MING: Mining Informative Entity Relationship Subgraphs

Gjergji Kasneci
Max-Planck Institute for
Informatics, Saarbrücken
kasneci@mpii.de

Shady Elbassuoni
Max-Planck Institute for
Informatics, Saarbrücken
elbass@mpii.de

Gerhard Weikum
Max-Planck Institute for
Informatics, Saarbrücken
weikum@mpii.de

ABSTRACT

Many modern applications are faced with the task of knowledge discovery in entity-relationship graphs, such as domain-specific knowledge bases or social networks. Mining an "informative" subgraph that can explain the relations between $k(\geq 2)$ given entities of interest is a frequent knowledge discovery scenario on such graphs. We present MING, a principled method for extracting an informative subgraph for given query nodes. MING builds on a new notion of informativeness of nodes. This is used in a random-walk-with-restarts process to compute the informativeness of entire subgraphs.

Categories and Subject Descriptors

H.0 [**Information Systems**]: General—*knowledge discovery*

General Terms

Algorithms, Design

1. INTRODUCTION

Many modern applications exploit information organized in entity-relationship (ER) graphs, such as domain-specific knowledge bases (e.g. metabolic or regulatory networks in biology, criminalistic networks for crime investigation, etc.) or social networks (such as data sharing or business-customer networks). One can represent them by relational or ER models, XML with XLinks, or RDF triples. An example of an ER graph is YAGO [17], which has been constructed by systematically harvesting semi-structured assets of Wikipedia (e.g., infoboxes, categories, lists, etc.). The YAGO graph consists of millions of nodes (representing entities, e.g. persons, movies, locations, dates, etc.) and tens of millions of labeled edges, representing facts about entity pairs, such as *Max_Planck fatherOf Erwin_Planck*, *Max_Planck isA Physicist*, *Max_Planck bornIn Germany*, etc. YAGO supports more than 100 relationship labels such as *isA, bornIn, citzenOf, marriedTo,* etc.

Other examples for ER graphs are GeneOntology or UMLS (in the biomedical domain), the graphs represented by IMDB (in the domain of movies and actors), DBLP (in the domain of Computer Science publications), and LOD [3] (for publishing interlinked Web data sets as RDF graphs), etc.

A knowledge discovery task on such graphs is to determine an "informative" subgraph that can explain the relations between $k(\geq 2)$ entities of interest. Examples are queries that aim at finding the relations between k given biomedical entities, the connections between k criminals, the most relevant data shared by k Web 2.0 users, etc. Formally, the general problem that motivates this work is: given a set $Q = \{q_1, ..., q_k\}, k \geq 2$, of nodes of interest (i.e. query nodes) from an entity relationship graph G and an integer $b > k$ (representing a node budget), find a connected subgraph S of G with at most b nodes that contains all query nodes and maximizes an "informativeness" function $g_{info}(S, Q)$.

As a concrete example, consider the query that asks for the relation between *Max_Planck, Albert_Einstein,* and *Niels_Bohr*. An informative subgraph that captures their relatedness should reveal that all three of them are physicists, scientists, Nobel Prize winners, etc., and should discourage long or obscure connections (e.g. connections through persons with same nationalities or same birth or death places as some of the query entities). Figure 1 depicts a possible answer.

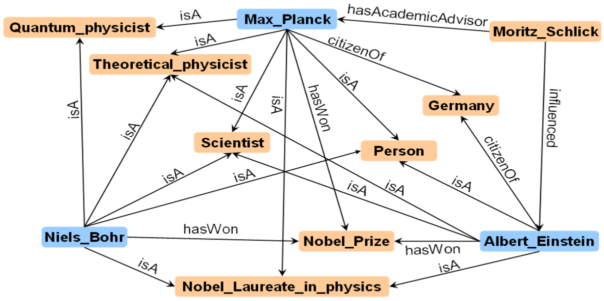

Figure 1: Answer returned by MING on YAGO

The above problem comes with two subproblems: (1) what is a good measure for representing the informativeness of relations between entities in ER graphs? (2) how to determine the most informative subgraph for the given query nodes? We address both problems in this work.

In previous approaches [16, 5, 7, 18, 8, 15], the notion of subgraph importance is mainly based on structural properties of the underlying graph (e.g. indegree or outdegree of a node, density or edge connectivity of a subgraph, etc.). More related to our approach are techniques based on influence propagation like [7] or [18]. The approach of [7] exploits a current-flow algorithm to determine an important subgraph for two query nodes. The approach of [18], CEPS, can handle more than two query nodes, and gives a random-walk-based solution for retrieving the most "central" nodes with respect to the query nodes, so called *centerpieces*: nodes that are closely connected to most of the query

CIKM'09, November 2–6, 2009, Hong Kong, China.

nodes. The centerpieces are exploited to mine an important subgraph for the query nodes. All mentioned approaches leave aside the problem of deriving measures for capturing the semantic importance of nodes and edges in ER graphs. Other, Steiner-tree-based, approaches [1, 11, 2, 12, 10, 6, 9, 13] have addressed the problem of retrieving the top-k minimum-cost subtrees that closely interconnect the given query nodes. Their result paradigm is tree-based. Hence, these approaches are not directly applicable to our problem of retrieving informative subgraphs.

Our approach gives a principled solution, while making the semantic aspect of entities and relationships in ER graphs a key ingredient for the measure of informativeness. Our main contributions are the following:

1. We give a clean notion of informativeness for nodes in ER graphs. Our informativeness measure builds on a natural extension of the random surfer model that underlies PageRank [4]. This measure is exploited to capture the informativeness of entire subgraphs.
2. We present MING, a robust algorithm for mining and extracting most informative subgraphs for $k(\geq 2)$ query nodes.
3. Based on user assessments, we demonstrate the quality of MING in comparison to prior work.

2. ER-BASED INFORMATIVENESS

In the following definition we introduce ER graphs as multi-graphs.

Definition 1 (ER graph). *Let Ent and Rel be finite sets of entity and relationship labels respectively. An ER graph over Ent and Rel is a tuple* $G = (V, l_{Ent}, E_{Rel})$, *where* V *is a set of nodes,* $l_{Ent} : V \rightarrow Ent$ *is an injective function,* $E_{Rel} \subseteq l_{Ent}(V) \times Rel \times l_{Ent}(V)$ *is a set of labeled edges.*

Since the direction of a relationship between two entities can always be interpreted in the converse direction, we view the edges of an ER graph as bidirectional. That is, we assume that for each edge $(u, r, v) \in E_{Rel}$ there is an edge $(v, r^{-}, u) \in E_{Rel}$, where r^{-} represents the inverse relation label of r.

For any subgraph S of an ER graph G, we denote by $Ent(S)$ the set of its labeled nodes (i.e., entities), and by $F(S)$ the set of its labeled edges (i.e., facts). Note that $F(S)$ contains edges of the form (α, β, γ), and that both $\alpha, \gamma \in Ent(S)$.

Discussion We believe that in order to compute the informativeness of nodes in ER graphs, the link structure has to be taken into account. But, as a matter of fact, edge directions in ER graphs do not always reflect a "clear" endorsement. For example, the fact *Albert_Einstein isA Physicist* can be represented as *Physicist hasInstance Albert_Einstein*. Consequently, measures that build on the link-based endorsement hypotheses such as PageRank [4] or HITS [14] are not applicable in a straight-forward way. In general, we argue that measures that rely on the graph structure alone are not sufficient, since ER graphs represent only a limited fraction of the real world.

Our informativeness measure for nodes overcomes these problems by building on edge weights that are based on co-occurrence statistics for entities and relationships. These statistics are derived from the domain represented by the ER graph. They will guide a random walk process on the adjacency matrix of the ER graph.

2.1 Statistics-based Edge Weights

For each fact represented by an edge, we compute two weights; one for each direction of the edge. Each of these weights will represent a special kind of endorsement, obtained from co-occurrence statistics for entities and relationships.

Definition 2 (Fact Pattern, Match, Binding). *Let* X *be a set of entity variables (placeholders for entities). A* fact pattern *from an ER graph* $G = (V, l_{Ent}, E_{Rel})$ *is a triple* $(\alpha, \beta, \gamma) \in (Ent \cup X) \times Rel \times (Ent \cup X)$, *in which either* $\alpha \in X$ *or* $\gamma \in X$, *such that if* $\alpha \in X$ *then there exists* $(\alpha', \beta, \gamma) \in E_{Rel}$, *and if* $\gamma \in X$ *there exists* $(\alpha, \beta, \gamma') \in E_{Rel}$.

Without loss of generality, let $\alpha \in X$. *The edge* (α', β, γ) *from* G *is called a* match *to the fact pattern* (α, β, γ), *and the entity* α' *is called a* binding *to the variable* α.

Consider the fact pattern x *isA Physicist*, $x \in X$. The facts *Albert_Einstein isA Physicist* and *Bob_Unknown isA Physicist* are matches to the above fact pattern. In our example, the fact *Albert_Einstein isA Physicist* should have a higher informativeness than *Bob_Unknown isA Physicist*, since Einstein is an important individual among the scientists. More precisely, the binding *Albert_Einstein* should be more informative than *Bob_Unknown*. To capture this notion of informativeness, we introduce a probabilistic model.

Let (α, β, γ) be a fact pattern, where $\alpha \in X$. Let α' be a binding of α. We estimate the informativeness of α' given the relationship β and the entity γ as:

$$P_{info}(\alpha'|\beta, \gamma) = \frac{P(\alpha', \beta, \gamma)}{P(\beta, \gamma)} \approx \frac{W(\alpha', \beta, \gamma)}{W(\beta, \gamma)} \qquad (1)$$

where $W(\alpha', \beta, \gamma)$ denotes the number of domain witnesses for the fact $\alpha'\ \beta\ \gamma$, i.e., the number of its occurrences in the underlying domain of the ER graph. Analogously, $W(\beta, \gamma)$ stands for the number of witnesses for the pattern $(*, \beta, \gamma)$, where the wild card '$*$' can be any entity. The value $P_{info}(\alpha'|\beta, \gamma)$ is assigned as a weight to the edge $\gamma \xrightarrow{\beta} \alpha'$.

In practice, these values can be estimated by means of inverted indexes on a background corpus, e.g. a large Web sample representing the domain of the ER graph. From the indexes one can compute (co-)occurrence statistics for (pairs of) entity names and estimate the needed parameters.

2.2 IRank for Node-based Informativeness

Our aim is an informativeness measure for nodes based on random walks on the – now weighted – ER graph. Our measure, coined *IRank* (Informativeness Rank), is related to PageRank [4].

Let $G = (V, l_{Ent}, E_{Rel})$ be an ER graph. Let $u \in l_{Ent}(V)$ be an entity and let $P(u)$ be the probability of encountering the entity u in the domain from which G was derived. This value can be estimated as $P(u) \approx \frac{W(u)}{\sum_{v \in Ent} W(v)}$, where again $W(u)$ denotes the number of occurrences of the entity u in the underlying domain. $P(u)$ can be viewed as an importance prior for u.

In IRank, the random surfer may decide to restart his walk from an entity $u \in l_{Ent}(V)$ with probability proportional to $P(u)$. Alternatively, the surfer may reach u from any neighboring entity v that occurs in an edge of the form $(v, r, u) \in E_{Rel}$ (given that the surfer is at one of these neighboring entities of u).

Let $N(u)$ denote the set of neighboring entities of u in G. The probability of reaching u via one of its neighbors would be proportional to:

$$\sum_{v \in N(u)} \sum_{\substack{r \\ (v,r,u) \in E_{Rel}}} P_{info}(u|r, v) \cdot IR(v) \qquad (2)$$

where $IR(v)$ denotes the probability that the surfer is at v.

Finally, the accumulated informativeness at a node $u \in l_{Ent}(V)$ is given by:

$$IR(u) = (1-q)P(u) + q \sum_{\substack{v,r \\ (v,r,u) \in E_{Rel}}} P_{info}(u|r,v) \cdot IR(v) \quad (3)$$

For practical reasons, the outgoing edge weights (i.e., the P_{info} weights) for each entity u are normalized by the sum of all outgoing edge weights of u. With this normalization step, Equation (3) represents an aperiodic and irreducible finite-state (i.e., an ergodic) Markov Chain. This guarantees the convergence and the stability of IRank. Although IRank is related to PageRank, the P_{info} values are crucial and make a big difference in the random walk process.

