denote the substring of x spanning string positions from $t - e$ to $t + e$. The simplest version of WDK is obtained by choosing $D = 1$ and a relation R depending on two integers $r \geq 0$ (the selector radius) and $l \geq r$ (the context radius) defined as $R = \{(s,z,x) : x \in \Sigma^*, s = x(t,r), z = x(t,l), l \leq t \leq |x| - l\}$. The kernel is then defined as

$$K(x,x') = \sum_{t=l}^{|x|-l} \sum_{\tau=l}^{|x'|-l} \delta(x(t,r), x'(\tau,r))\kappa(x(t,l), x'(\tau,l)).$$

Intuitively, when applied to protein sequences, this kernel computes the number of common $(2r+1)$-mers weighting matching pairs by the similarity between the amino acid composition of their environments — measured, for example, by one of the probability distribution kernels as defined in Eq. (4) or Eq. (5). Of course if $\kappa(\cdot,\cdot) \equiv 1$, then this WDK reduces to the spectrum kernel. Note that although the above equation seems to imply a complexity of $O(|x||x'|)$, more efficient implementations are possible (see Section 4).

3.5. A WDK for Molecules

A molecule is naturally represented by an undirected graph x where vertices are atoms and edges are covalent bonds. Vertices are annotated with attributes such as atom type, atom charge, membership to specific functional groups (i.e. whether the atom is part of a carbonyl, metil, alcohol or other group in the molecule) and edges are annotated with attributes such as bond type. Given a vertex v and an integer $l \geq 0$, we denote by $x(v,l)$ the subgraph of x induced by the set of vertices which are reachable from v by

a path of length at most l and by the set of all edges that have at least one end in the vertex set of $x(v,l)$. Also we write $x(v)$ as a shorthand for $x(v,0)$.

The first WDK we propose is obtained by choosing $D = 1$ and a relation R that depends on an integer $l \geq 0$ (context radius) defined as $R = \{(s,z,x) : x \in X, s = x(v), z = x(v,l), v \in V(x)\}$. The kernel is defined as

$$K(x,x') = \sum_{\substack{v \in V(x) \\ v' \in V(x')}} \delta(x(v), x'(v')) \cdot \kappa(x(v,l), x'(v',l)). \quad (7)$$

Note that selectors consist of single vertices, allowing us to compute δ in constant time. As discussed in Section 4, other options that still preserve efficiency may be available. In the second WDK we set $D = 2$ and use two types of contexts, $z_1(v,l) = x(v,l)$ and its graph complement denoted by $z_2(v,l)$. Probability kernels over contexts can be combined under direct sum obtaining $\kappa((\vec{z},v),(\vec{z}',v')) = \kappa_1(z_1(v,l), z_1'(v',l)) + \kappa_2(z_2(v,l), z_2'(v',l))$. Eq. (6) finally becomes

$$K(x,x') = \sum_{\substack{v \in V(x) \\ v' \in V(x')}} \delta(x(v), x'(v')) \cdot \kappa((\vec{z},v),(\vec{z}',v')). \quad (8)$$

