Fault-Tolerant Concept Detection in Information Networks

Tobias Kötter[1], Stephan Günnemann[1], Michael R. Berthold[2], and Christos Faloutsos[1]

[1] Carnegie Mellon University, USA
{koettert,sguennem,christos}@cs.cmu.edu
[2] University of Konstanz, Germany
berthold@ieee.org

Abstract. Given information about medical drugs and their properties, how can we automatically discover that Aspirin has blood-thinning properties, and thus prevents heart attacks? Expressed in more general terms, if we have a large information network that integrates data from heterogeneous data sources, how can we extract semantic information that provides a better understanding of the integrated data and also helps us to identify missing links? We propose to extract concepts that describe groups of objects and their common properties from the integrated data. The discovered concepts provide semantic information as well as an abstract view on the integrated data and thus improve the understanding of complex systems. Our proposed method has the following desirable properties: (a) it is *parameter-free* and therefore requires no user-defined parameters (b) it is *fault-tolerant*, allowing for the detection of missing links and (c) it is *scalable*, being linear on the input size. We demonstrate the effectiveness and scalability of the proposed method on real, publicly available graphs.

1 Introduction

If we have two sources about medical drugs, one source listing their medical properties, and the other describing their chemical behavior, how can we discover whether, for example, Aspirin has blood thinning properties - and can therefore also help prevent heart attacks and rheumatism, in addition to being a miracle painkiller? This is the precise focus of this work; Fig. 1 shows a selection of medical uses of Aspirin (Acetylsalicylic acid) that have been detected by our proposed algorithm.

In a nutshell, we want to integrate multiple sources (e.g. Wikipedia articles describing drugs), find hidden concepts (say, 'heart disease drugs') and discover missing connections (e.g. Aspirin is related to heart disease). Information networks [9] allow the integration of such heterogeneous data by modeling the relations between objects e.g. drugs and their properties. In comparison to heterogeneous information networks [17], for example, which *a-priori* assign each node to a certain type, information networks as defined in [9] do not a-priori categorize the integrated objects into different classes: any object can be a member described by its neighbors or a property describing the members to which it is related.

Given an information network (Fig. 2), we want to identify underlying concepts (Fig. 3) and analyze them. In order to do so we extract concept graphs [10] that allow the discovery and description of concepts in such networks.

V.S. Tseng et al. (Eds.): PAKDD 2014, Part I, LNAI 8443, pp. 410–421, 2014.

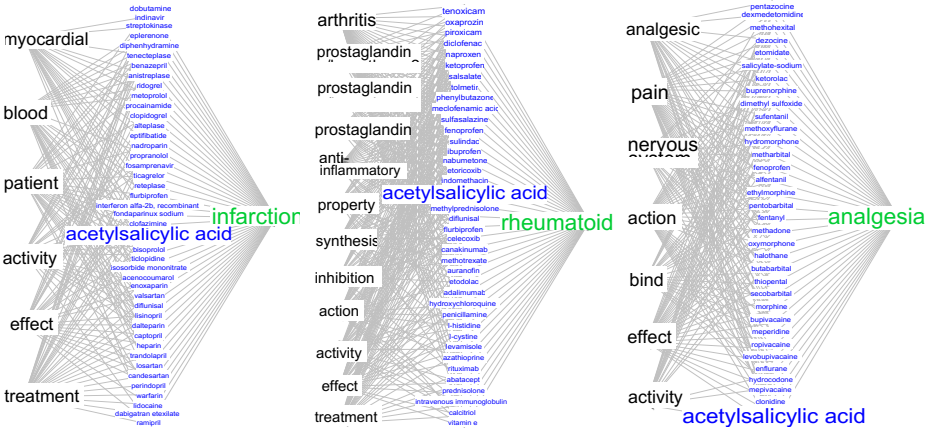


Fig. 1. Our method discovers Aspirin (Acetylsalicylic acid) as a member of the infarction, rheumatoid and analgesia (painkiller) concept graphs. See Sect. 4.1 for details.

Concept graphs allow the organization of information by grouping together objects (the *members*) that show common properties (the *aspects*), improving understanding of the concept graph and the actual concept it represents.