2.3 Most Informative Subgraphs with MING

As a first step, MING extracts a subgraph C of G that contains many connections between the query nodes. This is a rather recall-oriented step; most of the spurious regions of G are removed. The subgraph C can be extracted with heuristics similar to the ones presented in [7]. In a second step, we run the STAR algorithm [13] to determine a subtree T of C that closely interconnects all entities from Q. Assuming that T already captures some relatedness between the query entities, each node $v \in Ent(T)$ is viewed as informative; these nodes are assigned the label '+'. Nodes on the "rim" of C that do not represent query entities and have degree 1, i.e., nodes that do not contribute to any connection between query entities, are viewed as uninformative, and are labeled '$-$'.

For each unlabeled node $v \in Ent(C)$, we compute a score $P_-(v)$, representing how uninformative v is, and a score $P_+(v)$, representing how informative v is, with respect to the query entities.

The informative subgraph mining problem can be stated as follows.

Definition 3 (Informative Subgraph Mining). *Given the connected subgraph C, the set Q of query nodes, and an integer node budget $b >= |Ent(T)|$, solve the tasks:*

1. *Determine for each $v \in Ent(C)$ a label $lab(v) \in \{-,+\}$ as $lab(v) = \arg\max_{l \in \{-,+\}} P_l(v)$.*
2. *Extract a connected subgraph S of C that contains T and has the following properties: (1) every $v \in Ent(S)$ is labeled '+', (2) S contains at most b nodes, (3) S maximizes $\sum_{v \in Ent(S)} P_+(v)$.*

With the requirement that T be a subgraph of S, we guarantee that all query nodes are interconnected in the result graph.

Classification Method Our intuition is the following. Let $l \in \{-,+\}$. Consider all paths in C that connect any two nodes labeled l and cross an unlabeled node v. The higher the number of such paths, the higher the probability that v is also labeled l. On the other hand, the longer these paths, the smaller the probability that v is labeled l. To estimate $P_l(v)$, we need methods that capture and reward robust structural connectivity and discourage long and loose connections.

Consider a random walker that starts at a node labeled l in C and finishes his walk again at a node labeled l. For an unlabeled node $v \in Ent(C)$, let $P_l(v)$ denote the probability that v is visited during this random walk. As depicted in Figure 2, we estimate this probability as the composition of two probabilities $P_l^1(v)$ and $P_l^2(v)$. $P_l^1(v)$ represents the probability that the random walker starts at any l-labeled node and reaches v. $P_l^2(v)$ represents the probability that any l-labeled node is reached when the random walker starts his walk at v. It is straightforward to see that $P_l(v) = P_l^1(v) \cdot P_l^2(v)$.

In order to estimate $P_l^1(v)$, we extend IRank into a Random Walk with Restarts (RWR) process that restarts from the nodes labeled l. The walk starts at any l-labeled node v and follows the outgoing edges of v with a probability that is proportional to the edge weights (as edge weights on C we consider the P_{info} values from Equation (1)). The probability that our walk follows the outgoing edges of nodes is dampened by a factor q. With a probability $(1-q)$ the random walk restarts at any node labeled l.

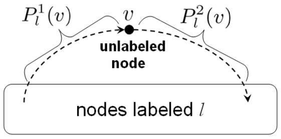

Figure 2: Probability P_l composed of P_l^1 and P_l^2.

In RWR processes long connectivity paths are discouraged in a natural way (by the restart probability). Furthermore, as reported in [19] and [18], RWRs have nice properties when it comes to capturing the structural connectivity between nodes. They overcome several limitations of traditional graph distance measures such as maximum flow, shortest paths, etc.

In order to compute P_l^2 for an unlabeled node v, we could use again RWRs. More precisely, we could run an RWR for every unlabeled node v and compute $P_l^2(v)$ as $P_l^2(v) = \sum_{u:lab(u)=l} P_v(u)$, where $P_v(u)$ would denote the stationary probability of u as determined by the RWR starting at v. However, there might be several hundreds of unlabeled nodes in C, and running an RWR for each of the unlabeled nodes is highly inefficient in practice. Hence, we estimate the P_l^2 in a more relaxed but more efficient way.

Let u be an unlabeled node in C. The probability of having been at node u one step before reaching any node v labeled l is given by:

$$P(u,1) = \sum_{\substack{v:lab(v)=l \\ v \in N(u)}} P_{info}(v|u) \quad (4)$$

where $N(u)$ denotes the set of neighboring nodes of u in C, and $P_{info}(v|u)$ is defined as:

$$P_{info}(v|u) := \sum_{\substack{r \\ (u,r,v) \in F(C)}} P_{info}(v|r,u)$$

Let $L \subseteq Ent(C)$ denote the set of nodes labeled l in C. Now, one can recursively define the probability that u is reached $s > 1$ steps before any node labeled l as:

$$P(u,s) = \sum_{v \in Ent(C) \setminus L} P_{info}(v|u) \cdot P(v,s-1) \quad (5)$$

Intuitively, s represents the depth of the recursion. As shown in Algorithm 1, the above recursion can be computed in an iterative manner in time $O(|F(C)|)$.

Algorithm 1 *p2lEstimation*(C)

```
 1: $X := \{v | lab(v) = l\}$
 2: for all $v \in X$ do
 3:     $P_l^2(v) = \frac{1}{|X|}$
 4: end for
 5: $Y := \emptyset; U := Ent(C) \setminus L$
 6: while $U$ is not empty do
 7:     for all adjacent nodes $u, v$ with $u \in U, v \in X$ do
 8:         compute $P_l^2(u) = \sum_{v:lab(v)=l} P_{info}(v|u) P_l^2(v)$
 9:         insert $u$ into $Y$
10:     end for
11:     $U := U \setminus Y$
12:     $X := Y; Y := \emptyset$
13: end while
```

In lines 1 - 4 of Algorithm 1, all nodes in X (which are exactly the nodes labeled l) are assigned the same P_l^2 value

$\frac{1}{|X|}$. The set U (line 5) contains in each iteration (lines 6 - 13) all unlabeled nodes that have no P_l^2 value. In each iteration, we exclude from U (line 11) all nodes for which a P_l^2 value was determined during the iteration (represented by the set Y, line 5). At the end of each iteration the set X is set to Y. In lines 7 - 10, for all adjacent nodes u, v with $u \in U$ and $v \in X$ we compute $P_l^2(u)$ (line 8). The algorithm terminates when the set U is empty.

At this point, each node v of C has for each $l \in \{-,+\}$ a probability $P_l(v) = P_l^1(v) \cdot P_l^2(v)$. The label of each node in $v \in Ent(C)$ can now be easily determined by $lab(v) = \arg\max_{l\in\{-,+\}} P_l(v)$. Finally, the most informative subgraph of C is the one that consists of all nodes v for which $lab(v) = +$. In case this subgraph has more than b nodes, we successively remove from it the node v that does not belong to T and has minimal $P_+(v)$. By the construction of our mining method, it is easy to see that S fulfills the desired properties of Definition 3.

3. USER STUDY

Setting The focus of our evaluation has been on the user perceived quality of MING's answers. Therefore, in a user evaluation, we compared the answers of MING to those returned by CEPS [18]. As data set we used YAGO. The P_{info} edge weights (see Equation (1)) for YAGO were approximated on the Wikipedia corpus, based on co-occurrences of entity names in Wikipedia articles. In general, it is quite difficult for users to decide whether an ER graph that interconnects a given set of query entities is informative, because: (1) informativeness is an intuitive and also subjective notion, (2) a user's intuition has to be supported by the data in the underlying ER graph, and (3) a user needs to have very broad knowledge to assess the informativeness of a result graph for any set of given query nodes (especially when the query nodes represent rather obscure entities). Therefore, for this evaluation, we generated queries in which the query nodes represented famous individuals. YAGO is very rich in terms of famous individuals and contains plenty of interesting facts about them. In order to generate our queries, we extracted from the Wikipedia lists, a list of famous physicists, a list of famous philosophers, and a list of famous actors. From each of these lists we randomly generated 20 queries, each of them consisting of 2 or 3 query entities, resulting in a set of 60 queries in total. For each of the 60 queries, we presented the results produced by CEPS and MING (on the same subgraph C) to human judges (not familiar with the project) on a graph-visualization Web interface, without telling them which method produced which graph. For visualization purposes, the result graphs of CEPS and MING were pruned, whenever they had more than 15 nodes. By restricting the result graphs to such a small number of nodes, both methods were challenged to maintain only the most important nodes in the result graphs. CEPS comes with its own pruning parameter (i.e., visualization parameter). For each query, the users were given the possibility to decide which of the presented subgraphs they perceived as more informative. That is, one of the results could be marked *informative*. We also allowed users to mark both result graphs as *informative*, if they perceived them both as equally informative. Additionally, the results of both methods could be left unmarked, meaning that they both did not suit the user's intuition. The results are presented in Table 1.

