Large-Scale Graph Mining Using Backbone Refinement Classes

Andreas Maunz MAUNZA@FDM.UNI-FREIBURG.DE
Freiburg Center for Data Analysis and Modeling (FDM), Hermann-Herder-Str. 3, D-79104 Freiburg i. Breisgau, Germany

Christoph Helma HELMA@IN-SILICO.DE
in-silico Toxicology, Altkircherstr. 4, CH-4054 Basel, Switzerland

Stefan Kramer KRAMER@IN.TUM.DE
Institut für Informatik/I12, Technische Universität München, Boltzmannstr. 3, D-85748 Garching b. München, Germany

Keywords: Graph Theory, Graph Algorithms, Trees, Database Applications, Data Mining

Abstract

We present a new approach to large-scale graph mining based on so-called backbone refinement classes. The method efficiently mines tree-shaped subgraph descriptors under minimum frequency and significance constraints, using classes of fragments to reduce feature set size and running times, defined in terms of fragments sharing a common backbone. The method is able to optimize structural inter-feature entropy as opposed to occurrences, which is characteristic for open or closed fragment mining. In the experiments, the proposed method reduces feature set sizes by > 90 % and > 30 % compared to complete tree mining and open tree mining, respectively. Evaluation using crossvalidation runs shows that their classification accuracy is similar to the complete set of trees but significantly better than that of open trees. Compared to open or closed fragment mining, a large part of the search space can be pruned due to an improved statistical constraint (dynamic upper bound adjustment), which is confirmed in the experiments in lower running times compared to static upper bound pruning. Further analysis using large-scale datasets confirms that the novel descriptors render large training sets feasible which previously might have been intractable.

Presented at *ILP-MLG-SRL*, Leuven, Belgium, 2009.

A C++ implementation is available at `http://www.maunz.de/libfminer-doc/`.

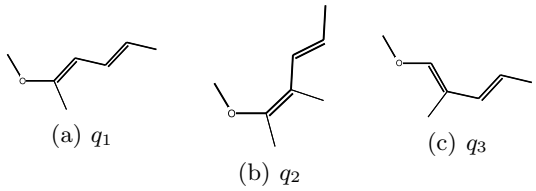
1. Introduction

Current methods for subgraph mining still suffer from scalability problems and, quite related, problems with excessively large solution sets. Most of the predominant approaches employ minimum frequency and possibly statistical correlation criteria such as χ^2 values (Nijssen & Kok, 2004; Yan & Han, 2002; Bringmann et al., 2006; Jahn & Kramer, 2005). In order to reduce the space of frequent and significant patterns, we use a natural property of tree-shaped subgraphs (the backbone) to represent classes, which renders the set usable for computational models even for large scale datasets (Maunz, Helma & Kramer 2009). In this way, we hope to obtain a more homogeneous and sparse distribution of patterns compared to occurrence summarization methods such as open or closed fragments.

1.1. Problem Formulation

Backbone Refinement Classes Undirected, labelled graphs are partially ordered via the refinement relation $\preceq$. Let $\mathcal{P}$ be the set of acyclic graphs with degree at most two (paths) and let $\mathcal{T}$ be the set of acyclic graphs (trees). Let $p = \{v_1, \ldots, v_m\} \in \mathcal{P}$ be a path, then its *sequence* is defined as the string $l(v_1)l((v1, v2)) \ldots l((v_{m-1}, v_m))l(v_m)$, obtained by concatenating node and edge labels along the path. Every tree $t \in \mathcal{T}$ has a *backbone* $b(t)$, which is defined as the longest path $p \subseteq t$ with the lowest sequence according to a lexicographic ordering (described

