An Efficient Sampling Scheme For Comparison of Large Graphs

Karsten M. Borgwardt (kb@dbs.ifi.lmu.de), Tobias Petri, S.V.N. Vishwanathan, and Hans-Peter Kriegel

As new graph structured data is being generated, graph comparison has become an important and challenging problem in application areas such as molecular biology, telecommunications, chemoinformatics, and social networks. Graph kernels have recently been proposed as a theoretically sound approach to this problem, and have been shown to achieve high accuracies on benchmark datasets. Different graph kernels compare different types of subgraphs in the input graphs. So far, the choice of subgraphs to compare is rather ad-hoc and is often motivated by runtime considerations. There is no clear indication that certain types of subgraphs are better than the others. On the other hand, comparing all possible subgraphs has been shown to be NP-hard, thus making it practically infeasible. These difficulties seriously limit the practical applicability of graph kernels.

In this article, we attempt to rectify the situation, and make graph kernels applicable for data mining on large graphs and large datasets. Our starting point is the matrix reconstruction theorem, which states that any matrix of size 5 or above can be reconstructed given all its principal minors. By applying this to the adjacency matrix of a graph, we recursively define a graph kernel and show that it can be efficiently computed by using the distribution of all size 4 subgraphs of a graph. This distribution, we argue, is similar to a sufficient statistic of the graph, especially when the graph is large. Exhaustive enumeration of these subgraphs is prohibitively expensive, scaling as $O(n^4)$. But, by bounding the deviation of the empirical estimates of the distribution from the true distribution, it suffices to sample a fixed number of subgraphs. Incidentally, our bounds are stronger than those found in the bio-informatics literature for similar techniques.

In our experimental evaluation, our graph kernel outperforms state-of-the-art graph kernels both in times of time and classification accuracy.

Categories and Subject Descriptors: []:

General Terms: Graph Similarity

1. INTRODUCTION

Graph models have been studied in fields as disparate as bioinformatics, chemoinformatics, chemistry, sociology and telecommunication. In these application areas graphs are used to model, for instance, protein-interaction networks, protein structures, social networks, or telecommunication networks. A question that one often encounters is: "Does this graph belong to a certain class of graphs or not"? For example, does a protein structure belong to the class of enzymes [Borgwardt et al. 2005]? Does a chemical compound belong to the class of compounds that show an effect against a certain disease [Horvath et al. 2004]? Does a group of people represent a special kind of social network, e.g. a family or a terrorist network? These questions are instances of a classification task on graphs. Exact matching rarely suffices, and one needs to learn to predict the class membership based on labeled training examples. In the machine learning community, algorithms have been developed which when given an oracle to predict similarity between the structure

of the input graph with that of previously seen graphs can use this information for prediction. In this paper, we will concentrate on developing an efficient graph comparison algorithm, *i.e.,* an oracle.

Broadly speaking, existing graph comparison algorithms may be classified into three categories: *set* based, *frequent subgraph* based, and *kernel* based. Set based approaches represent a graph as a set of edges, or as a set of nodes, or both. Graphs are then compared by measuring similarity between pairs of edges or pairs of nodes between two graphs. While these approaches are computationally feasible, they are rather naive, as they neglect the structure of the graphs, *i.e.,* their topology.

Frequent subgraph mining algorithms, on the other hand, aim to detect subgraphs that are frequent in a given dataset of graphs [Kuramochi and Karypis 2001; Inokuchi et al. 2003; Yan and Han 2002]. Afterwards, feature selection is applied to select the most discriminative subgraphs. Efficient heuristics such as gspan [Yan and Han 2002] have been developed for this task. Unfortunately, their computational complexity scales as exponential in the worst-case.

Graph kernels represent an attractive middle ground. They respect and exploit graph topology, but restrict themselves to comparing substructures of graphs that are computable in polynomial time. Many different graph kernels have been defined, which focus on different types of substructures in a graphs, such as random walks [Gärtner et al. 2003; Kashima et al. 2004], paths, subtrees [Ramon and Gärtner 2003], and cycles [Horvath et al. 2004]. Several studies have recently shown that these graph kernels can achieve state-of-the-art results on problems from bioinformatics [Borgwardt et al. 2005] and on benchmark datasets from chemistry [Ralaivola et al. 2005].

