

Walk-based Graph Kernel for Drug Discovery: A Review

Preeja M.P.

Center for Excellence in Computational Engineering and Networking
Amrita Vishwa Vidyapeetham
Coimbatore, Pin 641112 India

K.P Soman

Center for Excellence in Computational Engineering and Networking
Amrita Vishwa Vidyapeetham
Coimbatore, Pin 641112 India

ABSTRACT

The key motivation for the study of virtual screening is to reduce the time and cost requirement of the drug discovery process. Virtual screening is a computational method for finding an efficient drug molecule from pool of potential candidates. There are two different methods for virtual screening 1) structure based 2) ligand based. In the structure based method, 2D or 3D structure of a target molecule is used to screen the ligands which do not bind to the target molecule. Ligand based virtual screening is based on the fact that ligands similar to an active drug molecule might be active. The amount of information required is different in both the case. Structure based virtual screening is computationally intensive and complex while a few active ligand information is enough to start ligand based virtual screening. Based on the type of information, ligand based virtual screening can be performed in different ways. The machine learning approach using molecular graphs has been found to be very effective. Graph kernel is the similarity measure used to screen molecular graphs based on the structure. It is based on the fact that structurally similar molecules will have same property. In this review we have summarized the recent development in graph kernel for chemical molecule and elaborated upon the need of more accurate and efficient graph kernel with less computational complexity. The accuracy of different methods have been compared using standard dataset. The review shows the current state of art in the ongoing research in the design of efficient walk kernels.

General Terms

Pattern Recognition, Classification, chemical compounds.

Keywords

Drug Discovery, Virtual screening, SVM, Walk kernel.

1. INTRODUCTION

Increased amount of structured and semi-structured data in various fields entail for efficient data mining techniques. In pharmaceutical sector, huge amount of chemical information present in the databases pose new challenges for machine learning applications in computational drug discovery. One of the sources of active drug is natural extracts. Combinatorial chemistry can produce millions of compounds by changing the constituents of a compound. Combinatorial chemistry is a useful tool for making more effective drugs from the already existing ones. Computational techniques help to analyse the property of a drug even before it is synthesized. In this way, computational techniques can reduce the time and cost requirements of drug discovery process. Machine learning technique requires efficient data handling. Chemical databases contain nearly 10^{60} molecules from which effective drug

molecule can be pin pointed using efficient virtual screening. Virtual screening is an in-silico process of picking up effective drug from a pool of chemical compounds (virtual chemical library). The long list of compounds for the in-vivo and in-vitro analysis can be narrowed down by efficient virtual screening. Information about either the target molecule or the ligand or both is required for virtual screening. Based on the available information virtual screening can be accomplished in two ways –ligand based and structure based virtual screening. Ligand based virtual screening utilizes topological, structural or pharmacophore information. It is used when no information about the target is available. Structure based virtual screening is based on the interaction between target and ligand, wherein the structure of the target molecule is already known. Docking is the methodology used for the structure based virtual screening in which the ligand molecule binds a appropriately with the target. The drug like property of ligand is based on its ADMET (adsorption, distribution, metabolism, excretion and toxicity) properties [1]. In ligand based virtual screening ADMET property, lipophilicity and molecular weights are the commonly used filters [2]. Efficient storage and recovery of chemical information is required for combinatorial chemistry and virtual screening. Computational techniques like combinatorial chemistry and virtual screening reduces the time and cost of drug discovery. Ligand based screening can be performed in different ways but machine learning method is the preferred and popular choice. Decades of research in machine learning has provided a wide variety of methods like supervised learning, reinforcement learning, transduction etc.

Recent research works show that Support Vector Machine (SVM) can be used as an efficient tool for virtual screening in drug discovery. It can be applied to high dimensional space with less computational complexity. Some authors have introduced SVM-based methods for the prediction of ADMET properties of chemical and biological molecules [3]-[5]. Successful results have been obtained using one dimensional representation of the molecules like fingerprint and other types of descriptors. Recent works [6]-[24] show that higher dimensional representation can give more accurate prediction.

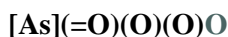
In recent years, various approaches have been proposed for classification of chemical drugs [6]-[12]. These approaches are generally grouped into three major categories based on the dimensionality used for the representation of molecules

i) One-dimensional (1-D) representations: 1D representations give information about the constituents of the molecule [6]-[9]

ii) Two-dimensional (2-D) graph gives topological information of the molecule [6]-[31];

iii) Three-dimensional (3-D) and other higher dimensional graphs include the surface information, pharmacophore and different configurations of molecules like conformers and isomers of molecules [7]-[9].

Figure 1.shows different representations of arsenic acid. Recent research works indicate that 3-D structure based classification is used for more accurate result, although 2-D structure based classification is widely used due to its lesser computational complexity.



(a) SMILE notation

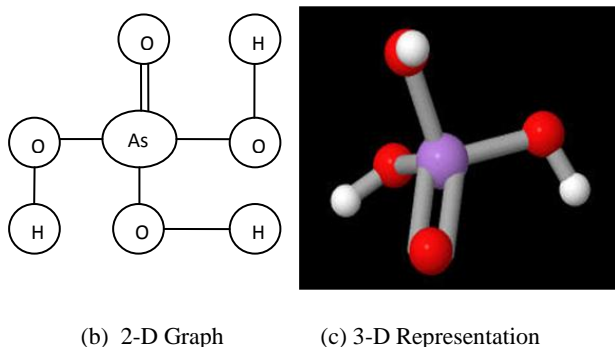


Figure1: Different structures of arsenic acid

2. SUPPORT VECTOR MACHINE

SVM is a widely used machine learning method for Virtual screening. SVM is a supervised learning method used for classification and regression. In support vector classification, X is the input data space and Y is the output space.

Support vector classification has two phases, training phase and testing phase. In training phase $X = \{x_1 \dots x_n\}$ is the training data and $Y := \{y_1 \dots y_n\}$ are the corresponding class labels (target values). x_i is the feature vector (attributes), dimension of feature vector depends upon the number of features used for the classification. In the training phase input data and the corresponding class labels are used to find classifier. Testing phase is to find the accuracy of the classifier.

In binary classification, $Y := \{+1, -1\}$ finding the parameters of classifiers is a convex optimization problem. Objective of Soft margin SVM is to find classifier with maximum margin and minimum number of misclassification. Number of misclassification can be reduced by including error term into the objective function. ϵ_i measures the degree of misclassification of x_i . Figure 2 represents the classifier and the bounding planes.

$$\min_{\omega, \gamma, \epsilon} \frac{\omega^T \omega}{2} + C \sum_{i=1}^N \epsilon_i \quad (1)$$

Subject to

$$y_i (w^T x - \gamma) + \epsilon_i \quad (2)$$

$$\epsilon_i \geq 0, 1 \leq i \leq N$$

Solution of the above convex optimization gives the values of ω and γ . The decision function is

$$f(x) = \text{sign} (\sum_{i \in SV} u_i y_i \langle x, x_i \rangle) \quad (3)$$

where u_i is the Lagrangian multiplier and $\langle x, x_i \rangle$ is the linear kernel function or similarity between x and x_i . Non-linearly separable data kernel function is $\phi(x), \phi(x_i)$ and $\langle \phi(x), \phi(x_i) \rangle$.

In equation (3) $x_i^T x_i$ is known as linear kernel and is useful for the classification of linearly separable data. Most of the real world datasets are not linearly separable. Nonlinear data in the input space can be classified by using linear classifier in the high dimensional space. ϕ is the transformation from input space X to feature space, $\phi: X \rightarrow H$, both these spaces are equipped with dot product. Figure 3 explains the feature mapping process.

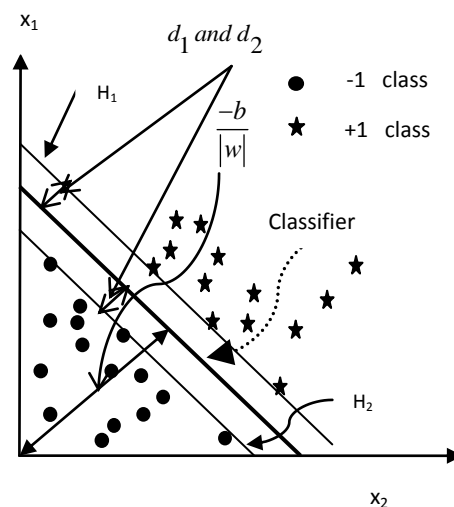


Figure 2: Illustration of classifier finding process

Kernel method is a set of algorithms in which inner product is replaced by a kernel function; kernel function is the measure of similarity between data points. Kernel method will reduce the computational complexity for the classification of non-linear data. Kernel matrix should be positive definite and the kernel should satisfy Mercer's property. In kernel matrix each element is an inner product between data points in the finite sample set $S, K_{i,j} = x_i^T x_j$. Kernel matrix is symmetric because $K(x, y) = K(y, x) = \phi(x)^T \phi(y)$. Mercer's theorem states that the symmetric function k is a kernel function for any finite sample space S if the kernel matrix for S is positive semi-definite. Gaussian kernel and Dirac kernel are the two famous Mercer's kernels.

Example of Mercer's kernel on structured data include walk kernel for undirected graphs, subtree kernel, shortest path kernel etc.

Kernel matrix for finite set S with N data points

$$K = [k(x_i, x_j)]_{1 \leq i, j \leq N}$$

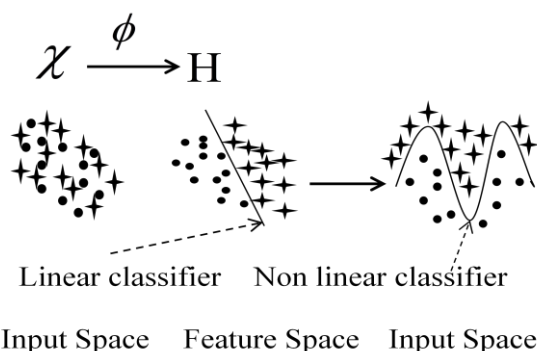


Figure 3: Illustration of transformation of data from input space to feature space.

3. GRAPHICAL REPRESENTATION OF CHEMICAL COMPOUNDS

Most of the real world problems can be modeled using graphs. Graphs can include topological information because of which chemical compounds are often represented as graphs. Graphical representation is very effective for the efficient storage, retrieval and reuse of the information. Graphical representation of data helps to classify the drug molecules based on ADMET properties. Chemical compounds can be represented as directed labeled, directed unlabelled, undirected labeled or undirected labeled graphs. Labeled undirected graphs are most commonly used in chemoinformatics. Graph G is defined by a finite set of vertices $V = (v_1, v_2 \dots v_n)$ and a finite set of edges $E = (e_1, e_2 \dots e_n)$. Label of nodes corresponds to the label of atoms and edge label corresponds to the bond between atoms. Apart from this many different types of labeling can be used to include further information.

4. GRAPH KERNEL

Structural data can be represented as a feature vector with numerical entries or with local structures. Data like gene sequences, small molecules, interaction networks and phylogenetic trees require pre-processing. Representation of structured data using numerical values require huge amount of pre-processing. Pre-processing is required to find the corresponding properties of each molecule from the structure analysis. In graph kernels, the feature vector of graph contain substructures (local structures) of graph like walk, subtrees, subgraph etc.

Two graphs G_1 and G_2 are same if all the local structures of both these graphs are identical. Graph Kernel is based on this type of comparative principle. Different types of graph kernels exist based on the types of features used for comparison. Walk kernels, tree kernels, diffusion kernels, wavelet alignment kernels, shortest path kernel and pharmacophore kernel are unremarkably used. The kernel function will return the similarity between substructures of two graphs. Recent research in kernel methods allows the learning in an infinitesimal dimensional space.

Graph kernel between two graphs G_1 and G_2 is defined as

$$k(G_1, G_2) = \sum_{i \in \text{substr}_1} \sum_{j \in \text{substr}_2} k(\text{substr}_i, \text{substr}_j) \quad (4)$$

where substr_1 and substr_2 are the set of substructures of graph G_1 and G_2 .

5. WALK KERNEL

Walk is a non-empty sequence of nodes $(v_1, v_2 \dots v_m)$ in graph G such that the adjacent nodes are connected by an edge. Figure 4 shows all the possible walks in the chemical graph of C_2H_4O starting from the first vertex. Walk with no repeated nodes is known as path. Cycle is also a path with $v_1 = v_n$ and $n \geq 4$. Length of walk is equal to the number of edges in the walk or *no of nodes* – 1. Random walk in a graph from particular vertex is walk, where the selection of adjacent vertices is a random process.

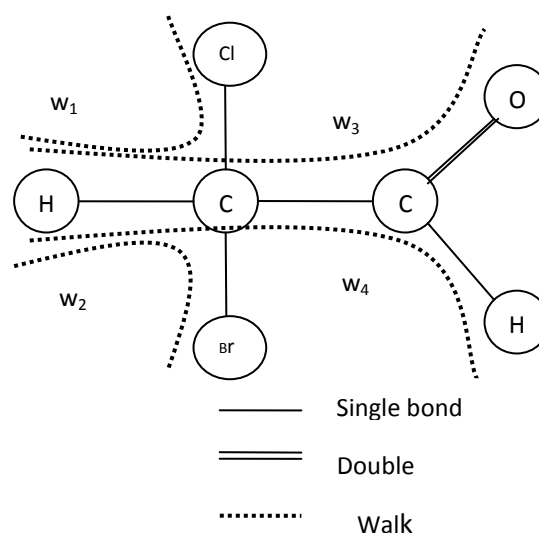


Figure 4: illustration of all possible Walks from first vertex v_1 in Graph G

Walk kernel was first proposed by Kashima and Inokuchi in 2002 [12]. Different authors have proposed walk kernels with different modifications [11–22].

Walk kernel between G and G' is defined as

$$K(G, G') = \sum_{w \in W} \sum_{w' \in W'} k(w, w') \quad (5)$$

where W and W' are the set of possible walks on graphs G and G' .

Most of the walk kernels are based on product graphs. There are four types of product graphs wherein direct product graph is used for walk kernel computation. In product graph kernel between Graphs G and G' is based on the number of common walks present in both graphs. Number of common walks is equal to the number walks in the product graph and it can be found out using adjacency matrix of product graph. Number of N length walk between *node i* and *node j* is the ij^{th} element in the N^{th} power of adjacency matrix of the product graph.

Vertex set and Edge set of Direct Product graph G_X of Graph G and G' are defined respectively

$$V(G \times G') = (v_i, v'_i) \in V_x: \text{if } \text{label}(v_i) = \text{label}(v'_i) \quad (6)$$

$$E(G \times G') = (v_i, v'_i)(v_j, v'_j) \in E_x \quad (7)$$

$$\text{if } \left\{ \begin{array}{l} (v_i, v_j) \in E \\ (v'_i, v'_j) \in E_x \\ \text{label}(v_i, v_j) = \text{label}(v'_i, v'_j) \end{array} \right\}$$

• Marginalized Walk Kernel

Kashima and Inokuchi in 2002 [12] proposed random walk kernel based vertex product graph with halting probability λ . The value of λ depends on the application and its value varies from 0 to 1. In the first part, authors have proposed similarity by comparing the vertex labels of walks in both graphs and in the second part local information are also included by comparing the labels of vertices and labels of its adjacent edges and vertices. The kernel function $K(G, G')$ between Graphs G and G' is defined as

$$K(G, G') = \frac{1}{|V||V'|} \sum_{v_1 \in V} \sum_{v'_1 \in V'} k(v_1, v'_1) \quad (8)$$

Where $k(v_1, v'_1)$ is defined as a delta function

$$k(v_1, v'_1) = \begin{cases} 1, & \text{label}(v_1) = \text{label}(v'_1) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Local information is added to the vertex labels for improving the expressiveness. The computational complexity is also high for this modified kernel between vertices in graphs G and G' .

