A Structural Cluster Kernel for Learning on Graphs

Madeleine Seeland
Technische Universität
München
Institut für Informatik I12
Boltzmannstr. 3
85748 Garching b. München,
Germany
madeleine.seeland@in.tum.de

Andreas Karwath
Johannes
Gutenberg-Universität Mainz
Institut für Informatik
Staudingerweg 9
55128 Mainz, Germany
karwath@informatik.uni-mainz.de

Stefan Kramer
Johannes
Gutenberg-Universität Mainz
Institut für Informatik
Staudingerweg 9
55128 Mainz, Germany
kramer@informatik.uni-mainz.de

ABSTRACT

In recent years, graph kernels have received considerable interest within the machine learning and data mining community. Here, we introduce a novel approach enabling kernel methods to utilize additional information hidden in the structural neighborhood of the graphs under consideration. Our novel structural cluster kernel (SCK) incorporates similarities induced by a structural clustering algorithm to improve state-of-the-art graph kernels. The approach taken is based on the idea that graph similarity can not only be described by the similarity between the graphs themselves, but also by the similarity they possess with respect to their structural neighborhood. We applied our novel kernel in a supervised and a semi-supervised setting to regression and classification problems on a number of real-world datasets of molecular graphs. Our results show that the structural cluster similarity information can indeed leverage the prediction performance of the base kernel, particularly when the dataset is structurally sparse and consequently structurally diverse. By additionally taking into account a large number of unlabeled instances the performance of the structural cluster kernel can further be improved.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms, Experimentation, Performance

Keywords

Cluster Kernels, Graph Kernels, Structural Graph Clustering, Cheminformatics, QSAR

KDD'12, August 12–16, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1462-6 /12/08 ...$15.00.

1. INTRODUCTION

The topic of graph similarity and in particular kernel approaches have attracted considerable interest in recent years [13, 18, 20, 25]. To determine the similarity of two graphs, most approaches decompose the graphs in different ways: either into a potentially very large set of smaller subgraphs or related graph features, or into one or more larger common subgraphs (connected or disconnected). In this paper, we investigate the question whether the structural neighborhood of two graphs can also contribute to similarity searches and consequently to improve prediction performance. In our setting, the structural neighborhood of a graph is determined by a recently proposed structural graph clustering approach called PSCG [21, 22].

In the work presented here, we propose a novel kernel called *structural cluster kernel* (SCK) which, in addition to existing kernel approaches, measures the similarity between two graphs, by their assignment to structural clusters found with PSCG. Our approach first employs the structural clustering algorithm to determine small, structurally homogeneous regions in the input space, and then uses the pairwise similarities between these regions to define a similarity measure for graphs. The approach taken here is to extend two state-of-the-art graph kernels using this structural distance measure: the weighted decomposition kernel (WDK) [18] and the neighborhood subgraph pairwise distance kernel (NSPDK) [10].

To study the effectiveness of the SCK, we measured the prediction performance in the regression and classification setting, by employing several real-world datasets of molecular graphs within our experiments. To show the advantage of combining graph similarity and structural cluster similarity, we compare our approach with the base kernels using graph similarity alone. Furthermore, we compare the SCK to a different approach also employing structural clustering during model construction. We also investigate the performance of the SCK approach in the semi-supervised setting, where the base kernel is deformed by a cluster kernel encoding similarities between both labeled and unlabeled examples.

This paper is organized as follows: After discussing related work in Section 2, we introduce our proposed structural cluster kernel in Section 3. Section 4 presents and discusses our experimental results, before we conclude in Section 5.

2. RELATED WORK

The idea of combining kernels to improve prediction performance has attracted attention recently. Several types of cluster kernels, relying on different clustering algorithms, have been proposed by Chapelle *et al.* [6]. The authors present a general framework for constructing cluster kernels which implements the cluster assumption, i.e., the induced distance depends on whether the points are in the same cluster or not. Weston *et al.* [26] investigated the use of cluster kernels for protein classification by developing two simple and scalable methods for modifying a base kernel. The neighborhood kernel uses averaging over a neighborhood of sequences defined by a local sequence similarity measure, and the bagged kernel uses bagged clustering of the full sequence dataset to modify the base kernel. In both the semi-supervised and transductive settings, these techniques greatly improve the classification performance when used with mismatch string kernels. In work by Bodo and Csato [3] a kernel construction algorithm for supervised and semi-supervised learning was proposed, which constitutes a general framework for semi-supervised kernel construction. The technique clusters the labeled and unlabeled data by an agglomerative clustering technique, and uses the linkage distances induced by the clustering hierarchy to construct the kernel. Bodo and Csato [4] proposed two cluster kernel methods for semi-supervised learning which can be used for different types of datasets: one using hierarchical clustering, and another kernel for reweighting an arbitrary base kernel taking into account the cluster structure of the data.

Similar to Weston *et al.* [26] and Bodo and Csato [4], the cluster kernel proposed in this paper leverages information of a clustering algorithm to modify a base kernel. However, our approach differs from existing work in several respects. First, our structural cluster kernel can be applied in the domain of graphs. Second, it builds on two state-of-the-art graph kernels and a recently proposed structural graph clustering algorithm to determine small, structurally homogeneous neighborhoods of the input space. The pairwise similarities between these neighborhoods are used to define a similarity measure for graphs which in turn is used to improve a base kernel. Third, the proposed cluster kernel can be used for both graph classification and regression, whereas the above mentioned cluster kernels were only tested on classification tasks.

3. METHOD

3.1 Structural Graph Clustering

Parallel Structural Clustering of Graphs (PSCG) [21, 22] investigates the problem of finding groups of graphs sharing some structural similarity. Graphs with similar structures are expected to be in the same cluster provided that their common substructures match to a satisfactory extent. The common substructure of a cluster can be considered as a scaffold present in all cluster members. Only connected substructures are considered as common substructures. The sizes of these common substructures are used as a measure of similarity between the graphs. A graph is assigned to a cluster provided that there exists at least one common substructure whose size is equal or greater than a user-defined threshold. In this way, a graph can simultaneously belong to multiple clusters (overlapping clustering) if the size of at least one common substructure with these clusters is equal or greater than the defined threshold. If a graph does not meet the threshold to share a common substructure with any cluster, the graph is not included in any cluster (non-exhaustive clustering). For one graph after the other, it is decided whether it belongs to an existing cluster or whether a new cluster should be created. Formally, the problem of structural clustering is defined as follows: given a set of graphs, $X = \{x_1, \ldots, x_n\}$, we assign each graph x_i to a cluster C_j, such that the similarity between graphs is based on their structural similarity, including multiple, or overlapping cluster assignments. In graph clustering, one objective considered is to maximize the average number of graphs contained in a cluster, such that for each cluster C_j there exists at least one common substructure that makes up a specific proportion, θ, of the size of each cluster member. Considering the state of a cluster $C = \{x_1, \ldots, x_m\}$[1] at any point in time, the criterion can formally be defined as:

$$\exists\, s \in cs(\{x_1, \ldots, x_m\}) \forall x_i \in C : |s| \geq \theta |x_i| \tag{1}$$

where cs determines all common subgraphs of a set of graphs, and $\theta \in [0, 1]$ is a user-defined similarity coefficient. If a new graph x_{m+1} is to be tested for inclusion in cluster C, we can thus infer a minimum size threshold for the substructures shared by this graph and the graphs in the cluster:

$$minSize = \theta\ max(|x_{max}|, |x_{m+1}|), \tag{2}$$

where $\theta \in [0, 1]$ and x_{max} is the largest graph in the cluster. To obtain meaningful and interpretable results, the minimum size of a graph considered for cluster membership is further constrained by a minimum graph size threshold. It excludes graphs that are too small from clustering. Thus, the identification of the cluster scaffold will not be impeded by the presence of a few graph structures whose shared common substructure is much smaller than the one the majority of the cluster members share. For computing common substructures in graphs, we modified the graph mining algorithm gSpan [16, 27] that mines frequent substructures in a database of graphs satisfying a given minimum frequency constraint. For details of these modifications, the structural clustering algorithm and its performance, we refer the interested reader to the original publications [21, 22].

3.2 Weighted Decomposition Kernel

The basic idea of the Weighted Decomposition Kernel (WDK) [18] is to focus 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.

More formally, a weighted decomposition kernel is characterized by a decomposition $R(s, z, x)$ where s is a subgraph of x called the selector and z is a subgraph of x called the context of occurrence of s in x. This setting results in the following general form of the kernel:

$$K(x, x') = \sum_{(s,x) \in R^{-1}(x)} \sum_{(s',x') \in R^{-1}(x')} \delta(s, s') \kappa(z, z'), \tag{3}$$

where κ is a kernel on contexts and δ is the exact matching kernel applied to selectors.

In this paper, selectors are single atoms and the matching kernel $\delta(s, s')$ is defined by the coincidence between the type of s and s'. The context kernel k is based on a soft match between substructures, defined by the distributions of label

[1] In slight abuse of notation, we use the same indices as above.

contents after discarding topology. In this paper, we use the following attributes labeling vertices and edges: atom type, atom charge and bond type. Contexts are formed as follows: Given a vertex $v \in V$ and an integer $r \geq 0$, called the context radius. We denote by $x(v, r)$ the substructure of x composed of the vertices within distance r from vertex v, and the set of all edges that have at least one end in the vertex set of $x(v, r)$. More formally, we define the decomposition relation depending on r as $R_r = \{(s, z, x) : x \in X, s = \{v\}, z = x(v, r), v \in V\}$, where s is the selector and z is the context for vertex v. In our case, the matching kernel $\delta(v, v')$ returns 1 if the two vertices v and v' have the same label.

3.3 Neighborhood Subgraph Pairwise Distance Kernel

The Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) [10] is based on exact matching between pairs of small subgraphs. Formally, let $R_{r,d}(A^v, B^u; G)$ denote the relation between two rooted graphs A^v,B^u and a graph G to be true iff both A^v and B^u are in $\{N_r^v : v \in V(G)\}$, where A^v (B^u) is isomorphic to some N_r and $D(u, v) = d$. In words: the relation $R_{r,d}$ selects all pairs of neighborhood graphs of radius r whose roots are at distance d in a given graph G.

We define $\kappa_{r,d}$ over $G \times G$ as the decomposition kernel on the relation $R_{r,d}$, i.e.,

$$\kappa_{r,d}(G, G') = \sum_{\substack{A^v, B^u \in R_{r,d}^{-1}(G) \\ A'^{v'}, B'^{u'} \in R_{r,d}^{-1}(G')}} \delta(A^v, A'^{v'})\delta(B^u, B'^{u'}) \quad (4)$$

where δ is the exact matching kernel. In words: $\kappa_{r,d}$ counts the number of identical pairs of neighboring graphs of radius r at distance d between two graphs.

The NSPDK is finally defined as:

$$K(G, G') = \sum_r \sum_d \kappa_{r,d}(G, G'). \quad (5)$$

In this work we impose an upper bound on the radius and the distance: $K_{r^*,d^*}(G, G') = \sum_{r=0}^{r^*} \sum_{d=0}^{d^*} \kappa_{r,d}(G, G')$ that is, NSPDK is limited to the sum of the $\kappa_{r,d}$ kernels for all increasing values of the radius (distance) parameter up to a maximum given value r^* (d^*).

3.4 Structural Cluster Kernel

In this section, we introduce a novel kernel, called structural cluster kernel, that leverages information of a clustering algorithm to improve a base kernel representation. The main idea is to change the similarity metric of a base kernel so that the relative similarity between two points is higher if the points are in the same cluster. Our kernel uses a combination of two similarity measures: (1) a base kernel that computes structural similarity between pairs of graphs and (2) a cluster based similarity measure that describes how close examples are to each other in terms of the similarities between the clusters they belong to. The similarity between two clusters is computed by taking the average of the similarities between the cluster instances. In our application to molecule regression and classification, we use the WDK and NSPDK (see Section 3.2 and 3.3) as the base kernel. For the cluster based kernel, we use the structural clustering algorithm introduced in section 3.1 that clusters a dataset of graphs based on structural similarity. The cluster similarity information is used to improve pointwise similarities, based on which we construct the final kernel.

In the following, we describe the steps which are necessary to build the structural cluster kernel. Let $D_{Trg} = \{(x_1, y_1), \ldots, (x_t, y_t)\}$ denote the training data, where $x_i \in X$ represent the data points and y_i their labels, respectively. Further, let $D_{Tst} = \{x_{t+1}, \ldots, x_n\}$ denote the set of test points. We first cluster the training set with the structural clustering procedure PSCG presented in Section 3.1. The resulting clusters are used to build a kernel representing the pairwise similarities between all clusters. In this kernel representation, each of the pairwise sets of the structural clusters is seen as a single data point, and a higher level kernel is designed so as to compare the two clusters. The similarity between two clusters is computed by taking the average of the sum of the pairwise similarities between all graph instances in both clusters. The kernel $K(C_i, C_j)$ is defined as

$$K(C_i, C_j) = \begin{cases} \frac{1}{|C_i||C_j|} \sum\limits_{x_k \in C_i} \sum\limits_{x_l \in C_j} K_b(x_k, x_l) & \text{if } i \neq j \\ 1 & \text{if } i = j, \end{cases} \quad (6)$$

