ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

Fast Data Mining with Sparse Chemical Graph Fingerprints by Estimating the Probability of Unique Patterns

Georg Hinselmann[1], Lars Rosenbaum[1], Andreas Jahn[1], and Andreas Zell[1]

1- Department for Computer Science, University of Tübingen,
Tübingen, Germany

Abstract. The aim of this work is to introduce a modification of chemical graphs fingerprints for data mining. The algorithm reduces the number of features by taking the probability of producing an unique feature at a specific search depth into account. We observed the probability of generating a non-unique feature depending on a search parameter (which leads to a power-law growths of features) and modeled it by a sigmoid function. This function was integrated into a fingerprinting routine to reduce the features according to their probability. The predictive performance was convincing with a considerable speedup for the training of a linear support vector machine for sparse instances.

1 Motivation

A decomposition of chemical graphs into nominal features is a convenient approach to encode and to compare chemical compounds. Recently, Benz et al. presented power-law functions which describe the growth of chemical graph features as a function of the search depth of a graph mining algorithm [1]. Furthermore, a hashed or lossy representation does not necessarily decrease the performance. The study by Shi et al. [2] showed that hashed kernels have a similar performance compared to the full kernels even if a significant amount of information is lost by applying hash functions. Similar techniques were applied to graph kernels [3, 4] where the influence of neighboring vertices is exponentially decreased.

In this work, we introduce a modification of the feature extraction routine which improves considerably the computation time for data mining on chemical graph encodings. The approach is based on the probability of observing an unique feature in a data set at a specific search depth. The idea is to reduce the impact of unique patterns among the exponentially growing number of patterns, which graph search algorithms generate with an increasing search depth.

We conducted experiments with a linear support vector machine on a toxicity data set comprising 6512 samples, considering computation time and predictive performance. The application of the modified fingerprinting algorithm led to a considerably decreased memory requirement and computation time in data mining and machine learning applications without having a significantly worse performance.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

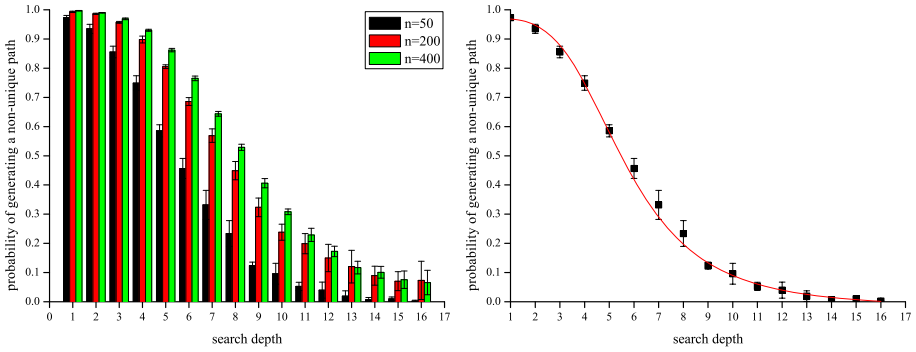
2 Methods

2.1 Probability Reduced Fingerprints for Chemical Graphs

In cheminformatics, an organic compound is usually encoded as a labeled and undirected graph composed of vertices (atoms) connected by edges (bonds). Such a graph has a restricted vertex degree (at most 4 for carbon atoms) and may contain contain cycles (e.g. aromatic rings). A good introduction to this topic can be found in [5].

In this section, we want to motivate and the concept of probability reduced fingerprints (PRF). The key idea is to reject a pattern according to a topological parameter which influences the growth of the set of possible patterns. By integrating a seeded probability-based filtering function into the fingerprinting routine it is possible to decrease the number of features while preserving the comparability of two encoded samples.

