Probabilistic Frequent Subtree Kernels

Pascal Welke, Tamás Horváth, and Stefan Wrobel

LWA 2015

universitätbonn

Idea of Frequent Subgraph Based Kernels

Express the similarity of graphs using substructures

Idea of Frequent Subgraph Based Kernels

Express the similarity of graphs using substructures

Two graphs are similar, if share common subgraphs

Idea of Frequent Subgraph Based Kernels

Express the similarity of graphs using substructures

Two graphs are similar, if share common subgraphs

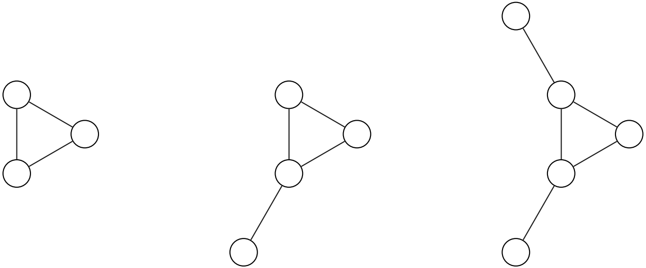
One way to do this: Consider *frequent subgraphs*

Idea of Frequent Subgraph Based Kernels

Express the similarity of graphs using substructures

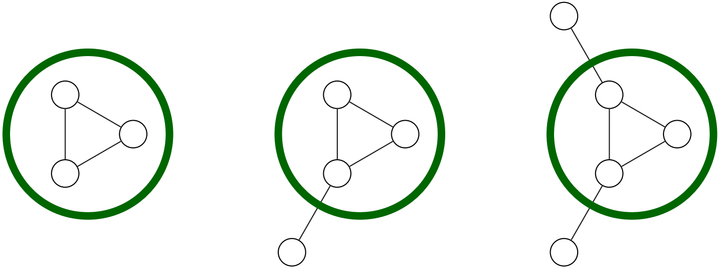
Two graphs are similar, if share common subgraphs

One way to do this: Consider *frequent subgraphs*



Idea of Frequent Subgraph Based Kernels

- Express the similarity of graphs using substructures
- Two graphs are similar, if share common subgraphs
- One way to do this: Consider *frequent subgraphs*



Motivation

Solving the Frequent Subgraph Problem is difficult and expensive

- Essentially, subgraph isomorphism has to be solved over and over again
- For general graphs, it is *not possible in output polynomial time* unless $P \neq NP$.

Motivation

Solving the Frequent Subgraph Problem is difficult and expensive

- Essentially, subgraph isomorphism has to be solved over and over again
- For general graphs, it is *not possible in output polynomial time* unless $P \neq NP$.

Often, frequent patterns are only a means to an end, not the required result

- Similarity
- Prediction of a target attribute
- ...

Motivation

Solving the Frequent Subgraph Problem is difficult and expensive

- Essentially, subgraph isomorphism has to be solved over and over again
- For general graphs, it is *not possible in output polynomial time* unless $P \neq NP$.

Often, frequent patterns are only a means to an end, not the required result

- Similarity
- Prediction of a target attribute
- ...

$\Rightarrow$ We can allow for a certain error, e.g.

- Compute an *incomplete* list of frequent patterns
- Compute an *incomplete* list of features for a given graph

Frequent Subtree Mining

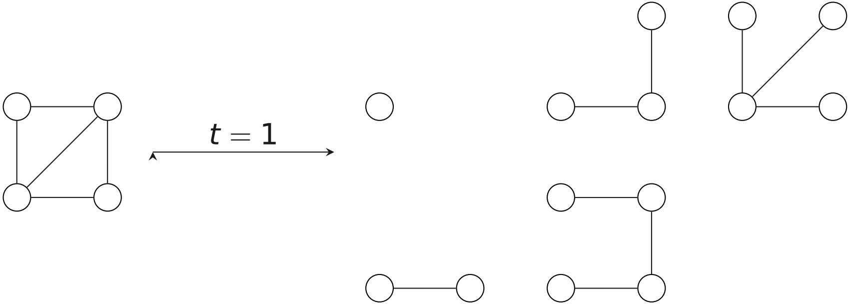
Given a set of graphs and a frequency threshold $t > 0$,

List the set of *frequent subtrees*

Frequent Subtree Mining

Given a set of graphs and a frequency threshold $t > 0$,

List the set of *frequent subtrees*



universitätbonn

Frequent Subtree Mining

...Frequent Subtree Mining is not possible in output polynomial time unless $P \neq NP$.

universitätbonn

Frequent Subtree Mining

...Frequent Subtree Mining is not possible in output polynomial time unless $P \neq NP$.

...but easy if the graphs in the database are trees themselves

Frequent Subtree Mining

...Frequent Subtree Mining is not possible in output polynomial time unless $P \neq NP$.

...but easy if the graphs in the database are trees themselves
$\Rightarrow$ just transform the graphs to trees

Probabilistic Subtree Kernels – Mining Algorithm

Input: A graph database $D \subseteq \mathcal{G}$, an integer $k > 0$,
and a threshold $t > 0$.

Output: A set of t-frequent subtrees of D.

$D' := \emptyset$
for all $G \in D$ **do**
    Sample k spanning trees of G uniformly at random
    Add the forest of those trees up to isomorphism to D'
List all t-frequent subgraphs in D'

So, Why Should We Do This?

Sampling a spanning tree: $\Theta(n^3)$

Isomorphism for trees: $O(n \log n)$

Subgraph isomorphism for trees: $O(n^{2.5} / \log n)$

universitätbonn

So, Why Should We Do This?

Sampling a spanning tree: $\Theta(n^3)$

Isomorphism for trees: $O(n \log n)$

Subgraph isomorphism for trees: $O(n^{2.5} / \log n)$

$\Rightarrow$ frequent subtrees can be efficiently mined in D'

universitätbonn

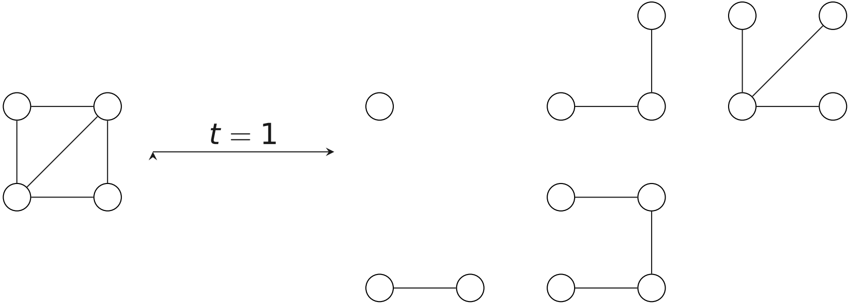
So, Why Should We Do This?

- Sampling a spanning tree: $\Theta(n^3)$
- Isomorphism for trees: $O(n \log n)$
- Subgraph isomorphism for trees: $O(n^{2.5} / \log n)$
- $\Rightarrow$ frequent subtrees can be efficiently mined in D'

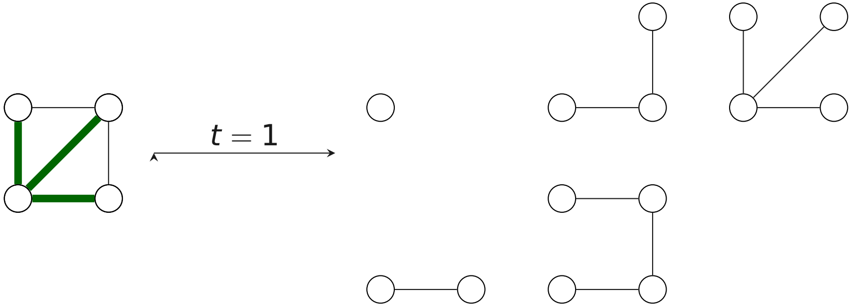
Theorem

Probabilistic Subtree Kernels can be computed efficiently for any database of graphs.

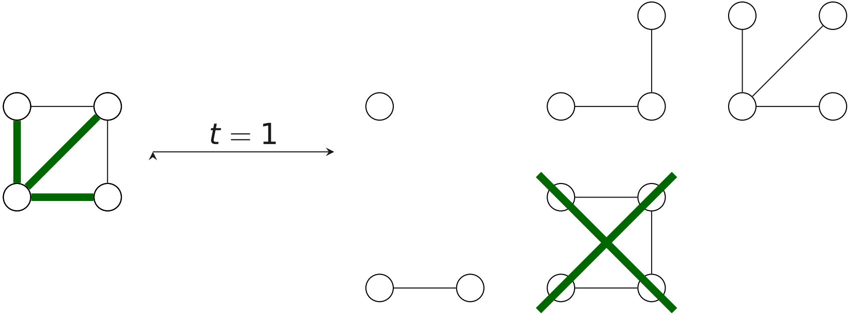
We Lose Some Patterns



We Lose Some Patterns



We Lose Some Patterns



Experiments

How much of the frequent patterns are lost?

How stable is the probabilistic pattern set?

Are probabilistic features useful for prediction?

How much faster is this, compared to frequent subtree mining?

Experiments

How much of the frequent patterns are lost?

How stable is the probabilistic pattern set?

Are probabilistic features useful for prediction?

How much faster is this, compared to frequent subtree mining?