by Nijssen and Kok (Nijssen & Kok, 2004)). An (immediate) tree refinement of $t \in \mathcal{T}$ is an addition of an edge and a node to t s.t. the result t' is still acyclic, i.e. $t' \in \mathcal{T}$. A *backbone refinement* is a tree refinement that is backbone preserving, i.e. $b(t') = b(t)$.
We are considering the *Backbone Refinement Classes* of $b \in \mathcal{P}$, denoted by $\mathcal{BBRC}_b = \{\mathcal{BBRC}_{b_1}, \ldots, \mathcal{BBRC}_{b_n}\}$, where each $\mathcal{BBRC}_{b_i}$ is the set of trees that are backbone refinements of each other with respect to b, i.e. for all $r, r' \in \mathcal{BBRC}_{b_i}$ it holds that $b(r) = b(r')$ and $r \preceq r'$ or $r' \preceq r$. We denote the backbone refinement class relation by $\preceq_b$. Note that the classes are not disjoint for the same backbone (but they are across different ones). For example, in Figure 1, q_1 and q_3 are in different classes, but q_2 is in the respective classes of both q_1 and q_3. The set of all backbone refinement classes for a graph database R is called $\mathcal{BBRC}_R$. We also assume a binary target class labelling function for the graphs.



(a) q_1 (b) q_2 (c) q_3

Figure 1. Three example trees with the same backbone (bold). Its sequence is '`c:c:c-C=C-O-C`' (reflecting that the fragments include part of an aromatic ring). It also holds that $q_1 \preceq_b q_2$ and $q_3 \preceq_b q_2$, but neither $q_1 \preceq_b q_3$ nor $q_3 \preceq_b q_1$. Therefore, q_1 and q_3 are not in the same Backbone Refinement Class.

Backbone Refinement Class Representative Mining (BBRC Mining). Given a graph database R, a user-defined minimum support f and user-defined minimum χ^2 value u, for all $B \in \mathcal{BBRC}_R$, find the most general of the most significant $t \in B$ that is *frequent*, i.e. $supp(t, R) \geq f$, and *significant* with respect to occurrence in the target classes, i.e. $\chi^2(t, R) \geq u$.
The complexity of BBRC mining is upper-bounded by the complexity of regular tree mining (Nijssen & Kok, 2004). We will however show that our approach decreases running times significantly for practical applications.

2. Methods

We modified the graph miner Gaston (Nijssen & Kok, 2004) to support BBRC mining[1]. Two specific properties allow for an efficient implementation: Gaston first enumerates all path refinements, and only thereafter starts enumerating tree refinements growing from all paths, thereby prohibiting backbone changes while applying tree refinements recursively. Furthermore, Gaston uses a very efficient canonical representation for graphs. Specifically, no refinement is enumerated twice, e.g., q_2 in Figure 1.
For significance testing, the χ^2 distribution test (instead of the independence test commonly used in graph mining) was employed. It is possible to calculate an upper bound for the χ^2 values of refinements of a pattern (Morishita & Sese, 2000), which can be used for antimonotonic pruning. Using a static, user-defined upper bound threshold is referred to as *static upper bound pruning* (Bringmann et al., 2006). To speed up the search, we may increase this threshold (*dynamic upper bound adjustment*). For any frequent subtree q, let $\chi^2(q, R)$ and $\chi^2_u(q, R)$ denote the χ^2 value for q and χ^2 upper bound for refinements of q, respectively. Let $u_{max}(q) = max\{\chi^2(p, R) \mid p \preceq_b q\}$. Then, if $u_{max}(q) > u$, u may be increased to $u_{max}(q)$, since we only search for the maximum class element.

3. Experiments

3.1. Descriptor Computation and Predictivity

We evaluated four types of fragment descriptors:
1. *All Linear Fragments*
2. *Significant Trees*: all trees that have minimum frequency 6 and χ^2 significance of 95 %.
3. *Open Trees*: the most general representatives of all trees with the same occurrences from 2.
4. *BBRC Representatives*: the most significant representatives of the backbone refinement classes from 2.
It should be pointed out that 3. and 4. form a summarization of the features in 2.

3.2. Cross-validation

We used four chemical datasets obtained from the Carcinogenic Potency Database (CPDB)[2], version 08/04/29: Fragment types 1., 2., and 4. were calculated with the proposed approach. For the open trees (3.), we used the method by Bringmann *et al.* (Bringmann et al., 2006)[3]. Overall, fragment set sizes could be reduced by 94 %, 91 % and 31 % through BBRC representatives compared to linear subgraphs, significant trees and open fragments, re-

[1] We used version 1.1 (with embedding lists), see `http://www.liacs.nl/~snijssen/gaston/`.

