Internal Report

Deliverable D1.3C:

Mechanism for handling large data sets in the rank of Big Data

Jozef Stefan Institute

Version 1 FINAL

Abstract: Semantic data mining (SDM) uses annotated data and interconnected background knowledge to generate rules that are easily interpreted by the end user. However, the complexity of SDM algorithms is high, resulting in long running times even when applied to relatively small data sets. On the other hand, network analysis algorithms are among the most scalable data mining algorithms. This paper proposes an effective SDM approach that combines semantic data mining and network analysis. The proposed approach uses network analysis to extract the most relevant part of the interconnected background knowledge, and then applies a semantic data mining algorithm on the pruned background knowledge. The application on acute lymphoblastic leukemia data set demonstrates that the approach is well motivated, is more efficient and results in rules that are comparable or better than the rules obtained by applying the incorporated SDM algorithm without network reduction in data preprocessing.

Document administrative information	
Project acronym:	HinLife
Project number:	J7-7303
Deliverable number:	D1.3
Deliverable full title:	Mechanism for handling large data sets in the rank of Big Data
Document identifier:	HinLife D1.3c v1.FINAL
Lead partner short name:	IJS
Report version:	Version 1, FINAL
Report preparation date:	13/02/2018
Lead author:	Jan Kralj
Co-authors:	Anže Vavpetič, Michel Dumontier, Nada Lavrač
Status:	Final

Network ranking assisted semantic data mining

Jan Kralj[1,2], Anže Vavpetič[1,2], Michel Dumontier[4] and Nada Lavrač[1,2,3]

[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] Jožef Stefan Int. Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia
[3] University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia
[4] Stanford Center for Biomedical Informatics Research, Stanford University
{jan.kralj,anze.vavpetic,nada.lavrac}@ijs.si
michel.dumontier@stanford.edu

Abstract. Semantic data mining (SDM) uses annotated data and interconnected background knowledge to generate rules that are easily interpreted by the end user. However, the complexity of SDM algorithms is high, resulting in long running times even when applied to relatively small data sets. On the other hand, network analysis algorithms are among the most scalable data mining algorithms. This paper proposes an effective SDM approach that combines semantic data mining and network analysis. The proposed approach uses network analysis to extract the most relevant part of the interconnected background knowledge, and then applies a semantic data mining algorithm on the pruned background knowledge. The application on acute lymphoblastic leukemia data set demonstrates that the approach is well motivated, is more efficient and results in rules that are comparable or better than the rules obtained by applying the incorporated SDM algorithm without network reduction in data preprocessing.

1 Introduction

Research into semantic data mining has so far focused on algorithms that produce complex, high quality rules that describe the data they are applied to. The complexity of the outputs of SDM algorithms results in a severe performance bottleneck because the search space in which the algorithms look for rules is huge, and grows exponentially with the size of the background knowledge. On the other hand, network analysis is a research field with an abundance of research done to increase the performance and scalability of algorithms, resulting in algorithms that are capable of analyzing huge networks. The sizes of background knowledge data used in SDM approaches are usually several orders of magnitude smaller than the problems typically encountered in network analysis. While, for example, our SDM algorithm Hedwig in [27] used a set of 337 examples and a background knowledge containing a total of 21,062 nodes, network analysis algorithms are capable of handling much larger data sets, composed of hundreds of millions of nodes.

Despite the large difference in the sizes of data analyzed by network analysis compared to SDM, the two research fields are not fundamentally incompatible.

In the most basic sense, both fields are interested in the question "Which part of the network structure is most important to my current interests?". This paper presents a method that is capable of utilizing aspects of network analysis, specifically the PageRank algorithm, along with the Hedwig semantic data mining algorithm, to produce high quality rules by only searching a fraction of the entire background knowledge space.

This paper is structured as follows. The related work is presented in Section 2. Section 3 presents Hedwig, the semantic data mining algorithm we used in the construction of our new algorithm. Section 4 presents how PageRank, a network ranking method, can be used to decrease the size of the background knowledge used by the Hedwig algorithm. Section 5 presents the setup and results of the experiments run with a method that merges both PageRank and Hedwig. Section 6 concludes the paper and describes further work that can be done to extend this research.

2 Related work

The related work for this paper consists of research done in several different fields of research.

Semantic pattern mining. Rule learning, which was initially focused on building predictive models formed of sets of classification rules, has recently shifted its focus to descriptive pattern mining. Well-known pattern mining techniques in the literature are based on association rule learning [2, 21]. While the initial studies in association rule mining have focused on finding interesting patterns from large data sets in an unsupervised setting, association rules have been used also in a supervised setting, to learn pattern descriptions from class-labeled data [17]. Building on top of the research in classification and association rule learning, subgroup discovery has emerged as a popular data mining methodology for finding patterns in class-labeled data, aiming to find interesting patterns as sets of individual rules that best describe the target variable [14, 29].

Subgroup descriptions in the form of propositional rules are suitable descriptions of groups of instances. However, given the abundance of taxonomies and ontologies that are readily available, these can also be used to provide higher-level descriptors and explanations of discovered subgroups. Especially in the domain of systems biology the GO ontology [5], KEGG orthology [19] and Entrez gene–gene interaction data [18] are good examples of structured domain knowledge that can be used as additional higher-level descriptors in the induced rules.

The challenge of incorporating the domain ontologies in data mining was addressed in recent research on semantic data mining (SDM) [16, 26, 28]. In [28] an engineering ontology of Computer-Aided Design (CAD) elements and structures was used as background knowledge to extract frequent product design patterns in CAD repositories and discovering predictive rules from CAD data. Using ontologies, algorithm Fr–ONT for mining frequent concepts expressed in $\mathcal{EL}^{++}$ DL was introduced in [16]. In [26] we described and evaluated the SDM toolkit that includes two semantic data mining systems: SDM-SEGS and SDM-Aleph. SDM-SEGS is an extension of earlier domain-specific algorithm SEGS [24] which allows for semantic subgroup discovery in gene expression data. SEGS constructs gene sets as combinations of GO ontology [5] terms, KEGG orthology [19] terms, and terms describing gene–gene interactions obtained from the Entrez database [18]. SDM-SEGS extends and generalizes this approach by allowing the user to input any set of ontologies in the OWL ontology specification language and an empirical data collection which is annotated by domain ontology terms. SDM-SEGS employs ontologies to constrain and guide the top-down search of a hierarchically structured space of induced hypotheses. SDM-Aleph, which is built using the inductive logic programming system Aleph [23] does not have the limitations of SDM-SEGS, imposed by the domain-specific algorithm SEGS. Additionally, SDM-Aleph can accept any number of OWL ontologies as background knowledge which is then used in the learning process.

Network node ranking. The task of network node ranking in an information network provides means for assigning a *score* (or *rank*) to each node in the network, thus ranking the nodes from the highest to the lowest ranked node. The most famous ranking algorithm is the PageRank algorithm [20] used by the Google search engine, however several other network raking methods have been proposed such as a weighted version of the PageRank method called Weighted PageRank [30], as well as the related Hubs and Authorities method [13]. Another method to rank nodes in the network is to use centrality measures, for example using Freeman's network centrality [8], betweenness centrality [7], closeness centrality [4] and the Katz centrality measure [12].

Previous work on acute lymphoblastic leukemia. In the analysis we explored the acute lymphoblastic leukemia (ALL) data set used in a previous publication. We followed the steps used in [22] to obtain a set of $1,000$ enriched genes from a set of $10,000$ genes. The enriched genes were annotated by concepts from the Gene Ontology [3] which formed the background knowledge for our experiments. The original publication analyzing the ALL data set compared the performance of the DAVID [10] algorithm and the SegMine algorithm. In this work, however, we used the same data set to measure how we can improve the performance of the Hedwig algorithm, the algorithm which was already shown to perform well in a biological setting. The goal of this work is to examine whether network node ranking can decrease the runtime and improve the performance of the Hedwig algorithm.

3 Semantic data mining

This section describes the recently developed semantic subgroup discovery system Hedwig [27]. Compared to standard subgroup discovery algorithms, Hedwig uses domain ontologies to structure the search space and formulate generalized hypotheses [27]. Existing semantic subgroup discovery algorithms are either specialized for a specific domain [25] or adapted from systems that do not take into the account the hierarchical structure of background knowledge [26]. Hedwig overcomes these limitations as it is designed to be a general purpose semantic subgroup discovery system.

In addition to a financial use case [27], Hedwig was already shown to perform well in a biological setting, namely analyzing DNA aberration data for various cancer types [1], where it was part of a three-step methodology, together with mixture models and banded matrices. In the analysis, additional background knowledge was used in the form of several ontologies: hierarchical structure of multiresolution data, chromosomal location of fragile sites, virus integration sites, cancer genes, and amplification hotspots, obtained from various sources.

Semantic subgroup discovery, as addressed by the Hedwig system, results in relational descriptive rules. Hedwig uses ontologies as background knowledge and training examples in the form of Resource Description Framework (RDF) triples. The semantic data mining task addressed in this work takes as inputs the empirical data in the form of a set of training examples expressed as RDF triples, domain knowledge in the form of ontologies, and an object-to-ontology mapping which associates each object from the RDF triplets with appropriate ontological concepts, and finds a hypothesis (a predictive model or a set of descriptive patterns), expressed by domain ontology terms, explaining the given empirical data.

Subgroup describing rules are first-order logical expressions. Consider the following rule, used to explain the format of induced subgroup describing rules, such as, for example: $\boxed{\texttt{Class(X)} \leftarrow \texttt{C1(X), R(X,Y), C2(Y)}}$ with True Positives (TP)=80 and False Positives (FP)=20. Variables `X`, `Y` represent sets of input instances, R is a binary relation between the examples and C_1, C_2 are ontolog-

```
Input  : Input examples $E$, background knowledge $B$, target class value $c$,
         beam size $k$, $p$-value threshold $\alpha$
Output: Set of rules
rules ← [default_rule($E$, $c$, $B$)]
while improvement(rules) do
    // Add specializations of each rule to the beam
    for rule ∈ rules do
        extend(rules, specialize(rule, $B$))
    end
    rules ← best(rules, $k$) // Select the top $k$ rules
end
rules ← validate(rules, $\alpha$) // Significance testing
return rules
```

Algorithm 1: Hedwig's `induce`(E, B, c, k, α) procedure.

ical concepts. This rule is interpreted as follows. If an example X is annotated with concept C_1, and is related with an example Y via R, and Y is annotated with concept C_2, then the conclusion $Class(X)$ holds. This rule condition is true for 100 input instances ($TP + FP$, also called *coverage*), 80 of which are of the target class (TP, also called *support*).

We implemented the algorithm described in Figures 1 and 2 to search for interesting subgroups. The Hedwig system, which implements this algorithm, supports ontologies and examples to be loaded as a collection of RDF triples (a graph). The system automatically parses the RDF graph for the `subClassOf` hierarchy, as well as any other user-defined binary relations. Hedwig alsodefines a

```
Input  : Rule to specialize rule, background knowledge $B$
Output: Set of specializations of rule

specializations ← []
// Predicates that can be specialized
eligible_preds ← eligible(predicates(rule))
for predicate ∈ eligible_preds do
    // Specialize by traversing the subClassOf hierarchy
    for subclass ∈ subclasses(predicate, B) do
        new_rule ← swap(rule, predicate, subclass)
        if can_specialize(new_rule) then
            append(specializations, new_rule)
        end
    end
    // Specialize by negating
    new_rule ← negate(rule, predicate)
    if can_specialize(new_rule) then
        append(specializations, new_rule)
    end
end
if rule ≠ default_rule then
    // Specialize by adding a new unary predicate
    new_predicate ← next_non_ancestor(eligible_preds)
    new_rule ← append(rule, new_predicate)
    if can_specialize(new_rule) and non_redundant(new_rule) then
        append(specializations, new_rule)
    end
end
if is_unary(last(predicates(rule))) then
    // Specialize by adding new binary predicates
    extend(specializations, specialize_binary(new_rule))
end
return specializations
```

Algorithm 2: Hedwig's `specialize`(*rule*, *B*) procedure.

namespace of classes and relations for specifying the training examples to which the input must adhere.

The algorithm uses beam search, where the beam contains the best N rules found so far. It starts with the default rule which covers all the input examples. In every iteration of the search, each rule from the beam is specialized via one of the four operations: (1) Replace predicate of a rule with a predicate that is a sub-class of the previous one, (2) negate predicate of a rule, (3) append a new unary predicate to the rule, or (4) append a new binary predicate, introducing a new existentially quantified variable.[5]

Rule induction via specializations is a well-established way of inducing rules, since every specialization either maintains or reduces the current number of covered examples. A rule will not be specialized once its coverage is zero or falls below some predetermined threshold. When adding a new conjunction, we check that if the extended rule does not improve the probability of the conclusion (we use the redundancy coefficient, as in [11]), then it is not added to the pool of specializations. After the specialization step is applied to each rule in the beam, we select new set of the best scoring N rules. If no improvement is made to the collection of rules, the search is stopped. In principle, our procedure supports any rule scoring function. Numerous rule scoring functions (for discrete targets) are available: χ^2, precision, WRAcc [15], leverage and lift. The latter is the default choice and was also used in our experiments. After the induction phase, the significance of the findings is tested using the Fisher's exact test [6]. To cope with the multiple-hypothesis testing problem, we use Holm-Bonferroni [9] direct adjustment method with $\alpha = 0.05$.

4 Using network node ranking to decrease background knowledge size

We used network ranking, in particular the personalized PageRank [20] algorithm, to asses the importance of each node in the background knowledge. The personalized PageRank of a set of nodes S (P-PR$_S$) in a network is defined as the stationary distribution of the position of a random walker who starts the walk in a randomly chosen member of S and then at each step either selects one of the outgoing connections or teleports back to a randomly selected member of S. The probability (denoted p) of continuing the walk is a parameter of the personalized PageRank algorithm and is usually set to 0.85.

The fundamental idea in our algorithm is that the PageRank method can be used to assess the relevance of a given background knowledge node for a particular experiment, and that Hedwig and other SDM algorithms are more likely to use highly relevant nodes when constructing rules. Therefore, if we allow the SDM algorithms to construct rules using only the most important nodes, the quality of the rules should increase. At the same time, because the background

[5] The new variable needs to be 'consumed' by a literal to be added as a conjunction to this clause in the next step of rule refinement.

knowledge is decreased in size, the SDM algorithm we use to construct the rules will have to search through a significantly reduced space of possible rules and should therefore take much less time to conclude.

The algorithm (described in pseudo-code as Algorithm 3) consists of three steps. In the first step, we construct a network which we will use to assess the importance of background knowledge nodes. We begin with a background knowledge represented as a graph $G = (V, E)$, where V is the set of nodes and E a set of edges, and a data set S we wish to analyze. The data set S is split into a set of positive examples S_+ and a set of negative examples S_-, i.e. $S_+ \cup S_- = S, S_+ \cap S_- = \emptyset$. Each example $s \in S$ is *annotated* with some set of background knowledge nodes.

From G and S, we construct a new network $G' = (V', E')$ by taking the original network G and adding all positive examples to the set of background knowledge nodes (in other words, we set $V' = V \cup S_+$), connecting them to background knowledge nodes through the annotations ($E' = E \cup \{(e, a) \in S \times V | \text{annotation } a \text{ annotates example e}\}$)

In the second step of the algorithm, we decrease the size of the background knowledge network G by removing less important nodes. We calculate the personalized PageRank values of the nodes in the expanded network G', setting the starting nodes of for the iteration to all nodes in S This allows the pagerank values to flow from the data set examples to the nodes that annotate them. The background knowledge network is then decreased by removing from it all but the top t percent of nodes, where t is the selected threshold and a parameter of our algorithm. We thus create a new background knowledge network G_s whose nodes consist of a subset of V and whose edges are induced by the edges in E.

In the final step, we use the Hedwig algorithm to construct rules, consisting of conjuncts of nodes in G_s, that best describe the set S_+.

```
Data: Background knowledge network $G$ and set of examples $S$ annotated
      with nodes from $G$
Result: Rules describing the positive examples
Parameters: PageRank restart probability $p \in [0, 1]$, Cutoff percendate
$c \in [0, 1]$ Set $G' = \{g \in G : \exists e \in S : e \text{ is annotated by } g\}$;
Calculate $r = PPR_G$;
for node $g \in G$ do
    if $|\{g' \in G : r(g') > r(g)\}| > c \cdot |G|$ then
        remove node $g$ from $G$.
    end
end
Run semantic data mining on $S$ using the pruned $G$ as background
knowledge return Rules, discovered by the SDM algorithm on the pruned
background knowledge
```

Algorithm 3: The proposed network ranking supported semantic data mining algorithm

5 Experiments

The experimental setup of our work consisted of two steps. In the first set of experiments, we ran the Hedwig algorithm on the data set to determine the baseline performance of the algorithm. We ran the algorithm with several settings of depth and beam width. The results of this round of experiments are shown in Table 1 and show that consistently, the gene ontology nodes that appear in the discovered rules have a PageRank value that is highly above normal.