4. Algorithms and Complexity

The computational efficiency of a decomposition kernel depends largely on the cost of constructing and matching substructures. In particular, exact matching of substructures might lead to intractability when dealing with general graphs (Gärtner et al., 2003). This problem is avoided in WDK. Selectors require exact matching but consist of small substructures that can be reasonably constructed and matched in $O(1)$. Examples of acceptable selectors include: short substrings for sequences, tuples formed by vertices and the ordered list of their children for trees (e.g. production rules in the case of parse trees) and single vertices for non-ordered graphs. Contexts may be large but in this case efficiency is achieved because attribute frequencies (histograms) and not subgraphs are matched. Efficient procedures can be devised for calculating the kernel under the assumption that graphs are labeled by categorical attributes. Before kernel calculation, each instance is pre-processed to construct a lexicographically sorted index that associates context histograms to selectors. The cost of this step is $O(m \log m) + T_c$ for each instance, where $m = |R^{-1}(x)|$ and T_c is the time for calculating all context histograms. The outer summation over selectors in Eq. (6) is then computed by scanning the two ordered indices of x and x'. This strategy leads to a complexity reduction of the kernel computation between two instances of the same size ranging from $O(m^2)$ up to $O(m)$, depending on indexing sparseness[1]. We now briefly discuss algorithmic ideas for computing label histograms efficiently when contexts are subgraphs formed by all vertices at bounded distance from the selector (this is the case for example in the kernels for biological sequences and for molecules proposed in Sections 3.4 and 3.5). In the case of sequences, context histograms can be updated in $O(1)$ moving along the sequence e.g. from left to right. Therefore, the time for constructing all context histograms is $T_c = O(m + l)$, where l is the size of each context and m in this case the length of the sequence. When data is organized as rooted trees, it is possible to use a vector of histograms associated with each node to store information on statistics at increasing distances. An algorithm can compute the histogram of the sub-tree dominated by a node in a recursive fashion, partially exploiting the histograms of its children. Such an algorithm can clearly construct all context histograms with complexity $O(m)$, with a linear increase in space complexity. Context histograms for DAGs can efficiently be computed following the same procedure as in the tree case (once DAGs have been topologically sorted). When a vertex has among its descendants two vertices that have a common neighbor q, care has to be taken to count the contribution of q only once. In addition to a vector of histograms, we need to associate a hash table for each vertex in order to efficiently access those descendant vertices with multiple incident edges and subtract their multiple contribution. The final complexity is bounded by $O(V + E)$. For the general case of undirected cyclic graphs directly, computing the histogram visiting in breadth-first the neighborhood of each vertex achieves a complexity bounded by $O(V^2 + VE)$.

5. Experimental Results

To demonstrate the effectiveness and versatility of our approach, we report experimental results on two protein classification tasks and two chemical compound classification tasks. In the experimentation, classification was performed using the Support Vector Machine (SVM) algorithm.

5.1. Protein Subcellular Localization

The protein subcellular localization task consists of predicting the cell compartment in which the mature protein will reside. An accurate localization prediction is considered a useful step towards understanding

[1] The best case is when each bucket contains a single context histogram, the worst when a single bucket contains all of them.

Table 1. Leave one out performance on the SubLoc data set (http://www.bioinfo.tsinghua.edu.cn/SubLoc/). The spectrum kernel is based on 3-mers and $C = 10$. For the WDK, contexts width was 15 residues and $C = 10$.

		Cytoplasmic				Extra–Cellular				Mitochondrial				Nuclear			
Method	Acc	Pre	Rec	gAv	MCC	Pre	Rec	gAv	MCC	Pre	Rec	gAv	MCC	Pre	Rec	gAv	MCC
SubLoc	79.4	72.6	76.6	.74	.64	81.2	79.7	.80	.77	70.8	57.3	.63	.58	85.2	87.4	.86	.74
Spectrum$_3$	84.9	80.4	83.3	.81	.74	90.6	85.5	.88	.86	75.8	61.4	.68	.63	88.3	92.6	.90	.82
WDK	87.9	82.6	87.9	.85	.79	96.9	87.7	.92	.91	89.7	62.3	.74	.71	88.7	95.5	.92	.85

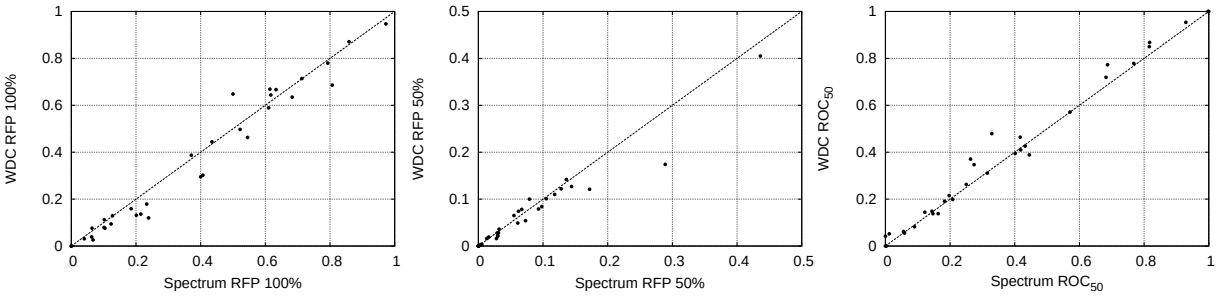
Table 2. Test set performance on the SwissProt data set defined by Nair and Rost (2003) (http://cubic.bioc.columbia.edu/results/2003/localization/). The spectrum kernel is based on 3-mers and $C = 5$. For the WDK, contexts width was 15 residues and $C = 5$.