[2] `http://potency.berkeley.edu/cpdb.html`

[3] The authors kindly provided us with a binary of their algorithm `sfgm`, pointing out that it may not be optimized for speed and uses a breadth first search technique known to be memory demanding.

spectively. Furthermore, dynamic upper bound adjustment was associated with a reduction in running time by 63.34 % and 60.92 % compared to using no statistical pruning and static upper bound pruning, respectively. The mined subgraphs were evaluated in a leave-one-out cross-validation using a nearest-neighbor approach. A paired t-test on the accuracy values revealed that BBRC representatives perform significantly better than open trees (mean accuracy $> 75\%$), while no significant difference between BBRC representatives and the complete set of trees was observed.

3.3. Large-Scale Analysis

We performed experiments on parts of the NCI Yeast Anticancer Drug Screen datasets[4] (April 2002 release). We used a subset of AC-one (stage 0) for cross-validation, composed of all actives and an equal number of inactives sampled randomly from the dataset (2*11,700 = 23,400 compounds). The second run, on AC-All (stage 1), used all actives and inactives (in total, 5,248+5,300 = 10,548 compounds). For AC-one (stage 0), the `sfgm` system computing open trees terminated with an error, while BBRC representatives took 4m52s. For AC-All (stage 1), open trees took $> 10h$, BBRC representatives took 1m13s. This time,

	AC-one (stage 0)	AC-all (stage 1)
Sign. Trees	1,190,763	291,729
Open Trees	?	216,206
Max. Trees	556,673	148,562
BBRC Repr.	31,450	14,381

Table 1. Feature counts for AC-One (stage 0) and AC-All (stage 1)

we investigated the set sizes of maximum patterns (the positive border as implied by minimum frequency and significance constraints (Al Hasan et al., 2007)) instead of linear fragments. Table 1 shows BBRC representatives had a very condensed representation of $\leq 5\%$; Indeed, BBRC representatives turned out to be the only practically useful feature type for cross-validation. With open trees, we obtained impractical prediction times of $> 60s$ and unacceptable RAM usage, whereas BBRC representatives gave a mean of $4.7s$ and $11.1s$, respectively, and accuracy of $> 70\%$.

4. Conclusion

Backbone Refinement Classes are a particularly useful class of subgraphs for mining databases of chemical compounds. Due to their formal properties, BBRCs can be mined efficiently and searched by existing graph mining systems like Gaston with only minor modifications. The overall method proves to be highly efficient compared to mining significant and open trees, dramatically reducing running time and the number of features mined. The experimental results revealed that the expressiveness of backbone refinement class representatives is significantly higher than that of open trees, because a lower number of features is associated with better accuracy, mainly due to higher specificity, reducing false alarms in classification tasks. The mined tree structures form a sparse and structurally diverse collection of patterns which cannot be guaranteed by occurrence summarization methods, such as open or closed subgraphs. In our experiments with large-scale datasets we showed that BBRCs can be computed within reasonable time and used effectively in simple predictive learning schemes.

[4] `http://dtp.nci.nih.gov/yacds/download.html`

References

Al Hasan, M., Chaoji, V., Salem, S., Besson, J., & Zaki, M. (2007). Origami: Mining Representative Orthogonal Graph Patterns. *ICDM 2007. Seventh IEEE International Conference on Data Mining*

Bringmann, B., Zimmermann, A., Raedt, L. D., & Nijssen, S. (2006). Don't be Afraid of Simpler Patterns. *Proceedings 10th PKDD*. Springer-Verlag.

Jahn, K., & Kramer, S. (2005). Optimizing gSpan for Molecular Datasets. *Proceedings of the Third International Workshop on Mining Graphs, Trees and Sequences (MGTS-2005)*.

Morishita, S., & Sese, J. (2000). Traversing Itemset Lattice with Statistical Metric Pruning. *Symposium on Principles of Database Systems*.

Nijssen, S., & Kok, J. N. (2004). A Quickstart in Frequent Structure Mining Can Make A Difference. *KDD '04: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM.

Yan, X., & Han, J. (2002). gSpan: Graph-Based Substructure Pattern Mining. *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*. Washington, DC, USA: IEEE Computer Society.

Maunz, A., Helma, C., & Kramer, S. (2009). Large-Scale Graph Mining Using Backbone Refinement Classes. *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM.