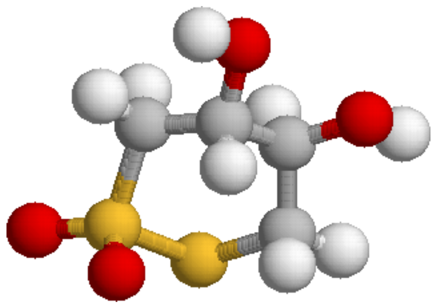
A Kernel between Sets of Vectors

Sebastian Marius Kirsch
skirsch@moebius.inka.de

Overview

1. Applications
2. Related work
3. Description of implemented approach
4. Evaluation, experiments and results
5. Conclusion

Application: Drug Screening



- determine drug activity against certain diseases (AIDS, cancer)
- drug data (molecule structure, 3d data, activity) available from National Cancer Institute
- construct kernel function on drug data for use in SVM

Related work

- graph kernels (cyclic pattern) for drug screening (Tamás Horváth and Thomas Gärtner, FHI AIS)
- set kernels between vectors using probability distributions and Bhattacharyya's affinity (Risi Kondor and Tony Jebara, Columbia Univ.)
- set kernels using kernel principal angles (Lior Wolf and Amnon Shashua, Hebrew Univ. Jerusalem)

Approach

- combine standard methods to produce a functioning kernel
- components:
  - kernel PCA
  - summation and inner products between vectors.
  - vertex colouring (increases number of labels on evaluation data from 63 to 1897.)

Kernel PCA

- principal component analysis applied in feature space
- implemented as eigenvalue decomposition of kernel matrix
- at most as many eigenvectors with eigenvalue $\neq 0$ as data points

$$
\begin{array}{ccc}
\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n) \in \mathcal{H} & \xrightarrow{\text{PCA}} & \phi(\mathbf{u}_1), \cdots \in \mathcal{H} \\
\phi \Big\uparrow & & \Big\downarrow \phi^{-1} \\
\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^m & \xrightarrow[\text{kernel PCA}]{} & \mathbf{u}_1, \ldots, \mathbf{u}_n \in \mathbb{R}^m
\end{array}
$$

Preprocessing of instances

Input data: $x = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^m$, labels $l_1, \ldots, l_n \in L(x)$

1. choose kernel function k between vectors (eg. RBF kernel)
2. compute kernel matrix K with $(K(x))_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$
3. compute kernel principal component vectors
4. project data points onto principal components in feature space $\Rightarrow$ transformed instances $\tilde{x} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n\}$
5. sum over component vectors with the same label

$$\phi_l(x) = \mathrm{abs}\left(\sum_{\substack{1 \leqslant i \leqslant n \\ l_i = l}} \tilde{\mathbf{x}}_i \right)$$

The kernel function

- Kernel between two instances x, x':

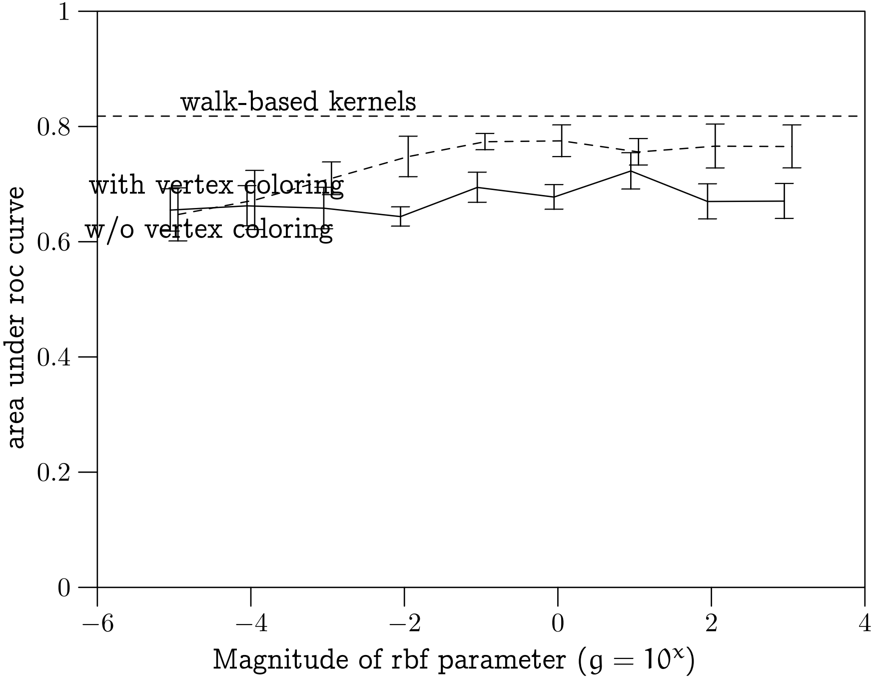
$$k_{set}(x, x') = \sum_{l \in L(x) \cap L(x')} \phi_l(x)^\top \phi_l(x')$$

- is a positive-definite kernel function, since only scalar products between vectors are used.
- May choose two kernel functions as modifier: for kernel PCA and for k_{set}. (Best results with RBF kernel for kernel PCA and linear kernel for k_{set}.)

Evaluation

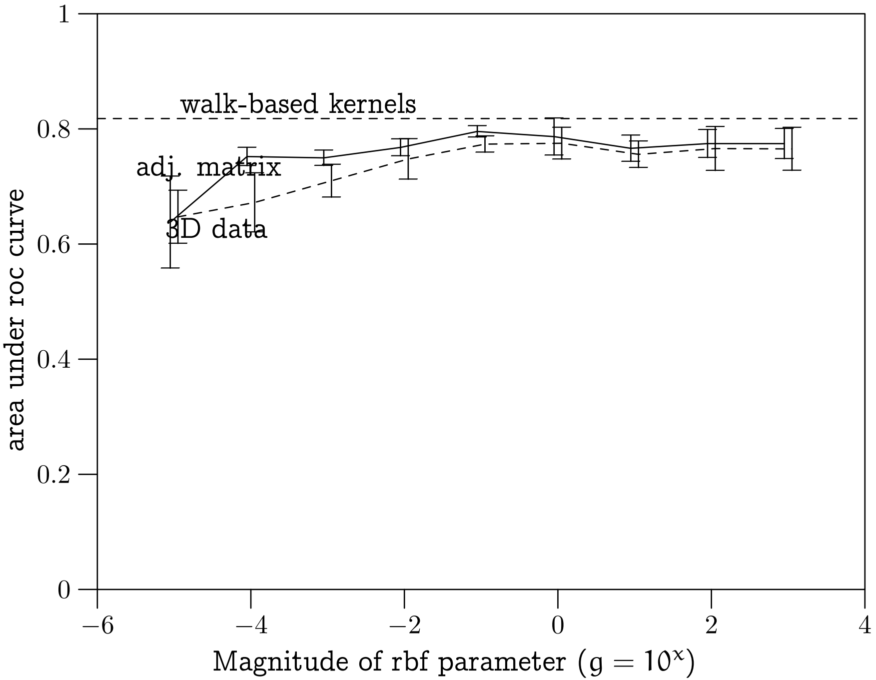
- AIDS screening data from the National Cancer Institute (March 2002)
- 3d structure data in SDF format
  - created algorithmically using CORINA (University of Nürnberg-Erlangen)
  - may be wrong (created using heuristics)
  - no data on stereochemistry available
- 42687 compounds tested
- three classes:
  - inactive (41184 compounds)
  - moderately active (1081 compounds)
  - active (422 compounds)

Experiments



Influence of vertex colouring on the area under ROC curve (data: CA vs. CM, five-fold crossvalidation.)

Experiments II



Comparison of the `pointSet` kernel on 3d data vs. adjacency lists (data: CA vs. CM, five-fold crossvalidation.)

Results

problem	cost	point set kernels on 3D data	point set kernels on adj. matrix	walk-based kernels	cyclic pattern kernels
CA vs. CM	1.0	0.774 ± 0.014	0.796 ± 0.010	0.818 ± 0.024	0.813 ± 0.014
CA vs. CM	2.5	0.767 ± 0.022	0.798 ± 0.022	0.825 ± 0.32	0.827 ± 0.013
CA vs. CM+CI	1.0	0.859 ± 0.023	0.858 ± 0.018	0.926 ± 0.015	0.908 ± 0.024
CA vs. CM+CI	100.0	0.840 ± 0.023	0.882 ± 0.022	0.928 ± 0.013	0.921 ± 0.026
CA+CM vs. CI	1.0	0.735 ± 0.017	0.732 ± 0.013	0.815 ± 0.015	0.775 ± 0.017
CA+CM vs. CI	35.0	—	0.751 ± 0.013	0.799 ± 0.011	0.801 ± 0.017
CA vs. CI	1.0	0.876 ± 0.026	0.873 ± 0.033	0.942 ± 0.015	0.919 ± 0.011
CA vs. CI	100.0	0.851 ± 0.030	0.886 ± 0.027	0.944 ± 0.015	0.929 ± 0.01

Conclusion

- viable kernel function from standard components
- efficient computation after preprocessing
- not as good as walk-based kernels/cyclic pattern kernels.
- good performance on adjacency lists not part of the design

More information in the report on `http://www.sebastian-kirsch.org/moebius/docs/praktikum.pdf`