The rules, discovered in this round of experiments, are also biologically significant. In all three settings when the search beam for the algorithm was set to 1, the only significant rule discovered was the gene ontology term GO:3674, a term denoting molecular function. This is a very broad term which offers little insight and shows that a larger search beam is necessary in order for Hedwig to make significant discoveries. The most interesting results are the results uncovered when the beam size is set to 10 and the support is set to 0.01. When the depth is set to 1 , the most important term GO:50851 (antigen receptor-mediated signaling pathway) is interesting as it relates to the immune system related cell type. When searching with a depth of 10, we discovered a conjunct of four terms: immune system process (GO:2376), immune response-activating cell surface receptor signaling pathway, (GO:2429), plasma membrane (GO:5886) and binding (GO:5488). This conjunct is interesting as it begins to provide some additional insight of the action (binding), effect (immune response signalling pathway), and location (plasma membrane).

In the second round of experiments, we decreased the size of the background data by removing low ranking nodes. We calculated the PageRank value of the GO nodes in two ways: in the first, we viewed `is_a` relations as directed edges pointing from the more specific GO term to the more general term. In the second, we viewed the relations as undirected edges. We ran the Hedwig algorithm on a gene ontology backgorund data set containing only the 5%, 10%, 20% and 50% of nodes with the highest PageRank value. We also only focused on setting the size

Rule [ranking]	Beam	Depth	Support	Lift
GO:3674[0.0046]	1	1	0.01	1
GO:3674, [0.0046]	1	10	0.01	1
GO:50851, [0.7368]	10	1	0.01	2.687
GO:2376 [0.16047], GO:2429[0.6880], GO:5886 [0.0790], GO:5488[0.0070]	10	10	0.01	3.42
GO:3674, [0.0046]	1	10	0.1	1
GO:2376, [0.1604]	10	1	0.1	1.292
GO:2376 [0.1604], GO:5488 [0.0070] GO:48518[0.4277]	10	10	0.1	1.414
GO:2376 [0.1604],GO:5488 [0.0070] GO:48518 [0.4277]	10	10	0.1	1.414

Table 1. Best rules discovered by the Hedwig algorithm for the ALL data set. Each row presents the conjuncts (Gene Ontology terms) of the top ranking rule. The number in parentheses is the percentage of GO terms with a PageRank higher than the term in the rule. The numbers are remarkably low, showing that Hedwig consistently constructs rules with the top 1% GO terms as ranked by the PageRank algorithm.

Cutoff	Rules	Beam	Depth	Support	Lift
0.05	GO:50851	10	1	0.01	2.687
0.1	GO:50851	10	1	0.01	2.687
0.2	GO:50851	10	1	0.01	2.687
0.5	GO:50851	10	1	0.01	2.687
1	GO:50851	10	1	0.01	2.687
0.05	GO:2376, GO:2694, GO:34110	10	10	0.01	3.235
0.1	GO:2376, GO:2694, GO:44459	10	10	0.01	4.09
0.2	GO:3824, GO:44283, GO:44444	10	10	0.01	4.257
0.5	GO:2376, GO:2429 GO:5886 GO:5488,	10	10	0.01	3.42
1	GO:2376, GO:2429, GO:5886, GO:5488	10	10	0.01	3.42
0.05	GO:43234	10	1	0.1	1.521
0.1	GO:43234	10	1	0.1	1.543
0.2	GO:43234	10	1	0.1	1.506
0.5	GO:2376	10	1	0.1	1.292
1	GO:2376	10	1	0.1	1.292
0.05	GO:43234, GO:44464	10	10	0.1	1.537
0.1	GO:65007, GO:43234, GO:44424	10	10	0.1	1.655
0.2	GO:16020, GO:43234, GO:8150	10	10	0.1	1.709
0.5	GO:2376, GO:5488, GO:48518	10	10	0.1	1.414
1	GO:2376, GO:5488, GO:48518	10	10	0.1	1.414

Table 2. The best rules discovered by the Hedwig using a truncated Gene Ontology as background knowledge. A cutoff value of 0.05 means that only ontology terms ranking in the top 5% were used in rule construction. The rank was calculated by calculating the PageRank value and viewing relations as directed edges.

of the beam for the search to 10, as the results of the first round of experiments showed that the rules obtained by setting it to 1, are too vague to be of any biological interest.