	MING	**CEPS**
# times preferred over competitor	182	4
# times marked *informative*	185	7
# times both marked *informative*	3	
# times both left unmarked	21	

Table 1: Results of the user evaluation

Results There were 210 assessments in total, corresponding to more than 3 assessments per query. The result graphs produced by MING were marked 185 times as informative, and out of these, 182 times, they were perceived more informative than the results produced by CEPS. On the other hand, the MING results were left 25 times unmarked, and out of these, only 4 times they were perceived to be less informative than the results produced by CEPS.

The fundamental factor for the superiority of MING is its subgraph learning method. It learns informative and structurally robust paths between the nodes of an initial tree T that closely interconnects the query nodes.

4. CONCLUSION

The motivation for this work has been to provide new techniques for exploring and discovering knowledge in ER graphs. Our method, MING, is a significant step forward in this realm. It contributes to new semantic measures for the relatedness between entities. MING exploits such measures for extracting informative subgraphs that connect two or more given entities. The results of the user study fortify our assumption that MING indeed captures the intuitive notion of informative subgraphs in most of the cases.

5. REFERENCES

[1] S. Agrawal, S. Chaudhuri, and G. Das. DBXplorer: A system for keyword-based search over relational databases. In *Proc. of ICDE*, 2002.

[2] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using BANKS. In *Proc. of ICDE*, 2002.

[3] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee. Linked data on the web (ldow2008). In *WWW*, 2008.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, 1998.

[5] S. Chakrabarti. Dynamic personalized pagerank in entity-relation graphs. In *WWW*, 2007.

[6] B. Ding, J. Yu, S. Wang, L. Qing, X. Zhang, and X. Lin. Finding top-k min-cost connected trees in databases. In *Proc. of ICDE*, 2007.

[7] C. Faloutsos, K. S. McCurley, and A. Tomkins. Fast discovery of connection subgraphs. In *KDD*, 2004.

[8] D. Gibson, J. M. Kleinberg, and P. Raghavan. Inferring web communities from link topology. *HYPERTEXT '98*, 1998.

[9] H. He, H. Wang, J. Yang, and P. Yu. BLINKS: Ranked keyword searches on graphs. In *Proc. of SIGMOD*, 2007.

[10] V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient ir-style keyword search over relational databases. In *Proc. of VLDB*, 2003.

[11] V. Hristidis and Y. Papakonstantinou. DISCOVER: Keyword search in relational databases. In *Proc. of VLDB*, 2002.

[12] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar. Bidirectional expansion for keyword search on graph databases. In *Proc. of VLDB*, 2005.

[13] G. Kasneci, M. Ramanath, M. Sozio, F. M. Suchanek, and G. Weikum. STAR: Steiner-Tree Approximation in Relationship Graphs. In *ICDE 2009*. IEEE, 2009.

[14] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5), 1999.

[15] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Computer Networks*, 1999.

[16] A. Papadopoulos, A. Lyritsis, and Y. Manolopoulos. Skygraph: an algorithm for important subgraph discovery in relational graphs. *Data Mining and Knowledge Discovery*, 17(1), 2008.

[17] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A Large Ontology from Wikipedia and WordNet. *Elsevier Journal of Web Semantics*, 2008.

[18] H. Tong and C. Faloutsos. Center-piece subgraphs: problem definition and fast solutions. In *KDD*, 2006.

[19] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM*, 2006.