We employed depth-first search (DFS) fingerprints to motivate the PRF concept. The growth of linear path-based fingerprints shows a power-law behavior on chemical graphs with a complexity of $O(n \cdot \alpha^d)$, for a compound consisting of n atoms and a branching factor of α at a search depth of d. A DFS fingerprint, as described by Ralaivola et al. [6], is the set of all unique paths obtained by running an exhaustive DFS from every atom up to a defined depth. During the search, vertices could be traversed multiple times, with the exception of cycles. The resulting set of features consists of all valid paths encoded by the sequence of vertices and bonds traversed during the search.



(a) Probability of non-unique patterns for different subset sizes

(b) Sigmoid fit for $n = 50$

Fig. 1: (a) The sigmoid filter function is determined by observing the probabilities of the patterns for a topological search parameter. In (b), the observed values are fitted for $n = 50$.

The probability of generating a non-unique feature equals the total number of features minus the unique features divided by the total number of features. The probability of generating a non-unique feature could be fitted to a sigmoid function for different typing schemes and subset sizes for compounds from ChemDB

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

[7] with a nearly perfect correlation and negligible error using R [1]. The probabilities can be fitted by a sigmoid function of the form $f(d) = A_2 + \frac{A_1 - A_2}{1+(\frac{d}{x_0})^p}$ where d equals the search depth. A_2, A_1, x_0, and p have to be chosen such that the correlation between the observed frequency of non-unique patterns and the estimated frequency is maximized and the squared error is minimized. The principle is presented in Figure 1. We implemented all fingerprinting algorithms utilizing the Chemistry Development Kit [8].

$f(d)$ is used to predict a probability threshold for a pattern at a search depth d. The generation of a PRF is shown in Algorithm 1 where the complete spectrum of features is stored in a prefix search tree (trie). In line 3, the method extracts the fingerprint features up to a predefined depth. The loop in line 5 iterates over all generated patterns. For each feature, the sigmoid function variable d equals the depth of the current *feature*. A_1, A_2, x_0 and p are predefined for an optimal fit of the observed probability of a non-unique feature at search depth d. This function returns the probability *acceptThreshold* of generating a non-unique path of a feature at depth d. If this value is larger than *acceptChance*, which is a random number $r \in [0, 1] \in \mathbb{R}$ (line 10-12), the feature is included in the final encoding. The features are extracted deterministically because the generation of r depends on the id of a *feature*. If *acceptChance* is smaller than *acceptThreshold*, $feature$.string is included in the fingerprint.

A basic requirement for the application as a fingerprinting routine is that the results must be comparable. In other words, if two structures have common patterns they must be covered by their fingerprints in the same way. This is realized in the algorithm because the random number generator is seeded with the value of the feature id $\in \mathbb{N}$ (Algorithm 1, line 8). Each graph feature has a numerical id ($feature$.id) computed by the CRC algorithm on its canonical string representation ($feature$.string) so that two features are equal if their numerical ids are equal. Thus, if a pattern passes the threshold it is ensured that it occurs in every fingerprint.

2.2 Experimental Setup

A large balanced binary classification dataset comprising 6512 toxic and non-toxic samples [9] was used as a real-world benchmark set. The learning task was to predict whether a compound shows toxic effects in the Ames test for mutagenicity.

We used LIBLINEAR [10] for the machine learning experiments. The machine learning performance was determined with a nested 10-fold cross-validation using the optimal parameters estimated by an inner 2-fold cross-validation. In the inner loop, a model selection determined the best C parameter with $\log_2 C \in \{-9, -8, ..., 1, 2\}$ (the overall penalty parameter for a misclassification) for the support vector machine. The cross-validation was conducted 10 times to reduce the impact of the initial seed value for generating the cross-validation folds. The significance of the AUC ROC performance was determined by the