The results of the second round of experiments are shown in Table 2 and Table 3. Both tables show a similar phenomenon: by decreasing the cutoff threshold the rules discovered by the Hedwig algorithm either stay the same or change to other rules with a higher lift value. For example, when searching for rules with depth set to 1, the same GO term, GO:50851, is discovered even when the size of the network is reduced to only 5% of the original network. When searching for longer rules, decreasing the size of the network by 50% still allows us to discover the same high quality conjunct of GO:2376, GO:2429, GO:5886 and GO:5488 as before, however decreasing the size further leaves a conjunct of GO:3824, GO:44283, GO:44444, GO:44238 which is more vague and less interesting.

While the increasing rule quality alone shows that using PageRank as a filter before applying the Hedwig algorithm can improve the performance of the algorithm, the results become even more promising if we also consider the fact that in the case when the cutoff threshold is low, the search space that Hedwig must analyze, and thus the computational complexity of the algorithm, is much smaller.

6 Conclusion and further work

The results show that network analysis method PageRank can be effectively used to reduce the size of the search space that needs to be examined by SDM

Cutoff	Rules	Beam	Depth	Support	Lift
0.05	GO:50851	10	1	0.01	2.687
0.1	GO:50851	10	1	0.01	2.687
0.2	GO:7584	10	1	0.01	1.811
0.5	GO:50851	10	1	0.01	2.687
1	GO:50851	10	1	0.01	2.687
0.05	GO:3824, GO:44283, GO:44444, GO:44238	10	10	0.01	3.741
0.1	GO:3824, GO:44283, GO:44444, GO:44238	10	10	0.01	3.769
0.2	GO:45936, GO:3824, GO:9892	10	10	0.01	2.219
0.5	GO:2376, GO:2429, GO:5886, GO:5488	10	10	0.01	3.42
1	GO:2376, GO:2429, GO:5886, GO:5488	10	10	0.01	3.42
0.05	GO:3824	10	1	0.1	1.322
0.1	GO:43234	10	1	0.1	1.334
0.2	GO:43234	10	1	0.1	1.524
0.5	GO:2376	10	1	0.1	1.296
1	GO:2376	10	1	0.1	1.292
0.05	GO:3824, GO:44444, GO:44710, GO:44238	10	10	0.1	1.725
0.1	GO:3824, GO:44444, GO:44710, GO:44238	10	10	0.1	1.744
0.2	GO:48518, GO:43234, GO:5488	10	10	0.1	1.721
0.5	GO:3824, GO:44444, GO:44710, GO:44238	10	10	0.1	1.639
1	GO:2376, GO:5488, GO:48518	10	10	0.1	1.414

Table 3. Table showing the second version of the experiment. The meaning of the rows is the same as in Table 2, but this time, the PageRank of terms was calculated by viewing relations as undirected edges.

methods without reducing their performance. Furthermore, the performance is in some cases even increased. This means that the proposed algorithm improvement approach shows great promise for future use of computationally expensive, but highly informative algorithms such as Hedwig, on data sets much larger than the ones used today.

In future work, we plan a more comprehensive examination of how the performance of Hedwig compares to existing enrichment methods like the SegMine method used in [22]. The comparison will be run on several biological data sets, including a data set of responses of rheumatoid arthritis patients to drug treatment.

Furthermore, we wish to perform further experiments with different methods of network reduction. For example, other network ranking methods or even other network analysis methods, such as community detection, can be used to identify the most relevant part of the background knowledge network. Also, network shrinking in our experiments was done in a basic way by simply removing all nodes whose PageRank value was too low and the edges that start or end in them. This method may cause some high ranking nodes to get “cut off” from the rest of the network, making them uninteresting for the Hedwig algorithm. In such a case, a better way may be to remove low ranking nodes, but keep the edges that start or end in them and simply extend these edges to the deleted node’s neighbors. Furthermore, we will run other experiments testing the performance of our algorithm using different settings for the Hedwig algorithm.

References

[1] Adhikari, P. R., Vavpetič, A., Kralj, J., Lavrač, N., and Hollmén, J. (2014). Explaining mixture models through semantic pattern mining and banded matrix visualization. In Džeroski, S., Panov, P., Kocev, D., and Todorovski, L., editors, *Discovery Science*, volume 8777 of *Lecture Notes in Computer Science*, pages 1–12. Springer International Publishing.

[2] Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In Bocca, J. B., Jarke, M., and Zaniolo, C., editors, *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[3] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29.

[4] Bavelas, A. (1950). Communication patterns in task-oriented groups. *Journal of the Acoustical Society of America.*

[5] Consortium, G. O. (2008). The Gene Ontology project in 2008. *Nucleic Acids Research*, 36(Database-Issue):440–444.

[6] Fisher, R. A. (1922). On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Stat. Society*, 85(1):87–94.

[7] Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

[8] Freeman, L. C. (1979). Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239.

[9] Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

[10] Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2008). Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature protocols*, 4(1):44–57.

[11] Hämäläinen, W. (2010). *Efficient search for statistically significant dependency rules in binary data.* PhD thesis, Department of Computer Science, University of Helsinki, Finland.

[12] Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43.

[13] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

[14] Klösgen, W. (1996). Explora: a multipattern and multistrategy discovery assistant. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pages 249–271. American Association for Artificial Intelligence.

[15] Lavrač, N., Kavšek, B., Flach, P. A., and Todorovski, L. (2004). Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188.

[16] Lawrynowicz, A. and Potoniec, J. (2011). Fr-ONT: an algorithm for frequent concept mining with formal ontologies. In Kryszkiewicz, M., Rybinski, H., Skowron, A., and Raś, Z. W., editors, *Foundations of Intelligent Systems, Proceedings of 19th International Symposium on Methodologies for Intelligent*

Systems (ISMIS 2011), volume 6804 of *Lecture Notes in Computer Science*, pages 428–437. Springer Berlin Heidelberg.

[17] Liu, B., Hsu, W., and Ma, Y. (1998). Integrating classification and association rule mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data mining (KDD'98)*, pages 80–86. AAAI Press.

[18] Maglott, D., Ostell, J., Pruitt, K. D., and Tatusova, T. (2005). Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, 33(Database issue):D54–D58.

[19] Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 27(1):29–34.

[20] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

[21] Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. In Piatetsky-Shapiro, G. and Frawley, W. J., editors, *Knowledge Discovery in Databases*. AAAI/MIT Press.

[22] Podpečan, V., Lavrač, N., Mozetič, I., Novak, P. K., Trajkovski, I., Langohr, L., Kulovesi, K., Toivonen, H., Petek, M., Motaln, H., et al. (2011). Segmine workflows for semantic microarray data analysis in orange4ws. *BMC bioinformatics*, 12(1):416.

[23] Srinivasan, A. (2007). Aleph Manual.

[24] Trajkovski, I., Lavrač, N., and Tolar, J. (2008a). SEGS: Search for enriched gene sets in microarray data. *Journal of Biomedical Informatics*, 41(4):588–601.

[25] Trajkovski, I., Železný, F., Lavrač, N., and Tolar, J. (2008b). Learning relational descriptions of differentially expressed gene groups. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(1):16–25.

[26] Vavpetič, A. and Lavrač, N. (2013). Semantic subgroup discovery systems and workflows in the SDM–toolkit. *The Computer Journal*, 56(3):304–320.

[27] Vavpetič, A., Novak, P. K., Grčar, M., Mozetič, I., and Lavrač, N. (2013). Semantic data mining of financial news articles. In Fürnkranz, J., Hüllermeier, E., and Higuchi, T., editors, *Proceedings of Sixteenth International Conference on Discovery Science (DS 2013)*, volume 8140 of *Lecture Notes in Computer Science*, pages 294–307. Springer Berlin Heidelberg.

[28] Žáková, M., Železný, F., Sedano, J., Tissot, C., Lavrač, N., Kremen, P., and Molina, J. (2006). Relational data mining applied to virtual engineering of product designs. In *Proceedings of the 16th International Conference on Inductive Logic Programming (ILP'06)*, pages 439–453, Santiago de Compostela, Spain. Springer-Verlag, Berlin / Heidelberg, Germany.

[29] Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In *Proceedings of the First European Conference on Principles of Data Mining and Knowledge Discovery (PKDD '97)*, pages 78–87. Springer.

[30] Xing, W. and Ghorbani, A. (2004). Weighted pagerank algorithm. In *2nd Annual Conference on Communication Networks and Services Research*, pages 305–314. IEEE.