where $K_b(x_k, x_l)$ represents the base kernel and $C_i, C_j \in \{C_1, \ldots, C_p\}$. As mentioned earlier, we use the WDK and NSPDK as base kernel to compute the pairwise similarities between graphs. In the next step, we build a kernel representation $K_{Cl}(x_i, x_j)$ based on the averaged pairwise similarities between the clusters x_i and x_j belong to. $K_{Cl}(x_i, x_j)$ is defined as

$$K_{Cl}(x_i, x_j) = \begin{cases} \frac{1}{|n_{x_i}||n_{x_j}|} \sum\limits_{C_k : x_i \in C_k} \sum\limits_{C_l : x_j \in C_l} K(C_k, C_l) & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \quad (7)$$

where n_{x_i} denotes the number of clusters containing x_i, n_{x_j} denotes the number of clusters containing x_j and $C_k, C_l \in \{C_1, \ldots, C_p\}$. Thus, we map the points to a feature space where the pointwise similarities are equal to the cluster similarities in the input space. The points belonging to the same cluster will result in matrix entries close to one, whereas for the points from different clusters, the entries will be close to zero. Figure 1 illustrates the cluster kernel concept.

The cluster similarity weights $K_{Cl}(x_i, x_j)$ are combined with the values of the base kernel $K_b(x_i, x_j)$, thus forming the final kernel matrix. To sum up, the new structural cluster kernel is

$$K_{SC}(x_i, x_j) = K_b(x_i, x_j) \times K_{Cl}(x_i, x_j) \quad (8)$$

We are faced with two problems in the construction of the above structural cluster kernel: (i) the base kernel matrix has to be positive semi-definite and (ii) the structural cluster kernel must be positive semi-definite. The first requirement is obvious, since we use the WDK and NSPDK as base kernels, which are known to be valid kernels. In the following, we provide a proof sketch to show that the structural cluster kernel is a valid kernel.

Proof Sketch:.

K_{Cl} is a valid kernel, since each kernel value $K_{Cl}(x_i, x_j)$ contains the average sum of pairwise similarities between all

Algorithm 1 Structural Cluster Kernel

```
Given: training points $D_{Trg} = \{(x_1, y_1), \ldots, (x_t, y_t)\}$ and test points $D_{Tst} = \{x_{t+1}, \ldots, x_n\}$, $x_i \in \mathbb{R}^n$, $i = 1, \ldots, n$
a) Cluster training points using PSCG [22, 21]
c) Build cluster matrix on the training set
   $K_{Cl}(x_i, x_j) = \frac{1}{|n_{x_i}||n_{x_j}|} \sum_{C_k : x_i \in C_k} \sum_{C_l : x_j \in C_l} K(C_k, C_l)$, $i, j \in \{1, \ldots, t\}$
d) Build the SCK on the training dataset by taking the product between the base kernel $K_b$ and the cluster kernel $K_{Cl}$
   $K_{SC}(x_i, x_j) = K_{Cl}(x_i, x_j) \times K_b(x_i, x_j)$, $i, j \in \{1, \ldots, t\}$
e) Compute cluster assignments for all test points
f) Compute $K_{SC}(x_j, x_i)$ between each test point $x_j$ and all training points $x_i$, $i = 1, \ldots, t$.
```

clusters, which in turn encompass the average sum of all training instances $x_i \in \{x_1, \ldots, x_t\}$.

For each pair of clusters, we define one kernel that returns the average similarity between the two clusters for the first instance in cluster one and the second in cluster two. For all other instances, it returns zero. As the sum of two valid kernels is again a valid kernel, the resulting function is a valid kernel as well. Applying the kernel to two instances, we only consider the clusters to which the two instances are assigned, consequently most of the summands are equal to zero:

$$
\begin{aligned}
K_{Cl}(x_i, x_j) =& \frac{1}{|n_{x_i}||n_{x_j}|} \sum_{C_k : x_i \in C_k} \sum_{C_l : x_j \in C_l} K(C_k, C_l) \\
=& \frac{1}{|n_{x_i}||n_{x_j}|} \sum_{C_k : x_i \in C_k} \sum_{C_l : x_j \in C_l} K(C_k, C_l) \\
&+ \frac{1}{|m_{x_i}||m_{x_j}|} \underbrace{\sum_{C_k : x_i \notin C_k} \sum_{C_l : x_j \notin C_l} K(C_k, C_l)}_{0} \\
&+ \frac{1}{|n_{x_i}||m_{x_j}|} \underbrace{\sum_{C_k : x_i \in C_k} \sum_{C_l : x_j \notin C_l} K(C_k, C_l)}_{0} \\
&+ \frac{1}{|m_{x_i}||n_{x_j}|} \underbrace{\sum_{C_k : x_i \notin C_k} \sum_{C_l : x_j \in C_l} K(C_k, C_l)}_{0},
\end{aligned}
\quad (9)
$$

where n_{x_i} denotes the number of clusters containing x_i and m_{x_i} denotes the number of clusters not containing x_i.

In kernel methods, for predicting the label of a new test point we need to perform kernel function calculations only between the test points and the training points. For computing the kernel entries, we first need to assign each test point to one or more clusters using the structural clustering procedure to compute $K_{Cl}(x_i, x_t)$. Based on this cluster assignment the similarity $K_{Cl}(x_i, x_t)$ between the test point x_t and all training points x_i is computed by averaging the pairwise similarities between all clusters x_t and x_i are assigned to (Equation 7). The kernel matrix K_{SC} is extended by taking the inner product between $K_{Cl}(x_i, x_t)$ and $K_b(x_i, x_t)$ between each test point x_t and all training points x_i, $i = 1, \ldots, t$.

The steps needed for the calculation of the structural cluster kernel are shown in Algorithm 1.

3.5 Semi-Supervised Setting

In semi-supervised learning, one tries to improve a classifier trained on labeled data by exploiting a relatively large set of unlabeled data. If unlabeled data is added to the relatively small labeled dataset, we expect that the new similarity, obtained via structural clustering and the use of unlabeled data, induces a better representational space for classification and regression than using only the labeled data. Therefore, we extend on the kernel construction in Section 3.4 by involving a large number of unlabeled data. The structural cluster kernel is constructed as follows: We first cluster both the labeled and unlabeled training data with the structural clustering procedure to determine small, structurally homogeneous neighborhoods of the input space. The resulting clusters are then used to build a kernel representing the pairwise similarities between all clusters. As in the supervised setting, the cluster similarity information is used to improve pointwise similarities between the labeled data samples, based on which the final structural cluster kernel is constructed.

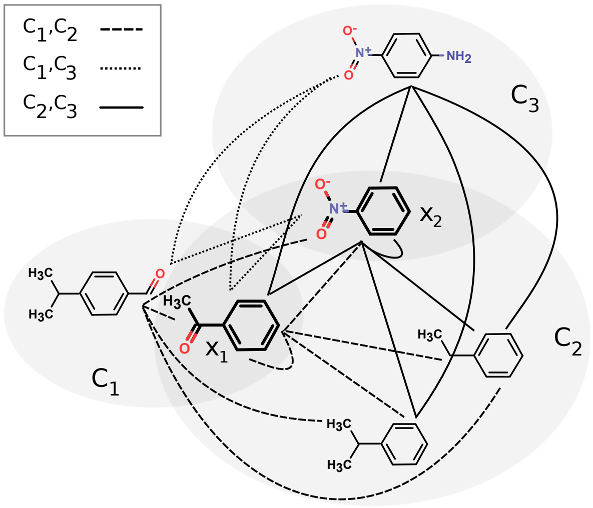


Figure 1: Illustration of the cluster kernel concept. The cluster based similarity $K_{Cl}(x_1, x_2)$ between the highlighted structures x_1 and x_2 is computed based on the averaged pairwise similarities between the clusters they belong to. x_1 belongs to C_1 and C_2, x_2 to C_2 and C_3. Thus, we need to compute the pairwise similarities between the cluster instances of cluster C_1C_2, C_1C_3, C_2C_2 (which equals 1) and C_2C_3.

4. EXPERIMENTAL RESULTS

In this section, we first study the performance of our proposed structural cluster kernel in a supervised setting. Next, we investigate the cluster kernel approach in a semi-supervised setting to test if the prediction performance can be improved by including a large amount of unlabeled data. For all experiments, we employed the chemical domain as our application area by using real datasets of molecular graphs. In Table 1 an overview of the datasets is provided.

Table 1: Overview of the datasets used for assessing our structural cluster kernel. n denotes the number of molecular graphs in the respective dataset.

Dataset	n	class.(SAR)/ regr.(QSAR)	Reference
4QSAR COX2	282	regression	4QSAR database [24]
4QSAR DHFR	362	regression	4QSAR database [24]
CPD MOUSE	442	regression	ACD DSSTox databases [14]
CPD RAT	580	regression	ACD DSSTox databases [14]
ISS MOUSE	316	regression	Benigni/Vari Carcinogenicity [2]
ISS RAT	375	regression	Benigni/Vari Carcinogenicity [2]
Suth COX2	414	regression	Sutherland dataset [23]
Suth DHFR	672	regression	Sutherland dataset [23]
Suth ER TOX	410	regression	Sutherland dataset [23]
FDAMDD	1216	regression	ACD DSSTox databases [17]
Biodeg	328	regression	Biodegradability dataset [11]
Tox09	1213	regression	Environmental Toxicity Prediction Challenge 2009 [1]
ER_LIT	381	regression	Sutherland dataset [23]
CYP INH 2C9	700	classification	Yap and Chen [28]
CYP SUB 2C9	700	classification	Yap and Chen [28]
Fontaine	435	classification	Fontaine *et al.* [12]
NCI AIDS	1000	classification	DTP AIDS Antiviral Screen [9]
CPDB MUT	684	classification	Mutagenicity dataset [15]

4.1 Supervised Setting

In this section, we empirically compare the performance of our structural cluster kernel approach against five methods.

1. **WDK**: The Weighted Decomposition Kernel is used to build a classification or regression model. Section 3.2 provides a detailed description of the WDK.

2. **NSPDK**: The Neighborhood Subgraph Pairwise Distance Kernel (see Section 3.3) is used to build a classification or regression model.

3. **LoMoGraph**: LoMoGraph [5] combines clustering and classification or regression for making predictions on graph structured data. The approach consists of two steps: First, the structural clustering procedure PSCG [22] is applied to find groups of graphs in a structural space that share a common structural scaffold with a minimum size. The sizes of these common subgraphs are used as a measure of similarity between the graphs. A graph is assigned to a cluster provided that there exists at least one common subgraph, whose size is equal or greater than a user-defined threshold θ. Second, one local model is learned per structural cluster using a feature-vector representation of the graphs where the features encode standard chemical descriptors in our setting of molecular graphs. In the prediction step, the query graph is assigned to one or more clusters using PSCG. Based on this assignment, the prediction is made. Since the structural clustering procedure is overlapping and non-exhaustive, a graph can fall into no cluster, one cluster or multiple clusters. If it falls into no cluster, a global model is applied for prediction. If the query graph falls into a single cluster, the local model based on this cluster is used for prediction, and if it is assigned to multiple clusters, weighted local models are used dependent on cluster membership. The weight for a cluster is linearly dependent on its size. Thus, larger weights are assigned to larger clusters, assuming that the more graphs a cluster has, the more reliable the corresponding model is.

4. **LoMoGraph WDK**: The method combines LoMoGraph with WDK. More precisely, one local model is learned per structural cluster based on the WDK.

5. **LoMoGraph NSPDK**: The method combines LoMoGraph with NSPDK, i.e., one local model is learned per structural cluster based on the NSPDK.

We investigated our structural cluster kernel approach using both NSPDK and WDK as base kernel. For SCK with NSPDK and SCK with WDK, we investigated not only the approach with the diagonal elements in the kernel matrices $K(C_i, C_j)$ (Equation 6) and $K_{Cl}(x_i, x_j)$ (Equation 7) set to one, but also a second approach, where the diagonal elements are computed in the same way as the non-diagonal elements. We refer to these four approaches as SCK NSPDK (d=1), SCK NSPDK (d≠1), SCK WDK (d=1) and SCK WDK (d≠1).

In the experiments, regression and classification were performed using the Support Vector Machine (SVM) algorithm. Several user parameters were optimized by internal cross-validation. For SCK WDK, SCK NSPDK, WDK, NSPDK, LoMoGraph WDK, and LoMoGraph NSPDK, the trade-off between training error and margin, C, was selected from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. Further, we optimized the radius r for WDK in $\{1, 2, 3, 4\}$. The parameter combination resulting in the lowest mean absolute error (the highest accuracy) was then used for building the final model. All other SVM parameters were left at their default values. For NSPDK, the maximum radius r^* was set to 2, and the maximum distance d^* to 5. For the SCK approaches, we set the similarity coefficient θ of PSCG to 0.5. For FDAMDD and NCI AIDS we made an exception and set θ to 0.3 to take into account the size and structural heterogeneity of the datasets. As for LoMoGraph, the parameters that were used for clustering were defined based on a set of criteria: the similarity coefficient of PSCG was chosen such that the local models consist of minimally 5% and maximally 20% of the training data. The rationale behind this choice is that a too small value of θ results in large, heterogeneous clusters whereas a too big value of θ produces very few, small clusters or no clusters at all. In both cases the predictivity of LoMoGraph would be negatively affected. For the experiments on SCK, we used the same values for θ for all datasets. Another parameter called minimum cluster size controls how many graphs a cluster must have at least so that a local model can be learned. This parameter was chosen greater than or equal to 20 as a lower bound for the number of graphs that are needed to train meaningful models.

Performance estimates are obtained using 100 times hold-out validation with a training set fraction of 66%. This means that 2/3 of the data are used for training a model while the remaining 1/3 is reserved for testing. To quantify predictive accuracy, we choose the relative mean error (regression) and classification accuracy (classification), which are standard measures in regression and classification settings. The Wilcoxon signed-rank test and the corrected re-

Table 2: Mean absolute errors with standard deviations of SCK NSPDK, NSPDK, LoMoGraph NSPDK and LoMoGraph on the regression datasets. Statistically significant results are reported using both the Wilcoxon signed-rank test and the corrected resampled t-test (separated by a '|').

Dataset	*SCK NSPDK (d≠1)*	SCK NSPDK (d=1)	NSPDK	LoMoGraph NSPDK	LoMoGraph
4QSAR COX2	*0.607 ± 0.055*	0.625 ± 0.055 ●|	0.601 ± 0.054 ○|	0.628 ± 0.055 ●|	0.868 ± 0.135 ●|●
4QSAR DHFR	*0.551 ± 0.035*	0.572 ± 0.037 ●|●	0.541 ± 0.033 ○|	0.562 ± 0.035 ●|	0.955 ± 0.102 ●|●
CPD MOUSE	*0.751 ± 0.041*	0.760 ± 0.041 ●|	0.764 ± 0.040 ●|	0.775 ± 0.039 ●|●	1.276 ± 0.154 ●|●
CPD RAT	*0.868 ± 0.045*	0.870 ± 0.044 ●|	0.887 ± 0.043 ●|●	0.877 ± 0.042 |	1.529 ± 0.116 ●|●
ISS MOUSE	*0.738 ± 0.046*	0.740 ± 0.046 |	0.753 ± 0.045 ●|●	0.761 ± 0.048 ●|●	1.197 ± 0.111 ●|●
ISS RAT	*0.860 ± 0.050*	0.863 ± 0.047 |	0.900 ± 0.046 ●|●	0.873 ± 0.051 ●|	1.371 ± 0.109 ●|●
Suth COX2	*0.559 ± 0.039*	0.586 ± 0.041 ●|●	0.552 ± 0.038 ○|○	0.573 ± 0.040 ●|	0.905 ± 0.137 ●|●
Suth DHFR	*0.504 ± 0.026*	0.519 ± 0.030 ●|●	0.493 ± 0.026 ○|○	0.498 ± 0.027 ○|	0.941 ± 0.066 ●|●
Suth ER TOX	*0.828 ± 0.046*	0.842 ± 0.045 ●|	0.820 ± 0.042 |	0.847 ± 0.052 ●|	1.216 ± 0.141 ●|●
FDAMDD	*0.629 ± 0.029*	0.647 ± 0.023 ●|●	0.612 ± 0.024 ○|○	0.621 ± 0.026 ○|	0.951 ± 0.051 ●|●
Biodeg	*0.844 ± 0.051*	0.875 ± 0.055 ●|●	0.867 ± 0.050 ●|●	0.874 ± 0.053 ●|●	-
Tox09	*0.345 ± 0.015*	0.386 ± 0.016 ●|●	0.342 ± 0.015 |	0.367 ± 0.016 ●|●	-
ER_LIT	*0.492 ± 0.039*	0.499 ± 0.035 ●|	0.495 ± 0.039 |	0.495 ± 0.041 |	-

●,○ statistically significant improvement, or degradation of SCK NSPDK (d≠1) with respect to the other methods

Table 3: Mean absolute errors with standard deviations of SCK WDK, WDK, LoMoGraph WDK and LoMoGraph on the regression datasets. Statistically significant results are reported using both the Wilcoxon signed-rank test and the corrected resampled t-test (separated by a '|').

Dataset	*SCK WDK (d=1)*	SCK WDK (d≠1)	WDK	LoMoGraph WDK	LoMoGraph
4QSAR COX2	*0.673 ± 0.088*	0.691 ± 0.056 ●|●	0.683 ± 0.055 |	0.676 ± 0.053 |	0.868 ± 0.135 ●|●
4QSAR DHFR	*0.669 ± 0.054*	0.805 ± 0.132 ●|●	0.733 ± 0.051 ●|●	0.702 ± 0.047 ●|	0.955 ± 0.102 ●|●
CPD MOUSE	*0.827 ± 0.052*	0.874 ± 0.085 ●|	0.888 ± 0.063 ●|●	0.880 ± 0.052 ●|●	1.276 ± 0.154 ●|●
CPD RAT	*1.002 ± 0.048*	1.070 ± 0.103 ●|	1.129 ± 0.154 ●|●	1.043 ± 0.048 ●|●	1.529 ± 0.116 ●|●
ISS MOUSE	*0.798 ± 0.049*	0.826 ± 0.053 ●|●	0.850 ± 0.056 ●|●	0.859 ± 0.062 ●|●	1.197 ± 0.111 ●|●
ISS RAT	*0.977 ± 0.064*	1.018 ± 0.074 ●|●	1.031 ± 0.061 ●|●	1.022 ± 0.062 ●|●	1.371 ± 0.109 ●|●
Suth COX2	*0.612 ± 0.042*	0.620 ± 0.047 ●|	0.603 ± 0.041 ○|	0.610 ± 0.044 |	0.905 ± 0.137 ●|●
Suth DHFR	*0.625 ± 0.036*	0.639 ± 0.086 ●|	0.633 ± 0.032 ●|	0.610 ± 0.031 ○|	0.941 ± 0.066 ●|●
Suth ER TOX	*0.993 ± 0.064*	1.301 ± 0.439 ●|●	1.175 ± 0.071 ●|●	1.130 ± 0.076 ●|●	1.216 ± 0.141 ●|●
FDAMDD	*0.727 ± 0.029*	0.809 ± 0.077 ●|●	0.833 ± 0.329 ●|●	0.727 ± 0.029 |	0.951 ± 0.051 ●|●
Biodeg	*1.011 ± 0.072*	1.051 ± 0.119 ●|	1.110 ± 0.071 ●|●	1.086 ± 0.080 ●|●	-
Tox09	*0.440 ± 0.021*	0.643 ± 0.168 ●|●	0.464 ± 0.017 ●|●	0.425 ± 0.017 ○|	-
ER_LIT	*0.594 ± 0.040*	0.590 ± 0.058 ○|	0.609 ± 0.040 ●|	0.591 ± 0.043 ●|	-

●,○ statistically significant improvement, or degradation of SCK WDK (d=1) with respect to the other methods

Table 4: Classification accuracies with standard deviations of SCK NSPDK, NSPDK, LoMoGraph NSPDK and LoMoGraph on the classification datasets. Statistically significant results are reported using both the Wilcoxon signed-rank test and the corrected resampled t-test (separated by a '|').

Dataset	*SCK NSPDK (d≠1)*	SCK NSPDK (d=1)	NSPDK	LoMoGraph NSPDK	LoMoGraph
CYP INH	*76.33 ± 2.35*	75.62 ± 2.52 ●|	75.55 ± 2.50 ●|	75.88 ± 2.56 ●|	74.08 ± 2.55 ●|●
CYP SUB	*76.84 ± 2.28*	75.11 ± 2.84 ●|●	75.95 ± 2.17 ●|	76.21 ± 1.99 ●|	71.37 ± 2.46 ●|●
Fontaine	*95.56 ± 1.57*	95.37 ± 1.57 ●|	95.71 ± 1.66 |	95.62 ± 1.60 |	92.08 ± 2.01 ●|●
NCI AIDS	*90.13 ± 1.58*	89.86 ± 1.69 ●|	90.58 ± 1.35 ●|	89.02 ± 1.62 ●|●	84.93 ± 1.61 ●|●
CPDB MUT	*76.71 ± 2.10*	75.15 ± 2.20 ●|●	77.25 ± 2.12 ○|	73.85 ± 2.50 ●|●	-

●,○ statistically significant improvement, or degradation of SCK NSPDK (d≠1) with respect to the other methods

Table 5: Classification accuracies with standard deviations of SCK WDK, WDK, LoMoGraph WDK and LoMoGraph on the classification datasets. Statistically significant results are reported using both the Wilcoxon signed-rank test and the corrected resampled t-test (separated by a '|').

Dataset	*SCK WDK (d=1)*	SCK WDK (d≠1)	WDK	LoMoGraph WDK	LoMoGraph
CYP INH	*74.05 ± 3.02*	70.78 ± 3.85 ●|●	75.05 ± 2.51 ○|	75.42 ± 2.42 ○|	74.08 ± 2.55 |
CYP SUB	*72.07 ± 3.84*	70.63 ± 4.63 ●|	75.77 ± 2.38 ○|○	75.74 ± 2.46 ○|○	71.37 ± 2.46 ●|
Fontaine	*94.01 ± 1.86*	94.48 ± 1.83 ○|	94.41 ± 1.54 ○|	93.03 ± 5.14 ●|	92.08 ± 2.01 ●|●
NCI AIDS	*84.28 ± 2.04*	83.67 ± 2.20 ●|	79.97 ± 8.32 ●|●	82.27 ± 1.82 ●|●	84.93 ± 1.61 ●|
CPDB MUT	*72.80 ± 2.48*	71.51 ± 2.88 ●|	73.29 ± 2.57 |	70.85 ± 2.64 ●|	-

●,○ statistically significant improvement, or degradation of SCK WDK (d=1) with respect to the other methods

sampled t-test [19] are applied to test for significant differences at a significance level of 5%.

Tables 2, 3, 4 and 5 show the detailed experimental results in terms of relative mean absolute error (regression) and accuracy (classification) for the various methods on all datasets. The results for LoMoGraph are taken from the original publication [5]. Since not all datasets were used in this paper, the table contains missing values. In the same tables the second column shows the performance of the respective SCK method as baseline to compare against. For better illustration we highlight the reference method in italic. We indicate whether the respective SCK method is significantly better or worse than the comparison methods at $p < 0.05$ using both the Wilcoxon signed-ranked test and the corrected resampled t-test. In the following, we discuss the results based on the more conservative corrected resampled t-test. Overall, our experimental results show that the structural cluster kernel with NSPDK as base kernel performs always better than all comparison methods using WDK as base kernel. This demonstrates that NSPDK is a much more powerful base kernel compared to WDK. Moreover, we observe that the choice of setting the diagonal entries in the kernel matrix has a different effect on both SCK methods. Whereas setting the diagonal entries of the kernel matrix unequal to one leads to better predictive performance for SCK NSPDK, setting the diagonal entries equal to one results in better predictive performance for SCK WDK. In the following, we analyze the performance of the SCK approaches on the different datasets. On the COX2 datasets, we observe no performance improvement of SCK NSPDK and SCK WDK over the respective base kernel. The datasets contain extremely similar molecules, often differing in only one atom. Hence, the base kernel cannot be improved by the similarities induced by the structural clustering procedure. For the CPD, ISS and Biodeg datasets, a comparison between the mean absolute errors shows a clear performance advantage of SCK using both WDK and NSPDK as base kernel. Primarily, we explain this positive effect as a result of the structurally heterogeneity of the datasets consisting of many small molecules (∼ up to 10 atoms). Hence, the NSPDK alone is not suited to determine similarity between graphs. As a consequence, for these datasets the pairwise similarities between the small, structurally homogeneous neighborhoods can contribute to similarity and consequently to predictive performance. On FDAMDD, the proposed structural cluster kernel with NSPDK as base kernel yields performance degradation compared to NSPDK. This shows that taking into account the similarities induced by PSCG has an adverse effect on the predictive performance. Although for this dataset a significant performance gain of SCK over the base kernel can be achieved by using WDK as base kernel, SCK WDK still has a higher mean absolute error compared to NSPDK. This demonstrates that NSPDK is much more powerful compared to WDK. For NCI AIDS and both CYP datasets the results on classification are clearly in favor of SCK NSPDK. On these datasets the structural cluster kernel with NSPDK improves over all other compared methods. However, for the corrected resampled t-test only four of the nine wins are statistically significant. Using WDK as base kernel, SCK can only achieve strong performance improvements on NCI AIDS. On the remaining classification datasets taking into account similarities induced by PSCG has either no significant effect or an adverse effect on predictive accuracy compared to the baseline methods (except for LoMoGraph on the Fontaine dataset). In summary, our structural cluster kernel approach is comparative to other methods, yet shows a strong performance increase on structurally more sparse datasets, i.e., chemically and structurally more diverse datasets. On these datasets the base kernel alone is not suited to determine similarities between graphs due to the high structural heterogeneity within the dataset. Hence, the structural neighborhood of two graphs can substantially contribute to graph similarity and therefore to predictive performance of the constructed models.