[1] http://www.r-project.org/

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

```
1  Trie getPRF(Molecule mol, maxDepth) {
2    Trie trie ← new Trie()
3    List<Feature> features ← getFingerprint(mol, maxDepth)
4
5    for (Feature feature ∈ features) {
6      d ← feature.depth
7      acceptTreshold ← A2 + (A1−A2) / (1+(d/x0)^p)
8      seed ← feature.id;
9      acceptChance ← getRandomNumber(seed)
10     if (acceptChance < acceptTreshold) {
11       trie.insertAsBranch(feature.string)
12     }
13   }
14   return trie
15 }
```

Algorithm 1: Generation of a PRF for a chemical graph *mol*. The probability of storing the feature depends on its depth. The id of a feature is a unique numerical identifier computed by the CRC algorithm on a canonical string representation. It is the seed for the pseudo random number generator.

corrected resampled t-test [11] on the 100 AUC ROC values provided by the 10×10 cross-validation results.

The complexity of the linear support vector machine LIBLINEAR is $O(n)$ for n samples (considering an ϵ-accurate solution, where ϵ is a constant) and a gradient descent with constant complexity. However, the update step of the gradient descent depends on the average number of features in a vector. Consequently, an increased sparseness of the fingerprints leads to faster training times. For further details on the computational complexity of the dual coordinate descent method of LIBLINEAR, please refer to the studies of Hsieh et al. [12] and Bottou and Bousquet [13]. To determine the improvement with respect to the computation time, we conducted the model selection described above using 10-fold cross-validation and compared the averaged values.

The sigmoid function used for the experiments was based on averaged observations on a subset of 50 compounds ($n = 50$, values averaged over 10 repetitions, fit with $R^2 > 0.999$) randomly sampled from ChemDB. The identifier id was computed as the 32-bit integer hash code of the string representation of a path.

3 Results

The AUC ROC performance of the DFS-PRF was comparable to the full DFS fingerprint. We investigated the computation time in the range from depth 3 to 8. The fastest computation time was determined at a depth of 3, which was 73% of the computation time of the DFS-PRF at depth 8. For higher values of d, the computation time using the DFS-PRF encoding was superior. The predictive performance of the DFS fingerprints at depth 3 to 5 was significantly worse compared to the DFS-PRF with $d = 8$. At a search depth of 6, the performance

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

did not differ significantly, but the computation time for the model selection was a factor of 3.36 slower compared to DFS-PRF. The computation times at depth 7 and 8 were 4.56 and 6.56 slower than the DFS-PRF at depth 8, but did not differ significantly.

Table 1: Performance and computation time of the standard DFS fingerprints in comparison to the PRF version ($d = 8$) using LIBLINEAR. Significantly best AUC ROC performance in bold font according to the corrected resampled t-test.

Depth	Type	AUC[a]	CT[b]	Ratio CT
8	PRF	$\mathbf{0.87421 \pm 0.01478}$	147	1.00
3	Full	0.83520 ± 0.01546	**108**	**0.73**
4	Full	0.85293 ± 0.01529	203	1.38
5	Full	0.85969 ± 0.01543	374	2.36
6	Full	$\mathbf{0.87567 \pm 0.01475}$	494	3.36
7	Full	$\mathbf{0.87705 \pm 0.01462}$	683	4.65
8	Full	$\mathbf{0.87841 \pm 0.01451}$	965	6.56

[a] AUC ROC performance (10×10 nested cross-validation with model selection)
[b] Model selection in seconds (single 10-fold cross-validation with model selection, two AMD Opteron 280 CPUs)

4 Discussion and Conclusion

We investigated a heuristic which reduces deterministically the features of chemical graphs graph based fingerprinting routines by analyzing the frequencies of patterns of a specific search depth on a large chemical database. As a consequence, the computation time for data mining algorithms can be significantly decreased. The principle is general and works in combination with similar fingerprinting algorithms which exhibit a power-law growth, e.g. extended connectivity fingerprints [14].

The approach has similarities to graph kernels. The marginalized graph kernel weights the features exponentially decreasing with their length [3] to compensate the effect of chance occurrences. Similar concepts were applied with the optimal assignment kernel where the influence of neighboring vertices are also exponentially decreased.