When using graph kernels, a practitioner is often faced with three questions: Out of the numerous variants, which graph kernel should I choose for a particular application? Does this kernel capture graph similarity better than the others? Is it cheap to compute? Unfortunately, these questions are far from being answered. Almost all existing approaches are ad-hoc and are generally motivated by the runtime considerations. To make matters worse, there is no theoretical justification on why certain types of subgraphs are better than the others. The other extreme of comparing all possible subgraphs has been shown to be NP-hard [Gärtner et al. 2003], thus making it practically infeasible. These difficulties seriously limit the practical applicability of graph kernels.

In essence, this problem could be solved by a rich enough representation that adequately captures the topology of the input graphs, while being cheap to compute. Drawing analogy from probability theory, we seek to efficiently compute the sufficient statistics of the graph.

1.1 Paper Contributions

Motivated by the matrix reconstruction theorem, which states that a matrix of size 5 or above can be reconstructed given all its principal minors, we recursively define a kernel on a graph and its principal subgraphs. We then show that our kernel can be efficiently computed by using the distribution of all size 4 subgraphs of a graph. This distribution, we argue, is similar to a sufficient statistic of the graph, especially when the graph is large. Exhaustive enumeration of these subgraphs is prohibitively expensive, scaling as $O(n^4)$. But, by bounding the deviation of

the empirical estimates of the distribution from the true distribution, it suffices to sample a fixed number of subgraphs. Incidentally, our bounds are stronger than those found in the bio-informatics literature for similar techniques. Finally, we will present experimental evidence to support our claims at the talk

2. SAMPLING FROM GRAPHS

In order to compute our kernel exactly one needs to exhaustively enumerate all graphlets of size 4 in the input graphs. Suppose, a given graph has n nodes, then there are $\binom{n}{4}$, or equivalently $O(n^4)$, graphlets, If the graphs are small, then this is feasible. But on large graphs (with n of the order of hundreds or thousands or more), runtime will degenerate. In this case one needs to resort to sampling.

The problem of sampling subgraphs from graphs has been widely studied in bio-informatics [Przulj 2007; Kashtan et al. 2004; Wernicke 2005]. Unfortunately, the algorithms proposed there are rather ad-hoc and do not provide bounds on the sample complexity. Recently, [Weissman et al. 2003] proved distribution dependent bounds for the L_1 deviation between the true and the empirical distributions. We adapt their results and derive sample complexity bounds which are much stronger than any previously known result for this problem.

2.1 Sample Complexity Bound

Let $\mathcal{A} = \{1, 2, \ldots, a\}$ denote a finite set of elements. For two probability distributions P and Q on $\mathcal{A}$, the L_1 distance between P and Q is defined as

$$||P - Q||_1 := \sum_{i=1}^{a} |P(i) - Q(i)|. \tag{1}$$

Given a multiset $X := \{X_j\}_{j=1}^m$ of independent identically distributed (iid) random variables X_j drawn from some distribution D, the empirical estimate of D is defined as

$$\hat{D}^m(i) = \frac{1}{m} \sum_{i=1}^{m} \delta(X_j = i), \tag{2}$$

where $\delta(\cdot)$ denotes the indicator function of the specified event. For $p \in [0, 1/2)$, define

$$\psi(p) = \frac{1}{1-2p} \log \frac{1-p}{p}, \tag{3}$$

and set $\psi(1/2) = 2$. Note that $\psi(p) \geq 2$ for all valid p. Furthermore, for a probability distribution D on $\mathcal{A}$ define:

$$\pi_D := \max_{A \subseteq \mathcal{A}} \min\{D(A), 1 - D(A)\}. \tag{4}$$

THEOREM 1. *[Weissman et al. 2003] Let D be a probability distribution on the finite set $\mathcal{A} = \{1, \ldots, a\}$. Let $X := \{X_j\}_{j=1}^m$, with $X_j \sim D$. Then for all $\epsilon > 0$*

$$P\left\{||D - \hat{D}^m||_1 \geq \epsilon\right\} \leq (2^a - 2)e^{-m\psi(\pi_D)\epsilon^2/4}. \tag{5}$$

The following corollary is straightforward:

COROLLARY 2. *Let D, $\mathcal{A}$, and X as above. For a given $\epsilon > 0$ and $\delta > 0$, at least*

$$m \geq \frac{2\left(\log 2 \cdot a + \log\left(\frac{1}{\delta}\right)\right)}{\epsilon^2} \tag{6}$$

samples are required to ensure that $P\left\{||D - \hat{D}^m||_1 \geq \epsilon\right\} \leq \delta$.

2.2 Implications of the Bound

First of all, notice that the bound, (6), is independent of n, the size of the graph. What this means in practise is that our sampling algorithm is highly scalable and works even for very large graphs. Secondly, notice that our sample complexity bound only had an additive dependence on a, the size of the set over which the distribution is defined. In our case, there are a total of 64 possible graphlets of size 4. But, modulo isomorphism, there are only 11 distinct graphlets [Przulj 2007].

3. CONCLUSIONS

In our talk, we will elaborate on the theory summarized above, and present experimental evidence to support our findings on comparison of large graphs.

REFERENCES

BORGWARDT, K. M., ONG, C. S., SCHONAUER, S., VISHWANATHAN, S. V. N., SMOLA, A. J., AND KRIEGEL, H. P. 2005. Protein function prediction via graph kernels. *Bioinformatics 21,* Suppl 1 (Jun), i47–i56.

GÄRTNER, T., FLACH, P., AND WROBEL, S. 2003. On graph kernels: Hardness results and efficient alternatives. In *Proc. Annual Conf. Computational Learning Theory*, B. Schölkopf and M. K. Warmuth, Eds. Springer, 129–143.

HORVATH, T., GÄRTNER, T., AND WROBEL, S. 2004. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. 158–167.

INOKUCHI, A., WASHIO, T., AND MOTODA, H. 2003. Complete mining of frequent patterns from graphs: Mining graph data. *Machine Learning 50,* 3, 321–354.

KASHIMA, H., TSUDA, K., AND INOKUCHI, A. 2004. Kernels on graphs. In *Kernels and Bioinformatics*, K. Tsuda, B. Schölkopf, and J. Vert, Eds. MIT Press, Cambridge, MA, 155–170.

KASHTAN, N., ITZKOVITZ, S., MILO, R., AND ALON, U. 2004. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics 20,* 11, 1746–1758.

KURAMOCHI, M. AND KARYPIS, G. 2001. Frequent subgraph discovery. In *ICDM*. 313–320.

PRZULJ, N. 2007. Biological network comparison using graphlet degree distribution. *Bioinformatics 23,* 2 (Jan), e177–e183.

RALAIVOLA, L., SWAMIDASS, S. J., SAIGO, H., AND BALDI, P. 2005. Graph kernels for chemical informatics. *Neural Networks 18,* 8 (October), 1093–1110.

RAMON, J. AND GÄRTNER, T. 2003. Expressivity versus efficiency of graph kernels. Tech. rep., First International Workshop on Mining Graphs, Trees and Sequences (held with ECML/PKDD'03).

WEISSMAN, T., ORDENTLICH, E., SEROUSSI, G., VERDU, S., AND WEINBERGER, M. J. 2003. Inequalities for the l_1 deviation of the empirical distribution. Tech. Rep. HPL-2003-97(R.1), HP Labs, HP Laboratories, Palo Alto. June.

WERNICKE, S. 2005. A faster algorithm for detecting network motifs. In *WABI*, R. Casadio and G. Myers, Eds. Lecture Notes in Computer Science, vol. 3692. Springer, 165–177.

YAN, X. AND HAN, J. 2002. gspan: Graph-based substructure pattern mining. In *ICDM*. 721–724.