		Cytoplasmic				Extra–Cellular				Mitochondrial				Nuclear			
Method	Acc	Pre	Rec	gAv	MCC	Pre	Rec	gAv	MCC	Pre	Rec	gAv	MCC	Pre	Rec	gAv	MCC
LOCNet	64.2	54.0	56.0	.54	-	76.0	86.0	.81	-	45.0	53.0	.49	-	71.0	73.0	.72	-
Spectrum$_3$	74.1	69.7	68.8	.69	.57	78.7	80.6	.79	.72	65.3	53.3	.59	.54	76.5	80.8	.78	.66
WDK	78.0	71.4	72.9	.72	.60	85.7	87.1	.86	.81	78.9	50.0	.62	.59	77.8	85.3	.81	.70

protein function, since proteins belonging to the same compartment could cooperate towards a common function. We report comparative results on two data sets previously studied in the literature. The first data set was prepared by Hua and Sun (2001) and consists of 2,427 eukaryotic sequences. The second data set was prepared by Nair and Rost (2003) and consists of 1,461 (train) and 512 (test) SwissProt proteins. In both cases proteins are grouped in four classes: cytoplasmic, extra-cellular, mitochondrial and nuclear. We cast the multiclass problem into four one-vs-all binary classification problems. We compare WDK against SubLoc (Hua & Sun, 2001) (an SVM predictor based on an amino acid composition kernel) and against LOCNet (Nair & Rost, 2003) (a more sophisticated connectionist approach that employs predicted secondary structure and solvent accessibility as additional inputs). In addition we compared results with our implementation of the spectrum kernel (Leslie et al., 2002). Performance was measured for each class in terms of precision, recall, geometric average, Matthew correlation coefficient (between targets and predictions) and overall 4-class accuracy. Kernel parameters have been optimized using cross-validation, obtaining context radius $l = 7$ for the WDK, 3-mers for both the spectrum kernel and WDK selector, and regularization parameter $C = 10$ for both kernels. The WDK is able to exploit larger amino acid subsequences in the form of context, while using large selectors (or large k-mers for the spectrum kernel) leads to worse generalization due to sparseness. In Table 1, we report leave one out classification results obtained with our implementation of SubLoc, the spectrum kernel and the WDK. As specified in Hua and Sun (2001), the SubLoc predictor was trained using an RBF kernel with $\gamma = 16$ and $C = 500$. In Table 2 we compare the performance of spectrum kernel, WDK and LOCNet (Nair & Rost, 2003) on the test set. The spectrum kernel consistently outperforms SubLoc (showing that features other than the overall amino acid composition are useful for this prediction task) and is also highly competitive against LOCNet. In all cases, WDK results show that further improvement over the spectrum kernel is possible by exploiting context information around 3-mers. We conjecture that WDK is capturing some short sorting signals in the protein sequence.

