Neurocomputing 74 (2010) 219–229

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Graph kernels for chemical compounds using topological and three-dimensional local atom pair environments

Georg Hinselmann *, Nikolas Fechner, Andreas Jahn, Matthias Eckert, Andreas Zell

Center for Bioinformatics (ZBIT), University of Tübingen, Tübingen, Germany

ARTICLE INFO

Article history:
Received 12 July 2009
Received in revised form
19 February 2010
Accepted 1 March 2010
Communicated by L. Kurgan
Available online 25 March 2010

Keywords:
Graph kernel
Support vector machine
Molecular graph
Cheminformatics
Structure–activity relationship

ABSTRACT

Approaches that can predict the biological activity or properties of a chemical compound are an important application of machine learning. In this paper, we introduce a new kernel function for measuring the similarity between chemical compounds and for learning their related properties and activities. The method is based on local atom pair environments which can be rapidly computed by using the topological all-shortest paths matrix and the geometrical distance matrix of a molecular graph as lookup tables. The local atom pair environments are stored in prefix search trees, so called tries, for an efficient comparison. The kernel can be either computed as an optimal assignment kernel or as a corresponding convolution kernel over all local atom similarities. We implemented the Tanimoto kernel, min kernel, minmax kernel and the dot product kernel as local kernels, which are computed recursively by traversing the tries.

We tested the approach on eight structure-activity and structure-property molecule benchmark data sets from the literature. The models were trained with ε- support vector regression and support vector classification. The local atom pair kernels showed to be at least competitive to state-of-the-art kernels in seven out of eight cases in a direct comparison. A comparison against literature results using similar experimental setups as in the original works confirmed these findings. The method is easy to implement and has robust default parameters.

1. Introduction

The prediction of properties and activities (i.e. the ability of a small molecule to interact with a protein) of chemical compounds by in silico models is an important task in cheminformatics. This procedure makes it possible to estimate a chemical property or activity using the data obtained in expensive in vitro experiments. A chemical compound is usually regarded either as molecular graph or as geometrical object in Computer Science.

An important aspect is the question of how to measure a similarity between two structures. Molecular information is typically categorized in 0D, 1D, 2D, 3D, and 4D. 0D refers to simple counts like the molecular weight or the number of heavy atoms. The 1D describes fragment information, like specific functional groups. The 2D approaches are based on the topology like depth-first-search (DFS) fingerprints of a molecule, whereas 3D methods use information depending on three-dimensional coordinates. The 4D methods take the conformational space of a structure into account.

A kernel machine infers a predictive model by exploiting the pairwise similarities of a set of training instances. To be a valid kernel, the similarity measure has to be a real-valued, symmetric, and positive semidefinite function. As the learning problem is only based on pairwise similarities, it is possible to work without an explicit representation of the samples (i.e. bit vectors of a fixed size).

Some kernels suitable for molecular graphs have been proposed recently. The marginalized graph kernel published in 2003 [1] computes the similarity by an expectation kernel on random walks with a length based on a probability distribution. The optimal assignment kernel [2–4] published in 2005 computes an optimal assignment between the sets of local atom environments of two molecules. A similar method is the iterative similarity optimal assignment kernel published by Rupp et al. in 2007 [5]. A flexibility extension for pairwise molecular similarity measures which considers 2D and 3D information has been applied to the optimal assignment kernel and was published by Fechner et al. [6] in 2009. Ralaivola et al. published a kernel which is closely related to DFS fingerprints in 2005 [7]. Borgwardt et al. proposed a graph kernel using all topological shortest paths [8] in 2005. Examples of 3D kernels are the Pharmacophore kernel by Mahé et al. [9], which compares geometrical p-point relationships, and kernels based on the alignment of bipods published by Mohr et al. in 2008 [10].

* Corresponding author. Tel.: +49 7071 2977174; fax: +49 7071 29 5091.
E-mail address: georg.hinselmann@uni-tuebingen.de (G. Hinselmann).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.03.008

There are various attempts to use optimal assignment techniques to match vertex neighborhoods of two bipartite graphs in different application domains. Gori et al. [11] proposed a general framework for graph matching based on Google's PageRank algorithm [12]. In this approach, the neighborhoods are encoded by random walks to enrich the information of the vertices of a graph. To approximate the edit distance between two graphs, Riesen et al. [13] proposed an algorithm based on Munkres's algorithm [14]. Here, the cost of assigning a vertex depends on a local optimal assignment of edit costs. In principle, this is the same idea as presented by Fröhlich et al. [3,4] for molecular graphs. Singh et al. [15] published a bipartite graph matching based on Google's PageRank. In this algorithm, the problem of matching similar topological neighborhoods is formulated as an eigenvalue problem. The object of the mapping is to obtain an approximated maximum common subgraph. Jain et al. [16] propose an algorithm for the maximum contact map overlap problem. The vertex neighborhoods are defined by assigning neighbors to a vertex if the spatial distance is below a predefined distance. Jain et al. [17] approximated the Schur–Hadamard inner product (a quadratic assignment problem) as a similarity between two graphs by using Hopfield networks. The resulting similarity matrices were used to train a support vector machine.

Our approach is based on the idea to assign each atom the distance information to the remaining atoms and their corresponding atom type in a molecule. Thus, each vertex (i.e. heavy atom) of a chemical compound is embedded in a matching environment. This is realized by assigning a trie encoding of the local atom pair environment to each atom. A trie is created by using the topological all-shortest paths matrix or the binned geometrical distance matrix as lookup table. The representation as a trie leads to an efficient computation of the local kernels because a large part of the feature space is discarded by using a branch and bound technique. The overall kernel is computed by an optimal assignment kernel or by a convolution kernel on the kernel similarities of the matching environments. The encoding of molecular information as atom pairs is an established approach in cheminformatics [18,19].