The increased sparseness of the fingerprints leads to a faster computation time of the similarity in any instance-based machine learning approach (e.g. by comparing the key sets of hash maps). From the results we can see that the computation time was improved up to factor 6.56 for the model selection of LIBLINEAR which is induced by the increased sparseness (only 11% of the non-zero bits remained). The improvement by using the DFS-PRF stems from the computation time of gradient descent, where the update step depends on the average number of non-zero bits in a sample.

We did not observe a significantly decreased predictive performance compared to the full DFS fingerprints on the toxicity classification benchmark. How-

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

ever, we found that the computation time of the linear support vector machine and the memory requirements for storing the fingerprints could be improved for the depth-first search encoding as a case in point. For a search depth of 3, the full DFS fingerprint had a faster computation time, but a significantly worse predictive performance. For search depths 4 and 5 the predictive performance of the full fingerprint was significantly worse with a higher computation time. To sum sup, we think that this approach is an alternative to the comparison of the full set of features obtained with a fixed search depth.

References

[1] Ryan W. Benz, S. Joshua Swamidass, and Pierre Baldi. Discovery of power-laws in chemical space. *J. Chem. Inf. Model.*, 48:1138–1151, 2008.

[2] Qinfeng Shi, James Petterson, Gideon Dror, John Langford, Alex Smola, and S. V. N. Vishwanathan. Hash kernels for structured data. *J. Mach. Learn. Res.*, 10:2615–2637, 2009.

[3] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized Kernels Between Labeled Graphs. In Tom Fawcett and Nina Mishra, editors, *20th International Conference on Machine Learning*, Proceedings of the 20th International Conference on Machine Learning, pages 321–328. AAAI Press, 2003.

[4] Holger Fröhlich, Jörg K. Wegner, Florian Sieker, and Andreas Zell. Kernel Functions for Attributed Molecular Graphs - A New Similarity-Based Approach to ADME Prediction in Classification and Regression. *QSAR Comb. Sci.*, 25:317–326, 2006.

[5] Diane J. Cook and Lawrence B. Holder. *Mining Graph Data*. John Wiley & Sons, 2006.

[6] Liva Ralaivola, Sanjay J. Swamidass, Hiroto Saigo, and Pierre Baldi. Graph Kernels for Chemical Informatics. *Neural Networks*, 18(8):1093–1110, 2005.

[7] Jonathan Chen, S. Joshua Swamidass, Yimeng Dou, Jocelyne Bruand, and Pierre Baldi. ChemDB: a public database of small molecules and related chemoinformatics resources. *Bioinformatics*, 21(22):4133–4139, 2005.

[8] Christoph Steinbeck, Yongquan Han, Stefan Kuhn, Oliver Horlacher, Edgar Luttmann, and Egon Willighagen. The Chemistry Development Kit (CDK): an open-source Java library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.*, 43(2):493–500, 2003.

[9] Katja Hansen, Sebastian Mika, Timon Schroeter, Andreas Sutter, Antonius ter Laak, Thomas Steger-Hartmann, Nikolaus Heinrich, and Klaus-Robert Müller. Benchmark data set for in silico prediction of ames mutagenicity. *J. Chem. Inf. Model.*, 49 (9):2077–2081, 2009.

[10] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.

[11] Remco R. Bouckaert and Eibe Frank. Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms. In Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang, editors, *Advances in Knowledge Discovery and Data Mining - Proceedings of 8th Pacific-Asia Conference, PAKDD 2004*, volume 3056, pages 3–12. Springer, May 2004.

[12] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 408–415, New York, NY, USA, 2008. ACM.

[13] Léon Bottou and Olivier Bousquet. Learning using large datasets. In *Mining Massive Data Sets for Security*, NATO ASI Workshop Series. IOS Press, Amsterdam, 2008.

[14] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, May 2010.