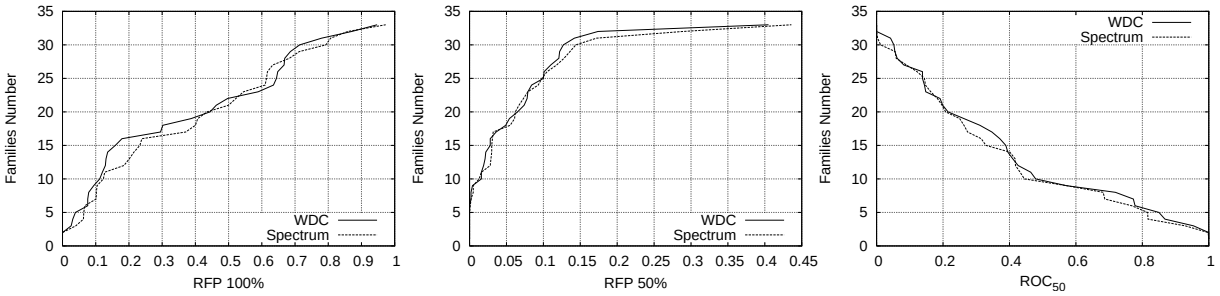
5.2. Protein Family Classification

Remote protein homology detection is the task to find homologies between proteins that are in the same superfamily but not necessarily in the same family. The superfamily classification is useful to annotate new unknown proteins with structural and functional features from similar known proteins. We tested WDK on the sample of the Structural Classification of Proteins (SCOP) dataset used in the experimental setup by Jaakkola et al. (2000) and Leslie et al. (2002). We followed Jaakkola et al. (2000) simulating the remote homology task by holding out all members of a target family from a given superfamily. The holding out family sequences were positive test examples, while remaining families in the superfamily were positive training examples; negative training and test examples were chosen from outside the target family fold. Classification performance was evaluated measuring $\text{RFP}_{100\%}$, $\text{RFP}_{50\%}$ (rates of false positives at recall 1 and 0.5 respectively) and ROC_{50} (the area under the ROC span-



(a) (b) (c)

Figure 1. Remote Protein Homologies: family by family comparison of the WDK and the spectrum kernel. The coordinates of each point are the RFP at 100% coverage (a), at 50% coverage (b) and the ROC_{50} scores (c) for one SCOP family, obtained using the WDK and spectrum kernel. Note that the better performance is under the diagonal in (a) and (b), while is over in (c).



(a) (b) (c)

Figure 2. Remote Protein Homologies: comparison of the WDK and spectrum kernel. The graphs plot the total number of families for which a given method is within a RFP at 100% coverage threshold (a), at 50% coverage threshold (b) and exceeds an ROC_{50} score threshold (c).

ning the first 50 false positive). Parameters for WDK have been chosen to be: selector radius $r = 1$, context radius $l = 7$ residues and regularization parameter $C = 1$. The spectrum kernel is based on 3-mers and $C = 1$. Sequences with more than one unknown residual are discarded. Results for all 33 SCOP families obtained with our implementation of the spectrum kernel and the WDK are reported in Figures 1–2. We note that WDK performs favorably against the spectrum kernel on relatively hard family to recognize, i.e. families with low ROC_{50} or high RFP, but also on the easy ones, while it is comparable on families laying in an intermediate region. The relative error reduction obtained by the WDK when measuring the ROC_{50}, $RFP_{100\%}$, $RFP_{50\%}$ averaged over all 33 families is 3.2%, 3.4% and 0.8% respectively.

5.3. HIV Dataset

The HIV dataset contains 42,687 compounds evaluated for evidence of anti–HIV activity by DTP AIDS Antiviral Screen of the National Cancer Institute, 422 of which are confirmed active (CA), 1081 are moderately active (CM) and 41184 are confirmed inactive. It is available at http://dtp.nci.nih.gov/docs/aids/aids_data.html. WDK was tested on three binary classification problems, following the experimental setup of Deshpande et al. (2003): CA vs. CM , CA vs. CI, and CA+CM vs. CI. Classification performance was measured by the mean and standard deviation of the ROC area on a five folds cross validation setup in which the original class distribution was preserved in each fold. We used as vertex attribute the atom type and as edge attribute both bond type and triplets encoding the bond type and the two bonded atoms types. The regularization parameter $C = 100$ was optimized on a four folds cross validation set. The misclassification cost β for positive examples was increased to match the ratio n^-/n^+ between positive and negative examples. In Table 3 we report results for the CA vs. CM task at increasing values for the context radius. In addition to the standard WDK with a single context Eq. (7) we tested the WDK using subgraph complements Eq. (8). We note that performance improves with larger context radii and, consistently, is better when using subgraph complement. In the subsequent experiments, reported

Table 3. HIV dataset: CA vs. CM task. Effect of varying the context radius l and the absence $D = 1$ or presence $D = 2$ of graph complement.

D	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$
1	80.1±0.8	81.8±0.9	81.6±0.8	82.0±1.5	81.6±1.5
2	82.2±1.2	83.5±0.7	83.8±1.7	84.2±1.2	83.8±0.8

Table 4. HIV dataset. FSG: best results (optimized support and $\beta = n^-/n^+$) reported by Deshpande et al. (2003) using topological features; CPK: results reported by Horváth et al. (2004) using $\beta = n^-/n^+$; CPK* same, using an optimized β^*; WDK: $\beta = n^-/n^+$.