The benchmarks on eight QSAR (quantitative structure–activity relationship) and ADME (absorption, distribution, metabolism, and excretion) data sets show that the new kernel is at least competitive to state-of-the-art methods in seven cases. We tested the performance of the kernels by a nested 10-fold cross-validation repeated 100 times on each data set. The significance was evaluated by the corrected resampled t-test. The method was also benchmarked in a variant combining topological and geometrical information by multiplying the local kernel similarities. A comparison of the method against literature results also yields promising results. The empirical computation times show that our implementation has a practical feasible computation time between 0.83 and 3.43 ms for a kernel calculation on the benchmark sets on an AMD Opteron 280 processor.

2. Materials and methods

2.1. Convolution kernel

A kernel is a real-valued, symmetric, and positive semidefinite function $k : \chi \times \chi \to \mathbb{R}$ defined on an arbitrary space χ. The class of kernel functions is enclosed under specific operations which allow the design of new kernels out of existing ones.

Haussler [20] introduced a general framework for the design of kernels based on decomposition of objects. The R-Convolution of kernels $k_1,\ldots,k_d$ is defined as

$$(k_1 \cdot k_2 \cdot \ldots \cdot k_d) \cdot (x,x') \coloneqq \sum_{R} \prod_{d=1}^{D} k_d(x_d,x'_d) \tag{1}$$

where the kernel is summed up on all decompositions allowed by R of x into $x_1, \ldots, x_d$ and x' into $x'_1,\ldots,x'_d$.

As a special case, this kernel can be computed on one type of decomposition and one type of kernel κ:

$$k_{\mathrm{CK}}(\mathbf{x},\mathbf{x}') \coloneqq \sum_{i,j}^{|x|,|x'|} \kappa(x_i,x'_j). \tag{2}$$

For example, this is the case for the k-spectrum kernel by Leslie et al. [21]. The convolution of two kernels is also a valid kernel. Given a decomposition into two types of representations $x,\overline{x}$ with exactly the same number of parts (i.e. $|\mathbf{x}| = |\overline{\mathbf{x}}|$ and $|\mathbf{x}'| = |\overline{\mathbf{x}}'|$), the convolution kernel of a single kernel can be written as

$$k_{\mathrm{CK}}(\mathbf{x},\mathbf{x}') \coloneqq \sum_{i,j}^{|x|,|x'|} \kappa(x_i,x'_j)\kappa(\overline{x_i},\overline{x'_j}). \tag{3}$$

In this work, the decomposition of a molecule into representation x,x' and $\overline{x},\overline{x}'$ leads to exactly the same number of elements for each representation.

2.2. Optimal assignment kernels

2.2.1. Optimal assignment problem

The goal of an optimal assignment is to assign two sets of disjoint arbitrary objects $X=(x_1,x_2,\ldots,x_n)$ and $Y=(y_1,y_2,\ldots,y_m)$, where $n \le m$ w.l.o.g. and $w(x_i,y_j)$ is the cost of assigning x_i to y_j such that the overall cost is minimal:

$$\mathrm{cost} = \min_{\pi \in \prod(X)} \sum_{i=1}^{m} w(x_{\pi(i)},y_i). \tag{4}$$

Here, $\Pi(X)$ is defined as the permutation of the indices.

It is possible to compute an optimal solution of this problem using the Hungarian method [14] that has a complexity of $O(\max(n,m)^3)$. As a case in point, this problem formulation can be used to approximate NP-hard graph matching problems (e.g. the quadratic assignment problem or the maximum common subgraph problem) by decomposing the graph into subgraphs and to find a minimum cost matching.

2.2.2. Optimal assignment kernel for molecular graphs

The idea of the optimal assignment kernel (OAK) [3,4] is to compute an optimal weighted assignment on two sets of objects and to use the normalized sum of the assignment weights as kernel. In this case, the two disjoint sets of features $\mathbf{x},\mathbf{x}'$, where $\mathbf{x} \coloneqq (x_1,x_2,\ldots,x_{|x|})$ and $\mathbf{x}' \coloneqq (x'_1,x'_2,\ldots,x'_{|x'|})$ are the sets of heavy atoms or atom environments (subgraphs, subtrees that included this atom) that compose the corresponding molecular graph. The kernel function of the optimal weighted assignment is defined as follows:

$$k_{\mathrm{OA}}(\mathbf{x},\mathbf{x}') \coloneqq \begin{cases} \max\limits_{\pi \in \Pi(\mathbf{x}')} \sum\limits_{i=1}^{|\mathbf{x}|} \kappa(x_i,x'_{\pi(i)}) & \text{if } |\mathbf{x}'| > |\mathbf{x}| \\ \max\limits_{\pi \in \Pi(\mathbf{x})} \sum\limits_{j=1}^{|\mathbf{x}'|} \kappa(x_{\pi(j)},x'_j) & \text{otherwise.} \end{cases} \tag{5}$$

$\Pi(\mathbf{x})$ and $\Pi(\mathbf{x}')$ denote all possible permutations of $\mathbf{x}$ and $\mathbf{x}'$, respectively. The atom-wise similarity is determined by a local kernel κ. The OAK uses a local atom environment which encodes the molecular neighborhood up to a defined depth using nominal and numerical features. $k_{\mathrm{OA}}(\mathbf{x},\mathbf{x}')$ computes the maximum score of all possible permutations.

An optimal assignment kernel is not positive semidefinite in some cases [22,23] and is therefore a pseudo-kernel. One possibility to fix this problem is to shift the spectrum of the kernel matrix via the transformation $K \leftarrow K - \lambda_{\min} I$, where $\lambda_{\min}$