Concept graphs further support the abstraction of complex systems by identifying vertices that can be used as representatives for the discovered concepts. E.g., when considering the concept of animals, a number of members such as lion, bird, etc. and their characteristic aspects e.g. they are alive, can move, etc. come into our mind. In addition, they also support *context dependent vertex types* since the type of a vertex is defined by a given concept graph. Thus, a member of one concept graph could be an aspect or a concept in another concept graph. E.g., a bird might be a member of the general concept of animals but it could simultaneously be the concept representing all different kinds of birds.

An example of a concept graph representing the concept of flightless birds is depicted in Fig. 3. The concept graph consists of the members Penguin, Ostrich and Weka and their shared aspects wing and feather as well as its symbolic representation. It further

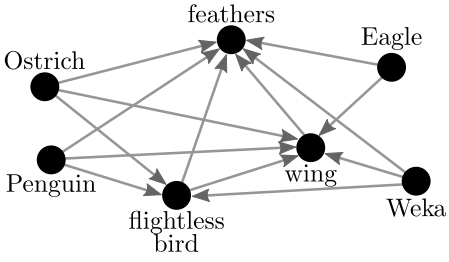


Fig. 2. Input network

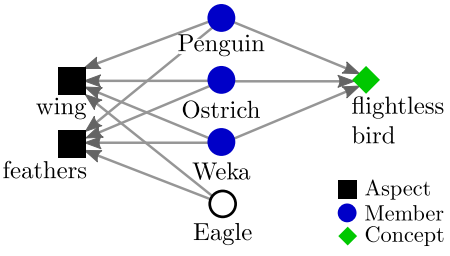


Fig. 3. Detected concept graph

contains the vertex Eagle which has the properties wing and feather but not the property flightless bird and is therefore not a member of the concept graph.

Novelty of the proposed method over competitors: The existing work [10] on identifying concept graphs is restricted to identifying only perfect concept graphs i.e. concept graphs where all members share all properties. To overcome this drawback we propose a new fault-tolerant algorithm called FCDA (Fault-tolerant Concept Detection Algorithm) to find imperfect concept graphs as well.

The main contributions of this paper are the following:

1. *Algorithm design:* We introduce FCDA, a novel *fault-tolerant* algorithm that detects concept graphs even when edges are missing, and it can spot such edges as a by-product.
2. *Automation:* FCDA *does not* require any user specified input such as the number of concepts, similarity functions, or any sort of threshold.
3. *Scalability:* The run time of FCDA grows *linearly* with the number of vertices.
4. *Effectiveness:* We evaluate our method on two real world data sets. Our results show that FCDA detects meaningful concepts that help to understand the integrated data.

For easy usage, FCDA has been implemented in KNIME [4]. The source code and experiments are available at `http://cs.cmu.edu/~koettert/pakdd2014`.

2 Proposed Method

In this section, we describe our model for fault-tolerant concept graph detection. We assume that we have a network $G = (V, E)$ with vertices V and directed edges $E \subseteq V \times V$, where $(u, v) \in E$ states that the vertex u possesses the property v. We denote the predecessors of v with $N^-(v) := (V \times \{v\}) \cap E$, and its successors with $N^+(v) := (\{v\} \times V) \cap E$. Given such a network we want to find all underlying concept graphs.

The general idea of concept graphs is to find disjoint sets of vertices $V_M, V_A \subseteq V$ such that each vertex of the member set V_M is connected to many vertices in the aspect set V_A, and vice versa. The existing method [10], which is based on frequent itemsets, was defined to identify *perfect concept graphs*. A perfect concept graph is one that forms a quasi biclique where each member is connected to all aspects of the concept. Real word data, however, is often noisy and incomplete and thus might not contain all of the connections between the members and aspects of a concept. Our solution to finding these imperfect concept graphs is to replace the strict fully connectedness requirement with a score that defines the quality of a dense quasi bipartite subgraph.

Besides finding the members (e.g. Penguin, Ostrich and Weka in Fig. 3) and aspects (e.g. wing and feather in Fig. 3) of a concept graph, one important aspect of our model is to identify vertices acting as representatives of the concept graph (e.g. flightless bird in Fig. 3). The idea is to determine those aspects of the concept graph that are most specific and simultaneously connected to all members of the concept. To measure the specificity of an aspect w.r.t. a given concept we refer to the notion of *cue validity* [3] which has been defined in [10] as

$$cv(v, V_M) = \frac{|N^-(v) \cap V_M|}{|N^-(v)|}. \tag{1}$$

For example, the cue validity of the aspect, feather, for the concept graph, flightless bird, in Fig. 3 is $\frac{3}{4}$ whereas the cue validity of the concept representative, flightless bird, is 1. Thus, flightless bird is the most specific aspect that is connected to all members of the graph and should therefore be selected as the representative. Note that the cue validity of a vertex depends on the currently selected set of members V_M.

In general, our model allows multiple vertices as representatives providing they share the same specificity. We can therefore handle concept graphs where two or more terms are used interchangeably as representatives of the same concept (e.g. synonyms) and thus refer to the same members and aspects. Overall, we define a concept graph as:

Definition 1. ***Concept graph.*** *Given an information network $G = (V, E)$, a concept graph $C = (V_M, V_A, V_C)$ is a triplet of concept graph members $V_M \subseteq V$, concept graph aspects $V_A \subseteq (V \backslash V_M)$ and concept graph representatives $V_C = \{v \in V'_C \mid cv(v, V_M) = \max_{v' \in V'_C} cv(v', V_M)\}$ such that the subgraph $S = (V_M \cup V_A, (V_M \times V_A) \cap E)$ is connected and $V_C \neq \emptyset$ with $V'_C = \{v \in V_A \mid V_M \subseteq N^-(v)\}$ defines the set of aspects, which are connected to all members of the concept.*

2.1 Concept Graph Score

The previous definition of concept graphs is very loose in the sense that many sets of vertices fulfill the definition. Thus, our goal is to focus on the 'most interesting' concepts graphs. We measure the interestingness of concept graphs via a score that incorporates three properties desired for detecting interesting concept graphs. The size of the concept graph, i.e. the number of members $|V_M|$ and aspects $|V_A|$. The connectivity of its members and aspects, i.e. the number of (missing) links connecting its members and aspects. The specificity of the aspects for the given concept (see Eq. 1). In general, the larger the concept graph, the more densely connected the two sets and the more specific the aspects, the higher the score.

Definition 2. *The concept graph score cs is defined as follows*

$$cs(V_M, V_A) = \sum_{m \in V_M} \sum_{a \in V_A} \epsilon(a, m) \frac{|N^-(a) \cap V_M|}{|N^-(a)|} = \sum_{m \in V_M} \sum_{a \in V_A} \epsilon(a, m) cv(a, V_M) \tag{2}$$

with $\epsilon(a, m) = 1$ if $m \in N^-(a)$, otherwise -1.

Given this definition, we are now interested in finding those concept graphs that maximize the score.

2.2 The FCDA Algorithm

The fault-tolerant detection of concept graphs is similar to the detection of maximum quasi-bicliques, which is NP-complete [13]. Thus, we cannot expect to find an efficient algorithm computing an exact solution. To improve performance we propose a greedy algorithm that maximizes the quality score cs of a given concept graph.

Intuition behind our algorithm: (Part 1) For each vertex $c' \in V'_C$ with incoming edges, extract its predecessors (potential members) and their successors (potential aspects). (Part 2) Optimize the two discovered sets using Eq. 2. The details are as follows:

Part 1: Find Initial Concept Graph

Step 1. Extract the set of potential members V'_M for the potential concept $c' \in V'_C$ which is the set of its predecessors $V'_M = N^-(c')$.

Step 2. Extract the set of potential aspects V'_A which are the successors of the potential members V'_M that are equal or less specific than the potential concept vertex c' thus $V'_A = \left\{ a \in \bigcup_{m \in V'_M} N^+(m) : |N^-(a)| \geq |N^-(c')| \right\}$.

Once we have extracted the set of potential aspects V'_A and potential members V'_M we have to identify the optimal subsets given the concept graph score cs (see Eq. 2). *Computational Speed considerations:* Obviously, enumerating all possible combinations to find the optimal solution is intractable that is why we follow an iterative approach.

Part 2: Refine Concept Graph

Step 3. Set $V^*_A = V'_A$, i.e. V^*_A contains all potential aspects.

Step 4. Find the subset of members V^*_M that maximizes the concept graph score cs for the current aspect set V^*_A by adding each member that improves the score, i.e. $V^*_M = \arg\max_{V^*_M \subseteq V'_M} cs(V^*_A, V^*_M)$.

Step 5. If the current concept graph score $cs(V^*_A, V^*_M) > cs(V_A, V_M)$ is the highest for the current potential concept c' set $V_M = V^*_M$ and $V_A = V^*_A$.

Step 6. Remove the aspect a' from V^*_A that has the least members in V'_M, i.e. the one with the lowest value for $|N^-(a') \cap V'_M|$, and start over from step 4 until V^*_A is empty.

Step 7. Once the aspect loop has been terminated, iterate over the remaining potential aspects $V'_A \setminus V_A$ and add all vertices to V_A that improve the score. This is necessary because an aspect might have been removed early that would improve the score due to vertices in V^*_A with a negative score. Accordingly, we iterate over the set $V'_M \setminus V_M$ to add vertices to V_M that improve the score.

Step 8. Finally check for vertices that are members of both sets V_A and V_M. This situation can arise as a vertex can have incoming and outgoing edges and thus end up in both sets. Remove a vertex $v \in V_M \cap V_A$ from the set that affects the score the least. E.g., if $v \in V_A \wedge v \in V_M \wedge cs(V_M/v, V_A) < cs(V_M, V_A/v)$ holds, remove v from V_M otherwise remove v from V_A.

By following these steps we extract the concept graphs and their scores for each potential concept $c' \in V'_C$. Since the computation of each potential concept $c' \in V'_C$ is independent of other concepts, computation is easily parallelized, which subsequently improves the run time significantly (Fig. 5c).

Efficient Incremental Score Computation. The above algorithm requires computing the concept graph score cs at multiple places. To speed up the computation we can make use of an incremental computation of the concept graph score. The following equations hold when adding or removing a single member or aspect from the concept graph:

• Adding a member m^+ to V_M:

$$cs(V_M \cup m^+, V_A) = cs(V_M, V_A) + \sum_{a \in V_A} \epsilon(a, m^+) \frac{|N^-(a) \cap (V_M \cup m^+)|}{|N^-(a)|} + \sum_{a \in V_A} \sum_{m \in V_M} \epsilon(a, m) \frac{|N^-(a) \cap m^+|}{|N^-(a)|}. \quad (3)$$

• Adding an aspect a^+ to V_A:

$$cs(V_M, V_A \cup a^+) = cs(V_M, V_A) + \sum_{m \in V_M} \epsilon(a^+, m) \frac{|N^-(a^+) \cap V_M|}{|N^-(a^+)|}. \quad (4)$$

• Removing a member m^- from V_M:

$$cs(V_M / m^-, V_A) = cs(V_M, V_A) - \sum_{a \in V_A} \epsilon(a, m^-) \frac{|N^-(a) \cap V_M|}{|N^-(a)|} - \sum_{a \in V_A} \sum_{m \in V_M / m^-} \epsilon(a, m) \frac{|N^-(a) \cap m^-|}{|N^-(a)|}. \quad (5)$$

• Removing an aspect a^- from V_A:

$$cs(V_M, V_A / a^-) = cs(V_M, V_A) - \sum_{m \in V_M} \epsilon(a^-, m) \frac{|N^-(a^-) \cap V_M|}{|N^-(a^-)|}. \quad (6)$$

2.3 Missing Link Recovery

The missing link recovery approach is based on the information from the discovered concept graphs. All of the connections that need to be added to a concept graph in order to make it fully connected are potentially missing links (see Fig. 4). However, not all of them are interesting or missing. That is why we provide two scoring functions that allow the global sorting of the recovered missing links based on their confidence and interestingness. Since the (missing) edge (m, a) between a member m and an aspect a can be missing in several concept graphs, we use $\mathbb{C}_{a,m}$ to denote all concept graphs that have the member $m \in V_M$ and the aspect $a \in V_A$.

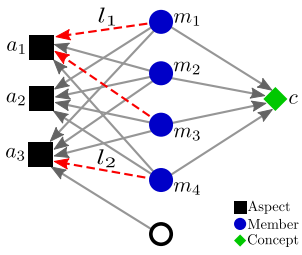


Fig. 4. Missing links example

Confidence. To measure the confidence of a missing link we can use the structure of the concept graphs based on the consideration that a single missing link in a large concept graph is much more likely to be an artifact than a link in a smaller concept graph that misses many links. E.g. the missing link l_1 in Fig. 4 has a lower confidence than the missing link l_2 since the aspect a_1 is only connected to two whereas the aspect a_3 is connected to three of the four concept members. We take the minimum to ensure that the possibility of the missing link is high for all concept graphs in $\mathbb{C}$.

Definition 3. *The confidence score of a missing link between a member* m *and an aspect* a *is defined as* $conf(m,a) = \min_{C(V_A,V_M,V_C)\in\mathbb{C}_{a,m}} \frac{|N^+(m)\cap V_A|}{|V_A|} \cdot \frac{|N^-(a)\cap V_M|}{|V_M|}$.

Interestingness. The interestingness of a recovered link is related to the cue validity (see Eq. 1) of the aspect. For example, a missing link to a very general aspect with a low cue validity is not as interesting as a missing link to a more specific concept with a high cue validity. E.g. The missing link l_1 in Fig. 4 is potentially more interesting than the missing link l_2 since the aspect a_1 has a cue validity of 1 whereas the aspect a_3 has a cue validity of $\frac{3}{4}$. Since the cue validity of an aspect is context dependent we take the minimum to ensure that the aspect is interesting from a global point of view.

Definition 4. *The global interestingness score of a missing link between a member* m *and an aspect* a *is defined as* $int(m,a) = \min_{C(V_A,V_M,V_C)\in\mathbb{C}_{a,m}} cv(a,V_M)$.

3 Experiments

This section demonstrates the quality of the proposed method based on networks that were extracted from two publicly available real world data sets from different domains.

3.1 Datasets

Wikipedia Selection for Schools. (Schools Wikipedia)[1] is a selection of the English Wikipedia for children. It has about 5500 articles organized into 154 subjects such as countries, religion, and science. Each article and each related subject is represented by a vertex. Hyperlinks are represented by directed edges with the article that contains the hyperlink as the source and the referenced article or subject as the target vertex. This results in a network with 5,770 vertices and 231,985 edges.

DrugBank. [8] is a publicly available data base with more than 6,000 entries describing 1,578 approved drugs and 5,000 experimental substances. Each drug is described by unstructured information e.g. textual descriptions as well as structured information such as target information. Drugs and their extracted properties are represented by vertices. The connections between drugs and their properties are represented by directed edges with the drugs as the source and the properties as target vertices. This results in a network with 18,574 vertices and 109,721 edges.

3.2 Quality

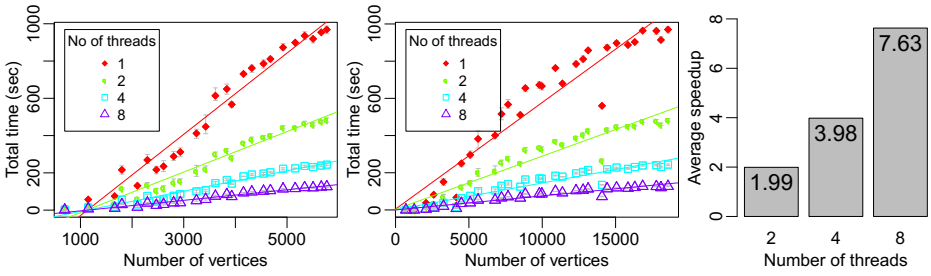
Since the proposed method is unique in its characteristics and most of the more similar methods require some kind of parameter to define the number of clusters, it is difficult to undertake a comparison. Therefore, we decided to use information that exists in each data source, defining specific groups of objects, to evaluate our results. In Schools Wikipedia we used the subject pages and the related subjects of each article. This leads