Modified kernel function $k(v_1, v'_1)$

$$\begin{aligned} k(v_1, v'_1) &= (1 - \lambda) k_0 + \lambda(1 - \lambda) \sum_{\substack{e_1 \in E(v_1) \\ e'_1 \in E(v'_1)}} k_1(v_1, v'_1, e_1, e'_1) \\ &+ \lambda^2(1 - \lambda) \sum_{\substack{e_1 \in E(v_1) \\ e'_1 \in E(v'_1)}} \sum_{\substack{e_2 \in E(N(v_1, e_1)) \\ e'_2 \in E(N(v'_1, e'_1))}} k_2(v_1, v'_1, e_1, e'_1, e_2, e'_2) \end{aligned} \quad (10)$$

Where $N(v, e)$ returns the neighbour of vertex v connected through edge e and $E(V)$ is the set of edges adjacent to v .

Kernel function defined by Kashima and Inokuchi [12] have a lot of similarity with diffusion kernel described by Lafferty and Lebanon in 2002 [11]. Diffusion kernel [11] is symmetric kernel and is for undirected graphs, while kernel [12] is asymmetric and is applicable for directed graphs. Kernel proposed Kashima and Inokuchi [12] is symmetric for undirected graphs and is useful for structures and semi-structured data. The accuracy of this kernel is appreciable compared to other existing kernels, but the computational complexity is very high for this type of kernels.

Kashima et al., 2003 [15] proposed marginalized kernel for labelled graphs. In Marginalized kernel the kernel computation is based on the latent variable, latent variables are those variables which cannot be measured directly but can be deducing from the observed variables. Tsuda et al., [13] introduced marginalized kernel for biological sequence in

2002. Kernel used by Kashima et al., [15] is the special case of this kernel. Kernel proposed by Tsuda et al., [13] is based on the method used by Jaakkola and Haussler in 1998 [14]. Tsuda et al., [13] proposed two types of marginalized kernels, Marginalized count kernel and Second order marginalized count kernel, and it's depends on both hidden variables and visible variables. The joint kernel will return the similarity between sequence of vertex and edge labels traversed by random walk. In Marginalized kernel [15] the random walk depends on the starting probability, transition probability and stopping probability. Instead of finding all feature vectors explicitly, all possible random walks are comparing. In the absence of prior knowledge the initial probability and transition probability will follow uniform distribution and stopping probability is equal to constant. Kashima et al., [15] defined marginalized kernel between labelled graphs Graph G and G' as

$$\begin{aligned} K(G, G') &= \sum_{l=1}^{\infty} \sum_h \sum_{h'} p_s(h_1) \prod_{i=2}^l p_t(h_i|h_{i-1}) p_q(h_l) \\ &\times p'_s(h'_1) \prod_{i=2}^l p'_t(h'_i|h'_{i-1}) p_q(h'_l) \\ &\times K(v_{h_1}, v_{h'_1}) \prod_{k=2}^l K(e_{h_{k-1}h_k}, e'_{h'_{k-1}h'_k}) K(v_{h_k}, v_{h'_k}) \end{aligned} \quad (11)$$

where $p_s(h)$ is the initial probability distribution and $p_t(h_i|h_{i-1})$ is the transition probability.

Kashima et al., in 2004 [17] proposed random walk kernels for different types of directed graphs. Authors used random walks as features and it is represented as a sequence of vertex labels and edges labels alternatively. The label sequence kernel $K_Z(h, h')$ for a sequence h and h' is defined as a product of all label kernels in those sequences. Label kernels are used to compare the labels. Two types of label kernels are used for comparing walks edge label kernel K_e and K_v vertex label kernel. The label kernel for non-numerical data is calculated using delta function and gaussian kernel for numerical data.

Label sequence kernel [17] is defined as

$$K_Z(h, h') = K_V(h, h') \prod_{i=2}^l K_e(h_{2i-2}, h'_{2i-2}) K_v(h_{2i-2}, h'_{2i-2}) \quad (12)$$

Label sequence kernel is equal to zero if the length of walks are different.