Evaluation on two molecular datasets

- NCI-HIV
- ZINC-leadlike

Recall of the Probabilistic Subtree Pattern Space

		$k = 1$	$k = 2$	$k = 10$
NCI-HIV	$t = 5\%$	0.2013 ± 0.0120	0.3553 ± 0.0134	0.7832 ± 0.0085
	$t = 10\%$	0.2026 ± 0.0222	0.3445 ± 0.0142	0.7994 ± 0.0182
	$t = 20\%$	0.2445 ± 0.0138	0.3976 ± 0.0168	0.8338 ± 0.0140
ZINC	$t = 5\%$	0.3680 ± 0.0087	0.5670 ± 0.0165	0.9250 ± 0.0045
	$t = 10\%$	0.3277 ± 0.0189	0.5136 ± 0.0184	0.9249 ± 0.0118
	$t = 20\%$	0.3103 ± 0.0259	0.4899 ± 0.0305	0.9053 ± 0.0128

Table: Recall with standard deviation of the probabilistic tree patterns on the NCI-HIV and ZINC datasets for frequency thresholds 5%, 10%, and 20%

Stability of Probabilistic Subtree Patterns

Iteration	1	2	3	4	5	6	7	8	9	10
NCI-HIV	3920	20	5	10	14	7	2	6	7	2
ZINC	9898	18	17	11	10	22	7	7	9	1

Table: Repetitions of the experiment with $k = 1$ sampled trees. The numbers reported are the number of probabilistic patterns that were not in the union of all probabilistic patterns found up to the current iteration.

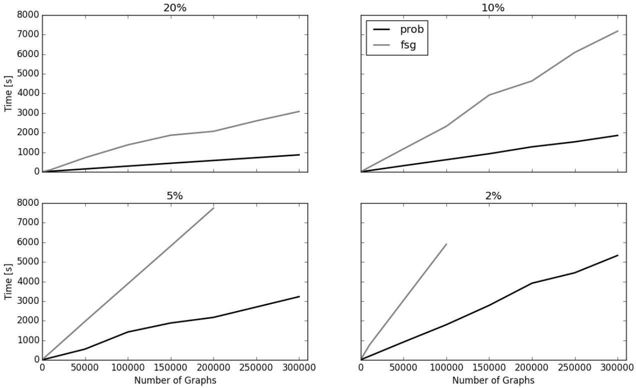
universitätbonn

Predictive Performance

t		AvsI	AMvsI	AvsMI
5%	Frequent Patterns	o.o.m	o.o.m	o.o.m
	Probabilistic, $k=1$	0.8927 ± 0.002	0.7235 ± 0.0023	0.8823 ± 0.0024
	Probabilistic, $k=2$	0.8994 ± 0.0012	0.7409 ± 0.0069	0.8909 ± 0.0074
10%	Frequent Patterns	0.9131 ± 0.0038	0.7529 ± 0.0024	0.9082 ± 0.0031
	Probabilistic, $k=1$	0.8853 ± 0.0081	0.7132 ± 0.0054	0.8745 ± 0.0118
	Probabilistic, $k=2$	0.8828 ± 0.0151	0.7109 ± 0.0021	0.8729 ± 0.0062
15%	Frequent Patterns	0.9134 ± 0.0026	0.7488 ± 0.0013	0.9093 ± 0.0031
	Probabilistic, $k=1$	0.8772 ± 0.0075	0.7062 ± 0.0055	0.8676 ± 0.0071
	Probabilistic, $k=2$	0.8762 ± 0.0071	0.7108 ± 0.0051	0.8676 ± 0.0042
20%	Frequent Patterns	0.9135 ± 0.0039	0.7424 ± 0.0026	0.9057 ± 0.0017
	Probabilistic, $k=1$	0.8675 ± 0.0076	0.6855 ± 0.0073	0.86 ± 0.0074
	Probabilistic, $k=2$	0.864 ± 0.01	0.6879 ± 0.0061	0.8579 ± 0.0074

Table: Average AUC values for the three learning problems on the NCI-HIV benchmark dataset for the frequent subgraph kernel and the probabilistic frequent subtree kernel for $k = 1, 2$ and for different frequency thresholds.

Speedup



universitätbonn

Conclusion and Future Work

Our method

- is efficient for arbitrary graph databases
- is faster and more memory efficient than frequent subgraph mining
- yields good classification results even for a single sampled spanning tree per graph

Conclusion and Future Work

Our method

- is efficient for arbitrary graph databases
- is faster and more memory efficient than frequent subgraph mining
- yields good classification results even for a single sampled spanning tree per graph

Future Work

- More Experiments (other graph classes)
- Specialized Tree Miners may improve the performance significantly
- Can we guarantee a certain error bound?

Resubmission of:

Pascal Welke, Tamás Horváth, and Stefan Wrobel:

Probabilistic Subtree Kernels

4th Workshop on New Frontiers in Mining Complex Patterns at ECML 2015