4.2 Semi-Supervised Setting

In this section, we investigate whether incorporating unlabeled data in the clustering process can positively contribute to predictive performance. Since semi-supervised methods potentially give the greatest benefit when a large amount of unlabeled data is used, we tested our structural cluster kernel approach in large-scale experiments, enriching the training data by a large number of molecules from the vast chemical space. For this, we employed the ChemDB database, which contains nearly 5 M commercially available small molecules [7, 8], as a source of unlabeled data, randomly sampling 100,000 structures from it. Since in the supervised setting, SCK NSPDK (d≠1) performs always better than or equal to all methods using WDK as base kernel as well as to SCK NSPDK (d=1), LoMoGraph NSPDK and LoMoGraph, we only compared SCK in the semi-supervised setting against SCK NSPDK (d≠1) and NSPDK. As in the supervised setting, the SVM complexity constant, C, was selected from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. Further, we used the same parameter setting for the NSPDK and the similarity coefficient θ.

The experimental results are shown in Tables 6 and 7 and in the bar charts in Figure 2. For completeness, the bar charts also depict the results for LoMoGraph NSPDK and LoMoGraph. The following discussion is based on the corrected resampled t-test. The results show that in the semi-supervised setting, SCK NSPDK achieves a strong performance gain on all datasets over the supervised approach: 10 of the 18 wins are statistically significant. For regression, the best results can be achieved on the toxicity datasets consisting of structurally more heterogeneous graphs. The results indicate that incorporating a large set of unlabeled data into the structural clustering process has a definite positive effect on the predictive performance. As opposed to the supervised setting, SCK NSPDK can improve over the base kernel on the FDAMDD dataset. This dataset is the largest one, comprising structurally heterogeneous molecules. Hence, exploiting a large set of unlabeled data in the clustering step can contribute to graph similarity. The strongest performance gains with respect to NSPDK can be achieved on the classification datasets. Whereas in the supervised setting SCK NSPDK was not able to gain significantly with respect to the base kernel on the classification datasets, the semi-supervised approach shows significant improvements over NSPDK in three out of five cases.

Table 6: Mean absolute errors with standard deviations of SCK NSPDK in both the semi-supervised and supervised setting and NSPDK on the classification datasets. Statistically significant results are reported using the Wilcoxon signed-rank test and the corrected resampled t-test (separated by a '|').

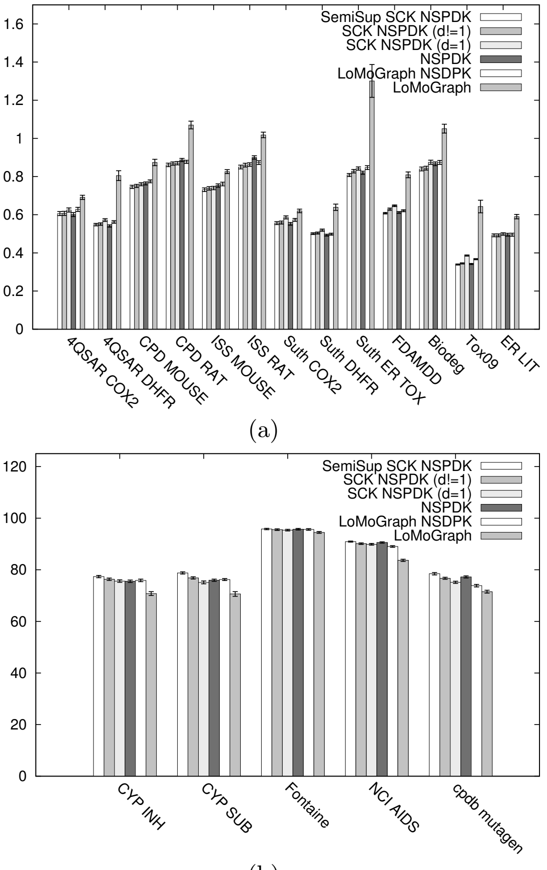
Dataset	*SCK NSPDK* *Semi-Sup*	SCK NSPDK (d≠1)	NSPDK
4QSAR COX2	*0.606 ± 0.055*	0.607 ± 0.055 |	0.601 ± 0.054 ○|
4QSAR DHFR	*0.548 ± 0.033*	0.551 ± 0.035 |	0.541 ± 0.033 ○|
CPD MOUSE	*0.746 ± 0.043*	0.751 ± 0.041 ●|	0.764 ± 0.040 ●|●
CPD RAT	*0.861 ± 0.046*	0.868 ± 0.045 ●|●	0.887 ± 0.043 ●|●
ISS MOUSE	*0.731 ± 0.049*	0.738 ± 0.046 ●|●	0.753 ± 0.045 ●|●
ISS RAT	*0.850 ± 0.050*	0.860 ± 0.050 ●|●	0.900 ± 0.046 ●|●
Suth COX2	*0.556 ± 0.040*	0.559 ± 0.039 ●|	0.552 ± 0.038 |
Suth DHFR	*0.501 ± 0.025*	0.504 ± 0.026 |	0.493 ± 0.026 ○|○
Suth ER TOX	*0.808 ± 0.041*	0.828 ± 0.046 ●|●	0.820 ± 0.042 ●|●
FDAMDD	*0.608 ± 0.020*	0.629 ± 0.029 ●|●	0.612 ± 0.024 ●|
Biodeg	*0.840 ± 0.051*	0.844 ± 0.051 ●|●	0.867 ± 0.050 ●|●
Tox09	*0.339 ± 0.014*	0.345 ± 0.015 ●|●	0.342 ± 0.015 ●|
ER_LIT	*0.492 ± 0.039*	0.492 ± 0.039 |	0.495 ± 0.039 ●|

●,○ statistically significant improvement, or degradation of SCK NSPDK Semi-Sup with respect to the other methods

Table 7: Classification accuracies with standard deviations of SCK NSPDK in both the semi-supervised and supervised setting and NSPDK on the classification datasets. Statistically significant results are reported using the Wilcoxon signed-rank test and the corrected resampled t-test (separated by a '|').

Dataset	*SCK NSPDK* *Semi-Sup*	SCK NSPDK (d≠1)	NSPDK
CYP INH	*77.37 ± 2.26*	76.33 ± 2.35 ●|●	75.55 ± 2.50 ●|●
CYP SUB	*78.78 ± 2.16*	76.84 ± 2.28 ●|●	75.95 ± 2.17 ●|●
Fontaine	*95.80 ± 1.42*	95.56 ± 1.57 |	95.71 ± 1.66 |
NCI AIDS	*90.92 ± 1.00*	90.13 ± 1.58 ●|	90.58 ± 1.35 ●|
CPDB MUT	*78.47 ± 2.40*	76.71 ± 2.10 ●|●	77.25 ± 2.12 ●|●

●,○ statistically significant improvement, or degradation of SCK NSPDK Semi-Sup with respect to the other methods



(a)

(b)

Figure 2: a) Mean absolute errors with 95% confidence intervals and b) classification accuracies with 95% confidence intervals on the different comparison methods for the data sets in Table 1.