[1] `http://schools-wikipedia.org`

to a total of 154 groups including topics like dinosaurs, chemical elements, computer programming, and artists. For DrugBank we used the drug classes as well as the pharmaceutical and pathway based classification available from the DrugBank homepage. This leads to a total of 651 groups of which 584 corresponding vertices can be found in the network. Given the predefined groups of objects and the concept graphs detected by our method, we can compute the F_1 score [16] to measure the quality of our algorithm. The F_1 score for Schools Wikipedia is 0.803 with precision 0.769 and recall 0.924. For DrugBank the F_1 score is 0.859 with precision 0.863 and recall 0.862.

3.3 Scalability

In this section, we demonstrate the time complexity and parallelizability of FCDA experimentally. In order to measure the running time on different sized graphs we extracted several subgraphs of different sizes from the two data sets using snowball sampling [18]. Figure 5 shows the running time w.r.t. increasing number of vertices as well as the average speedup over all experiments.



(a) Schools Wikipedia (b) DrugBank (c) Average speedup

Fig. 5. Run time of FCDA versus the number of vertices (average over 5 runs) and average speedup over all experiments

Usually the two vertex sets V_A and V_M form a Galois connection [7], which describes the correspondence between two partially ordered sets. It states that if one of the two sets increases in size, the size of the other set will decrease. Therefore, if we are looking at a very general concept with a lot of members the set of aspects is very small and vice versa. This behavior contributes to the efficiency of the algorithm.

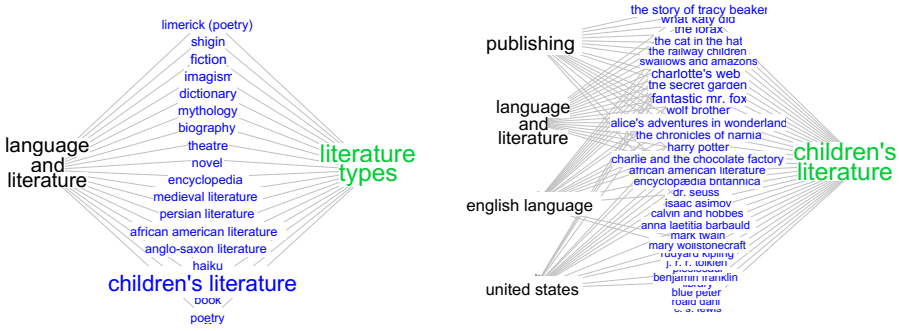
4 Discoveries

This section describes the discoveries we were able to make by applying our proposed method to the two data sets mentioned above. The concept graph images always depict the **aspects** of the concept graph in the left column, the members in the middle column, and the concept representative in the right column. In addition, aspects are ordered based on their cue validity from highly specific at the top to more general at the bottom.

4.1 Concept Detection

Observation 1 (Aspirin as blood thinner). *The concept graphs in Fig. 1 demonstrate the ability to discover members e.g. Aspirin (Acetylsalicylic acid) that are part of different concepts e.g. painkillers, rheumatoid and infarction. It further shows the value of the extracted semantic information which helps to better understand the extracted concepts e.g that rheumatism is caused by inflammation.*

Observation 2 (Context dependent vertex types). *The ability to model context dependent vertex types (e.g. concept, aspect or member), of concept graphs and* FCDA *is necessary to model real world concepts. See Fig. 6 for an example of two concept graphs from Schools Wikipedia where the type of the children's literature vertex changes from member (Fig. 6a) to concept (Fig. 6b).*



(a) Children's literature as member.

(b) Children's literature as concept.

Fig. 6. Children's literature as an example for context dependent vertex types

4.2 Missing Link Recovery

Thanks to the fault-tolerance of FCDA we can identify missing links between an aspect and a member of a detected concept graph. Using the semantic information provided by the discovered concept graphs we can further assign a global interestingness and a confidence score to all potentially missing links. Table 1 shows the top 5 missing links for the DrugBank data set based on the combined scores.

Observation 3 (Haloperidol can impair the effect of antiparkinson agents). *A very interesting connection is the one between Haloperidol and 'antiparkinson agents'. This connection is not directly mentioned on the DrugCard. However, by searching for the two terms we learned that Haloperidol can impair the effect of antiparkinson agents [1]. Thus, our method successfully reveals novel information which is not encoded in the current data set but predicted based on the discovered concept graphs.*

Table 1. Top 5 missing links based on confidence and interestingness

Aspect (Property)	Member (Drug)	$conf$	int	$conf \cdot int$
anhydrases	trichlormethiazide	0.82	0.86	0.70
shigella	netilmicin	0.73	0.89	0.65
cell-cell	enflurane	0.72	0.8	0.58
aminoglycoside antibiotic	neomycin	0.72	0.8	0.57
antiparkinson agents	**haloperidol**	0.70	0.81	0.57

Observation 4 (Detected missing links are often true omissions). *The top 5 discovered missing links in Table 1 are true omissions. Most of them do not exist in the network due to the limitations of the used text mining algorithm. For example, the connection between 'aminoglycoside antibiotic' and Neomycin is mentioned on its DrugCard but does not exist in the network. However, thanks to the missing link recovery we are still able to recover these links.*

5 Related Work

Network-Based Approaches: *(Quasi)-clique detection* methods [12] aim at finding dense subgraphs in a given graph. These methods do not distinguish between the members and aspects of a subgraph but treat all vertices equally. Thus, these methods cannot be used to find concept graphs since, e.g., the members of a concept graph need not be connected at all. Methods [11] have been proposed for *quasi-biclique detection*, which explicitly distinguish between members and aspects. The existing methods, however, are not aware of the concept of cue validity and are not able to determine representative vertices for each concept. *Block-Modeling* techniques [2,5] try to find homogeneous blocks in the adjacency matrix of a graph. However, none of the existing approaches takes the cue validity into account and are not able to identify representative vertices. Moreover, most of the techniques require important parameters to be specified manually, e.g., the maximal fraction of missing edges allowed in a pattern.

Global Pattern Discovery: Finding vertices sharing similar properties might be regarded as an instance of *shared nearest neighbor* clustering. That is, using the distance function d that assigns low values to pairs of vertices sharing many neighbors and high values to pairs of vertices sharing no neighbors, any distance based clustering method as, e.g., k-medoid, might be used to find concept graphs. This principle, however, does not consider the current context of the subgraph as our method does. Additionally, most of the methods either do not identify representative vertices or do not allow for overlapping clusters or require certain parameters e.g. the number of clusters to detect.

Local Pattern Discovery: *Co-Clustering/Biclustering* [14] is the task of simultaneously clustering the rows and columns of a data matrix. In our scenario, the data matrix would correspond to the binary adjacency matrix. Co-clustering can be roughly divided into four categories [14]. According, our method is mostly related to the category for finding patterns with constant values, as we are interested in detecting dense (bipartite)

subgraphs of the network. In this regard, co-clustering would reflect a certain kind of block-modeling sharing the same drawbacks as discussed above. *Frequent itemset mining* aims at finding groups of items that frequently occur together in a set of transactions. By considering each row of the adjacency matrix as a transaction, and each column as an item, frequent itemset mining can be used to detect bicliques in a network. This idea has been exploited by existing methods for concept graph detection [10]. However, when based on frequent itemsets, these methods are highly sensitive to errors and missing values in the data. While fault-tolerant extensions of frequent itemset mining [15] and, even more general, subspace clustering [6] have been proposed to handle missing data, such methods are often not scalable and require hard to set parameters. In contrast, our technique requires no user defined parameters and the computational costs increase linear in the number of vertices.

In conclusion, none of the described methods supports all of our goals, namely, scalability, fault-tolerance, parameter free, dynamic vertex types, and the detection of concept representatives.