The label sequence kernel between graphs G and G' is calculated as the expectation of $K_Z(h, h')$ of all possible walks in both the graphs.

$$K(G, G') = \sum_h \sum_{h'} K_Z(h, h') p(h|G) p(h'|G') \quad (13)$$

where $p(h|G)$ is the probability of getting path h in graph G .

Kashima et al., [17] proposed label sequence kernel for acyclic graphs, general directed graphs and for graph with multiple edges between vertices.

Mahe et al., in 2005[18] proposed new kernel for molecules with label enrichment using Morgan index where tottering is avoided using second order Markov random walk. Label enrichment techniques increase the specificity of labels; more specific labelling reduces the number of common walks between graphs. Reduction of common walks further reduces the time requirement for the kernel computation. In Morgan indexing local environment information is used for atom labelling instead of atom symbol. In case of molecules, it is essential to distinguish atoms with same label belonging to different functional group. Authors modified Marginalized kernel [15] using second order Markov random walk and label enrichment using Morgan Indices. Second order Markov random walk reduces the number of random walks by avoiding tottering walks.

• Random Walk Kernel

Gartner et al., in 2003 [16] proposed kernels for contiguous label sequence and non-contiguous label sequences. Both these kernel computations are based on the direct product graph. In case of contiguous label sequence direct product kernel and for non-contiguous label sequence product graph with gap penalty is proposed [16].

Direct product kernel for contiguous label sequence in graph G and G' as

$$K(G, G') = \sum_{i,j=1}^{|V_X|} [\sum_{n=0}^{\infty} \lambda_n E_X^n]_{i,j} \quad (14)$$

and for non-contiguous label sequence in graph G and G' as

$$K(G, G') = \sum_{i,j=1}^{|V_X|} [\sum_{n=0}^{\infty} \lambda_n [(1 - \alpha)E_X + \alpha E_X]^n]_{i,j} \quad (15)$$

Where λ is the weight given for each sequence $\lambda = \lambda_0, \lambda_1, \dots (\lambda_i \in \mathfrak{R}, \lambda_i \geq 0) \forall i \in \mathbb{N}$ and α is for penalizing the gap and its value varies from [0,1].

• Fast Computing Random Walk Kernel

Vishwanathan et al., [22] calculated similarity between graphs using direct product kernel in 2006. The adjacency matrix A_X of the direct product graph G_X between graphs G and G' is

defined as the Kronecker product between adjacency matrixes of graph G and G' .

$$A_X = G \otimes G' \quad (16)$$

Where A and A' are the adjacency matrices of graph G and G'

Authors defined random walk kernel between unlabelled graphs G and G' as.

$$K(G, G') = \sum_{n=0}^{\infty} \tau(i) q_X^T A_X^n p_X \quad (17)$$

where p_X and q_X are the starting and stopping probability distributions. λ is the weight given for different length walks.

In case of edge labelled graph G and G' random walk kernel is defined using edge weighted adjacency matrix W_X of the product graph G_X

$$K(G, G') = \sum_{n=0}^{\infty} \lambda(i) q_X^T W_X^n p_X \quad (18)$$

Vishwanathan et al., [22] have proposed three efficient ways of computation of the above equation. They used conjugate gradient and fixed point iteration for utilizing sparsity in reducing the computational complexity from $O(n^6)$ to sub-cubic complexity and Sylvester equation for reducing the complexity from $O(n^6)$ to $O(n^3)$.

6. RESULT AND DISCUSSION

We compared the accuracy of each method using PTC (Predictive Toxicology for Cancer) dataset. All the codes were written in JAVA and the experiments were run on Intel Pentium 4 PC with 4GB main memory. The prediction accuracy was calculated for different methods and is specified in Table1.

7. CONCLUSION

Molecular graphs are very difficult to represent as vector with numerical entries. Graph kernel tackles this problem by using substructures as features for the representation of graphs. These features are used to find the structural similarity between graphs. In this paper, we have discussed different types of walk kernels and the methodology used to get more accurate results. Different types of walks are used for feature extraction. The aim of this paper is to provide a comprehensive approach to discuss about different types of walk kernels for chemical graphs.

8. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of walk kernels.

Table 1. Accuracies of different types of walk kernel

Methods	Type of graph	Features	Methodology	C	Accuracy
Kashima et.al (2002)	Both edge and vertex labeled graph	Random walk	Vertex product graph	0.0011	59.6
Kashima et.al (2003) [16]	Labeled graphs	First order markov random walk	Marginalized Kernel	0.01	60

Kashima et al.(2004)[18]	Directed acyclic labeled graph	First order markov random walk	Marginalized Kernel	0.005	61(Avg)	
	General directed labeled graph					
	Labeled graphs with multiple edges between vertices					
Gartner et al.(2003)[17]	Labeled graphs	Labels of first and last vertex of a walk	Marginalized Kernel	0.01	62.08	
	Labeled graphs	Contiguous label sequence of a walk				
	Labeled graphs	Noncontiguous label sequence of a walk				
Mahe et al.(2005)[19]	Labeled graphs	Second order markov random walk	Marginalized Kernel	0.1	64.02	
	Labeled graph (Label enrichment using MI)	Second order markov random walk				
Vishwanathan et al.(2006)[23]	Unlabeled graph	Random walk Kernel	Conjugate gradient	Efficient kernel computation	0.001	61.8
	Edge weighted graphs		Sylvester Equation			
			Fixed point			

9. REFERENCES

- [1] Tingjun Hou and Xiaojie Xu, Recent Development and Application of Virtual Screening in Drug Discovery: An Overview, *Current Pharmaceutical Design*, pp. 1011-1033, vol. 10, 2004
- [2] Vivek Vyas, Anurekha Jain, Avijeet Jain, Arun Gupta, Virtual Screening: A Fast Tool for Drug Design, *Sci Pharm.*, pp. 333-360, vol. 76, pp. 0803-03, 2008.
- [3] S. Kramer, E. Frank, and C. Helma, Fragment generation and support vector machines for inducing SARs, *SAR QSAR Environ Res.*, vol. 13, no. 05, pp. 509-523, July 2002.
- [4] 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.*, vol. 44, no. 04, pp. 1402-1411, 2004.
- [5] Feng Luan, Ruisheng Zhang, Chunyan Zhao, Xiaojun Yao, Mancang Liu, Zhide Hu, and Botao Fan, Classification of the carcinogenicity of N-nitroso compounds based on support vector machines and linear discriminant analysis, *Chem Res Toxicol.*, vol. 18, no. 02, pp. 198-203, Feb 2005.
- [6] S. Joshua Swamidass, Jonathan Chen, Jocelyne Bruand, Peter Phung, Liva Ralaivola and Pierre Baldi, Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity, *Bioinformatics 2005, ISMB 2005 Proceedings*, Vol 21, no 01, i359-i368. March 27, 2005.
- [7] Pierre Mahé, Liva Ralaivola, Véronique Stoven and Jean-Philippe Vert, The Pharmacophore Kernel for Virtual Screening with Support Vector Machines, *J. Chem. Inf. Model.*, vol. 46 no. 05, pp 2003–2014, Aug 2006.
- [8] L. Ralaivola, S. J. Swamidass, J. H. Saigo, and P. Baldi, Graph kernels for chemical informatics, *In Advances in Neural Information Processing Systems*, vol. 19, no. 01, pp. 1093-1110, Sep 2005.
- [9] C.-A. Azencott, A. Ksikes, S. J. Swamidass, J. H. Chen, L. Ralaivola, and P. Baldi, One- to four dimensional kernels for virtual screening and the prediction of physical, chemical, and biological properties, *J Chem Inf Model*, vol. 47 no. 03 pp. 965-974, 2007.