5. CONCLUSION

In the work presented here, we proposed a novel graph kernel approach that incorporates similarity information based on structural graph clustering [21, 22] to improve state-of-the-art graph kernels. The proposed kernel is based on the idea that graph similarity can not only be determined by the similarity of the graphs alone, i.e., their structure, but also by the similarity of the graphs' structural neighborhood. We investigated the performance of the structural cluster kernels for regression and classification by using several real-world datasets of molecular graphs. In our experiments we performed a comparison with the weighted decomposition kernel, the neighborhood subgraph pairwise distance kernel, and a learning method combining clustering with classification or regression for the prediction task. The results demonstrate that the proposed kernel approach yields an increase in performance on a number of datasets, in particular on structurally more diverse datasets. We also investigated the performance of our approach in the semi-supervised setting, by enriching relatively small labeled datasets by a large set of unlabeled data instances from the vast chemical space. The results show that within the semi-supervised setting our approach achieves gains in performance when compared to the supervised version as well as to the pure base kernel, in particular for classification. We believe that the approach presented is general as such, and can also be employed in conjunction with a variety of different kernels and clustering approaches and is therefore not restricted to graph mining alone.

6. REFERENCES

[1] Environmental Toxicity Prediction Challenge, http://www.cadaster.eu/node/65.

[2] R. Benigni, C. Bossa, and M. Vari. Chemical Carcinogens: Structures and Experimental Data, http://www.iss.it/binary/ampp/cont/ISSCANv2aEn.1134647480.pdf.

[3] Z. Bodo. Hierarchical cluster kernels for supervised and semi-supervised learning. In *4th International Conference on Intelligent Computer Communication and Processing*, ICCP 2008, pages 9–16, 2008.

[4] Z. Bodo and L. Csato. Hierarchical and reweighting cluster kernels for semi-supervised learning. *Int. J. Comput. Comm. Contr.*, 5(4):469–476, 2010.

[5] F. Buchwald, T. Girschick, M. Seeland, and S. Kramer. Using local models to improve (Q)SAR predictivity. *Mol. Inf.*, 30(2-3):205–218, 2011.

[6] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 15 of *NIPS 2002*, pages 585–592. MIT Press, 2003.

[7] J. Chen, S. J. Swamidass, Y. Dou, and P. Baldi. ChemDB: a public database of small molecules and related chemoinformatics resources. *Bioinf.*, 21:4133–4139, 2005.

[8] J. H. Chen, E. Linstead, S. J. Swamidass, D. Wang, and P. Baldi. ChemDB update – full text search and virtual chemical space. *Bioinf.*, 23:2348–2351, 2007.

[9] J. M. Collins. The DTP AIDS Antiviral Screen Program 1999, http://dtp.nci.nih.gov/docs/aids/aidsdata.html.

[10] F. Costa and K. De Grave. Fast neighborhood subgraph pairwise distance kernel. In *Proc. 26th International Conference on Machine Learning*, ICML'10, pages 255–262, 2010.

[11] S. Dzeroski, H. Blockeel, B. Kompare, S. Kramer, B. Pfahringer, and W. Van Laer. Experiments in predicting biodegradability. In *Applied Artificial Intelligence*, pages 80–91. Springer, 1999.

[12] F. Fontaine, M. Pastor, I. Zamora, and F. Sanz. Anchor-GRIND: Filling the gap between standard 3D QSAR and the GRid-INdependent descriptors. *J. Med. Chem.*, 48(7):2687–2694, 2005.

[13] T. Gärtner. *Kernels for Structured Data.* PhD thesis, Universität Bonn, 2005.

[14] L. Gold, T. Slone, B. Ames, N. Manley, G. Garfinkel, and L. Rohrbach. *Handbook of Carcinogenic Potency and Genotoxicity Databases*, chapter Carcinogenic Potency Database, pages 1–605. 1997.

[15] C. Helma, T. Cramer, S. Kramer, and L. De Raedt. Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds. *J. Chem. Inf. Comput. Sci.*, 44(4):1402–1411, 2004.

[16] K. Jahn and S. Kramer. Optimizing gSpan for molecular datasets. In *Proc. 3rd International Workshop on Mining Graphs, Trees and Sequences*, 2005.

[17] E. Matthews, N. Kruhlak, R. Benz, and J. Contrera. Assessment of the health effects of chemicals in humans: I. QSAR estimation of the maximum recommended therapeutic dose (MRTD) and no effect level (NOEL) of organic chemicals based on clinical trial data. *Curr. Drug. Disc. Tech.*, 1(1):61–76, 2004.

[18] S. Menchetti, F. Costa, and P. Frasconi. Weighted decomposition kernels. In *Proc. 22nd International Conference on Machine Learning*, ICML 2005, pages 585–592. ACM Press, 2005.

[19] C. Nadeau and Y. Bengio. Inference for the generalization error. *Mach. Learn.*, 52:239–281, 2003.

[20] U. Rückert, T. Girschick, F. Buchwald, and S. Kramer. Adapted transfer of distance measures for quantitative structure-activity relationships. In *Proc. 13th International Conference on Discovery Science*, DS 2010, pages 341–355. Springer-Verlag, 2010.

[21] M. Seeland, S. A. Berger, A. Stamatakis, and S. Kramer. Parallel structural graph clustering. In *Proc. 2011 European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML/PKDD 2011, pages 256–272, 2011.

[22] M. Seeland, T. Girschick, F. Buchwald, and S. Kramer. Online structural graph clustering using frequent subgraph mining. In *Proc. 2010 European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML/PKDD 2010, pages 213–228, 2010.

[23] J. J. Sutherland, L. A. O'Brien, and D. F. Weaver. Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships. *J. Chem. Inf. Comput. Sci.*, 43(6):1906–1915, 2003.

[24] J. J. Sutherland, L. A. O'Brien, and D. F. Weaver. A comparison of methods for modeling quantitative structure-activity relationships. *J. Med. Chem.*, 47(22):5541–5554, 2004.

[25] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *J. Mach. Learn. Res.*, 11:1201–1242, 2010.

[26] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. *Bioinf.*, 21(15):3241–3247, 2005.

[27] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *Proc. 2002 IEEE International Conference on Data Mining*, pages 721–724, 2002.

[28] C. W. Yap and Y. Z. Chen. Prediction of cytochrome P450 3A4, 2D6, and 2C9 inhibitors and substrates by using support vector machines. *J. Chem. Inf. Model.*, 45(4):982–992, 2005.