	CA vs. CM	CA+CM vs. CI	CA vs. CI
FSG	79.2	79.4	90.8
CPK	82.7±1.3	80.1±1.7	92.8±1.0
CPK*	82.9±1.2	80.1±1.7	93.4±1.1
WDK	84.2±1.2	81.7±1.8	94.0±1.5

in Table 4, we used graph complement and context radius $l = 4$. For comparison, Table 4 shows the best results reported on this data set by Deshpande et al. (2003) and by Horváth et al. (2004) (also measured by five-fold cross validation but on a different split). Note however that we compare our results on a fair base, i.e. against Deshpande et al. (2003) without additional geometrical features and against Horváth et al. (2004) without the composition with gaussian kernel.

5.4. Predictive Toxicology Challenge (PTC)

The PTC is a classification problem over the carcinogenicity properties of chemical compounds on mice and rats. To test WDK classification performance on this task we used the U.S. National Institute for Environment Health Studies dataset available at http://www.predictive-toxicology.org/ptc. The dataset (Helma et al., 2001) lists the bioassays of 417 chemical compounds for four type of rodents: male mice (MM), female mice (FM), male rats (MR) and female rats (FR), which give rise to four distinct and independent classification problems. Each compound is classified as clear evidence (CE), positive (P), some evidence (SE), negative (N), no evidence (NE), equivocal (E), equivocal evidence (EE) and inadequate study (IS). We followed the experimental design of Deshpande et al. (2003), ignoring E, EE, IS classes, grouping CE, P, SE in the positive class and N, NE in the negative one. In addition to the atom type attribute we enriched vertex information with a discrete attribute on atom charge (taking values in $\{-1, 0, 1\}$) and functional group membership, that is, whether the atom is part of one among 28 different group types such as carbonyl, ester, anhydrid, ketone, alcohol, etc. Edge attributes comprise both bond type and triplets encoding the bond type and the two bonded atoms types. The regularization parameter C was optimized on a four folds cross validation for each one of the four classification problems. Both in the optimization and training phase the misclassification cost for positive examples was increased to match the positive to negative example ratio. Classification performance was evaluated measuring the mean and standard deviation of the area under the ROC curve on a five folds cross validation preserving the original class distribution on each fold. We performed two experiments to identify the effect of different parameters over classification performance. In the first experiment we let context radius l vary and we contrast single context WDK against WDK with additional complementary context. Results reported in Table 5 show that the presence of the graph complement ($D = 2$) increases performance for MR and FR, while a larger context radius is useful for FM and FR. In the second experiment we compared four WDKs obtanied combining tensor product, direct sum, Bhattacharyya and histogram intersection kernels. Results indicate that the direct sum version generally outperforms the tensor product kernel. Our conjecture is that simpler problems benefit from the smaller feature space generated by direct sum version, while more complex problems can be best solved in a larger feature space induced by the tensor product kernel. We finally compared our result to the FSG (Deshpande et al., 2003) and the extended marginalized graph kernel (EMGK) (Mahé et al., 2004). The best results obtained by the three methods are compa-

Table 5. PTC: effect of varying the context radius l and the absence $D = 1$ or presence $D = 2$ of graph complement.

$D = 1$	$l = 1$	$l = 2$	$l = 3$
MM	70.5±4.3	70.0±5.5	69.9±6.3
FM	67.4±6.9	68.1±9.7	69.1±5.8
MR	63.8±6.4	67.8±7.2	68.4±6.3
FR	61.5±8.1	61.3±7.4	60.4±5.7
$D = 2$	$l = 1$	$l = 2$	$l = 3$
MM	68.1±6.2	68.1±6.2	68.1±5.8
FM	65.4±7.6	65.1±8.8	66.9±8.1
MR	69.7±7.2	69.1±7.3	67.7±6.3
FR	62.2±4.8	62.2±5.6	64.9±5.1

rable. The WDK defined by Eq. (7) and (8) has the advantage of not requiring a computationally expensive graph pre-processing phase compared to FSG. In addition, it exhibits a stable behavior with respect to model parameters, as opposed to EMGK.