- [10] Gartner, T., Exponential and geometric kernels for graphs, *NIPS-02 Workshop on Unreal Data: Principles of Modelling Nonvectorial Data*, Whistler, British Columbia, December, 2002.
- [11] J. Lafferty and G. Lebanon, Information diffusion kernels, *In Neural Information Processing Systems*, volume 15. MIT Press, 2002.
- [12] H. Kashima and A. Inokuchi, Kernels for Graph Classification, *Proc. 1st ICDM Workshop on Active Mining (AM-2002)*, Maebashi, Japan, 2002.
- [13] Koji Tsuda, Taishin Kin and Kiyoshi Asai, Marginalized kernels for biological sequences, *Bioinformatics*, Vol 1 pp.1-8, 2002.
- [14] T. Jaakkola and D. Haussler, Exploiting generative models in discriminative classifiers, *In Advances in Neural Information Processing Systems*, Vol 11, 1998.
- [15] H. Kashima, K. Tsuda and A. Inokuchi, Marginalized Kernels between Labeled Graphs, *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 321-328, AAAI Press, 2003.
- [16] T. Gartner, P. Flach, and S. Wrobel, On graph kernels: hardness results and efficient alternatives, *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory and the Seventh Annual Workshop on Kernel Machines*, vol. 2777 of Lecture Notes in Computer Science, pp. 129-143, July 2003.
- [17] H. Kashima, K. Tsuda, and A. Inokuchi, Kernels for graphs, *Kernel Methods in Computational Biology*, pp. 155-170. MIT Press, 2004.
- [18] P. Mahe, N. Ueda, T. Akutsu, J.L. Perret, and J.P. Vert, Graph kernels for molecular structure-activity relationship analysis with support vector machines, *J ChemInf Model*, vol. 44, no. 04, pp. 939-951, 2005.
- [19] J. Ramon and T. Gartner, Expressivity versus efficiency of graph kernels, *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, pp. 65-74, 2003.
- [20] K. M. Borgwardt and H.P. Kriegel, Shortest-path kernels on graphs, *Proc. 5th IEEE Int. Conf. on Data Mining*, Houston, Texas, USA, pp. 74-81, 2005.
- [21] P. Mahe and J.P. Vert, Graph kernels based on tree patterns for molecules, *Technical Report ccsd-00095488*, HAL, September 2006.
- [22] S. V. N. Vishwanathan, Karsten M. Borgwardt, and Nicol N. Schraudolph, Fast Computation of Graph Kernels, *In Advances in Neural Information Processing Systems*, vol. 19, MIT Press, Cambridge MA, 2006.
- [23] *Computer-Aided Molecular Design*, Vol 12, pp.471 – 490, 1998.
- [24] M. Collins and N. Duffy, Convolution kernel for natural language, *In Proc. of the 14th NIPS*, Vancouver, British Columbia, Canada, 2001.
- [25] S. V. N. Vishwanathan and Alexander J. Smola, Fast Kernels for String and Tree Matching, *Kernel Methods in Computational Biology*, MIT Press, Cambridge, MA, 2004.
- [26] Shervashidze N. and K. M. Borgwardt, Fast subtree kernels on graph, *Proceedings of the 2009 Conference NIPS*, BC, Canada, pp.1660-1668, 2009.
- [27] T. Gartner, J.W. Lloyd, and P.A. Flach, Kernels and distances for structured data, *Journal Machine Learning*, Vol 57, no 03, pp.205-232 September 2006.
- [28] C.Z. Cai, W.L. Wang, L.Z. Sun, and Y.Z. Chen, Protein function classification via support vector machine approach, *Math. Biosci.*, vol. 185, no. 02, pp. 111-122, 2003.
- [29] H. Frohlich, J. K. Wegner, F. Sieker, and A. Zell, Optimal assignment kernels for attributed molecular graphs, *In Proceedings of the 22nd international conference on Machine learning*, New York, NY, USA, pp. 225-232, 2005.
- [30] T. Horvath, T. Gartner, and S. Wrobel, Cyclic pattern kernels for predictive graph mining, *In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, pp. 158-167, 2004.
- [31] D. Haussler, Convolution Kernels on Discrete Structures, *Technical Report UCSC-CRL-99-10*, UC Santa Cruz, 1999.
- [32] Matthias Rarely and J. Scott Dixon, Feature trees: A new molecular similarity measure based on tree-matching, *J.*