6 Conclusions

We have proposed FCDA, the first fault-tolerant and parameter-free method for finding concept graphs in information networks. The detected concept graphs provide valuable information about the members of a concept and their characteristic properties, improving understanding of the concept graph and the represented concept itself. They further support the abstraction of complex systems by identifying vertices that can be used as representatives for the discovered concepts. In addition, they also support overlapping concept graphs and context dependent vertex types. The main contributions of our work include:

- *Algorithm design:* We introduces FCDA, a novel algorithm that detects concept graphs in information networks even when edges are missing, and it can spot such edges as a by-product.
- *Automation:* FCDA *is fully automatic*. It does not require any user specified input such as the number of concepts, similarity functions, or any sort of threshold.
- *Scalability:* The run time of FCDA grows *linearly* with the number of vertices.
- *Effectiveness:* We demonstrate that FCDA detects meaningful concepts that help to understand the integrated data in diverse real-world data sets. E.g. the discovery of Aspirin's benefit in the treatment of infarction and rheumatism in addition to it being a painkiller. We further show the potential of the algorithm for discovering missing relations such as the detection that Haloperidol can impair the effect of antiparkinson agents.

Acknowledgments. T. Kötter was supported by stipend KO 4661/1-1 of the "Deutsche Forschungsgemeinschaft" (DFG). S. Günnemann was supported by a fellowship within the postdoc-program of the German Academic Exchange Service (DAAD). This material is based upon work supported by the National Science Foundation under Grant No. IIS-1247489. Research was also sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, DARPA, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

1. Ahmed, S.P., Siddiq, A., Baig, S.G., Khan, R.A.: Comparative efficacy of haloperidol and risperidone: A review. Pakistan Journal of Pharmacology 24, 55–64 (2007)
2. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. Journal of Machine Learning Research 9, 1981–2014 (2008)
3. Beach, L.R.: Cue probabilism and inference behavior. Psychological Monographs: General and Applied 78, 1–20 (1964)
4. Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., Wiswedel, B.: KNIME: The Konstanz Information Miner. In: Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007). Springer (2007)
5. Chakrabarti, D., Papadimitriou, S., Modha, D.S., Faloutsos, C.: Fully automatic cross-associations. In: KDD, pp. 79–88 (2004)
6. Günnemann, S., Müller, E., Raubach, S., Seidl, T.: Flexible fault tolerant subspace clustering for data with missing values. In: ICDM, pp. 231–240 (2011)
7. Herrlich, H., Husek, M.: Galois connections. In: Mathematical Foundations of Programming Semantics, pp. 122–134 (1985)
8. Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., Pon, A., Banco, K., Mak, C., Neveu, V., Djoumbou, Y., Eisner, R., Guo, A.C., Wishart, D.S.: Drugbank 3.0: a comprehensive resource for 'omics' research on drugs. Nucleic Acids Research 38, 1–7 (2010)
9. Kötter, T., Berthold, M.R.: From information networks to bisociative information networks. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 33–50. Springer, Heidelberg (2012)
10. Kötter, T., Berthold, M.R.: (Missing) concept discovery in heterogeneous information networks. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 230–245. Springer, Heidelberg (2012)
11. Li, J., Sim, K., Liu, G., Wong, L.: Maximal quasi-bicliques with balanced noise tolerance: Concepts and co-clustering applications. In: SDM, pp. 72–83 (2008)
12. Liu, G., Wong, L.: Effective pruning techniques for mining quasi-cliques. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 33–49. Springer, Heidelberg (2008)
13. Liu, X., Li, J., Wang, L.: Quasi-bicliques: Complexity and binding pairs. In: Hu, X., Wang, J. (eds.) COCOON 2008. LNCS, vol. 5092, pp. 255–264. Springer, Heidelberg (2008)
14. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1, 24–45 (2004)
15. Poernomo, A.K., Gopalkrishnan, V.: Towards efficient mining of proportional fault-tolerant frequent itemsets. In: KDD, pp. 697–706 (2009)
16. Rijsbergen, C.J.V.: Information Retrieval, 2nd edn. Butterworth-Heinemann, Newton (1979)
17. Sun, Y., Yu, Y., Han, J.: Ranking-based clustering of heterogeneous information networks with star network schema. In: KDD, pp. 797–806 (2009)
18. Thompson, S.: Sampling. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., New York (2002)