6. Conclusions

We introduced the weighted decomposition kernels, a computationally efficient and general family of kernels on decomposable objects. We report experimental evidence showing that the proposed kernel performs remarkably well with respect to more complex and computationally demanding methods on a number of different bioinformatics problems ranging from protein sequences to molecule graphs classification. Future working directions include the extension of the proposed approach to non trivial selectors (using for example frequent subgraph mining algorithms) and to probability distributions over subgraphs occurrences.

Acknowledgments

This research is supported by EU Grant APrIL II (project n° 508861), EU NoE BIOPATTERN (project n° 508803), and MIUR Grant 2003091149_002.

References

Ben-David, S., Eiron, N., & Simon, H. U. (2002). Limitations of learning via embeddings in euclidean half spaces. *J. of Mach. Learning Research*, *3*, 441–461.

Collins, M., & Duffy, N. (2001). Convolution kernels for natural language. *NIPS 14* (pp. 625–632).

Cortes, C., Haffner, P., & Mohri, M. (2004). Rational kernels: Theory and algorithms. *J. of Machine Learning Research*, *5*, 1035–1062.

Cumby, C. M., & Roth, D. (2003). On kernel methods for relational learning. *Proceedings of ICML'03*.

Deshpande, M., Kuramochi, M., & Karypis, G. (2003). Frequent Sub–Structure–Based Approaches for Classifying Chemical Compounds. *Proceedings of ICDM 2003* (pp. 35–42).

Gärtner, T. (2003). A survey of kernels for structured data. *SIGKDD Explor. Newsl.*, *5*, 49–58.

Gärtner, T., Flach, P., & Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. *Proc. of COLT/Kernel '03* (pp. 129–143).

Haussler, D. (1999). *Convolution kernels on discrete structures* (Technical Report UCSC-CRL-99-10). University of California, Santa Cruz.

Helma, C., King, R. D., Kramer, S., & Srinivasan, A. (2001). The Predictive Toxicology Challenge 2000–2001. *Bioinformatics*, *17*, 107–108.

Horváth, T., Gärtner, T., & Wrobel, S. (2004). Cyclic pattern kernels for predictive graph mining. *Proceedings of KDD'04* (pp. 158–167). ACM Press.

Hua, S., & Sun, Z. (2001). Support Vector Machine for Protein Subcellular Localization Prediction. *Bioinformatics*, *17*, 721–728.

Jaakkola, T., Diekhans, M., & Haussler, D. (2000). A Discriminative Framework for Detecting Remote Protein Homologies. *J. of Comp. Biology*, *7*, 95–114.

Jebara, T., Kondor, R., & Howard, A. (2004). Probability product kernels. *J. Mach. Learn. Res.*, *5*, 819–844.

Kashima, H., Tsuda, K., & Inokuchi, A. (2003). Marginalized kernels between labeled graphs. *Proceedings of ICML'03* (pp. 321–328).

Kramer, S., Raedt, L. D., & Helma, C. (2001). Molecular feature mining in HIV data. *Proc. of KDD-01* (pp. 136–143). ACM Press.

Leslie, C. S., Eskin, E., & Noble, W. S. (2002). The spectrum kernel: A string kernel for SVM protein classification. *Pacific Symposium on Biocomputing* (pp. 566–575).

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *J. Mach. Learn. Res.*, *2*, 419–444.

Mahé, P., Ueda, N., Akutsu, T., Perret, J.-L., & Vert, J.-P. (2004). Extensions of marginalized graph kernels. *Proceedings of ICML'04* (pp. 552–559).

Nair, R., & Rost, B. (2003). Better Prediction of Sub–Cellular Localization by Combining Evolutionary and Structural Information. *Proteins: Structure, Function, and Genetics*, *53*, 917–930.

Odone, F., Barla, A., & Verri, A. (2005). Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing*, *14*, 169–180.

Schölkopf, B., Weston, J., Eskin, E., Leslie, C. S., & Noble, W. S. (2002). A kernel approach for learning from almost orthogonal patterns. *Proc. of ECML'02* (pp. 511–528).

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.