Title: Graph Classification in Heterogeneous Networks

Name: Xiangnan Kong[1], Philip S. Yu[1]

Affil./Addr.: Department of Computer Science

University of Illinois at Chicago

Chicago, IL, USA

E-mail: {xkong4, psyu}@uic.edu

Graph Classification in Heterogeneous Networks

Synonyms

heterogeneous networks, graph classification, meta-path

Glossary

HIN: Heterogeneous information network.

Definition

Information networks have been intensively studied in recent years, ranging from community detection to graph classification. Typical applications of information networks include web mining, social network analysis, bioinformatics, *etc.* Most previous research on information networks focuses on homogeneous networks, which involve one type of nodes and one type of links, e.g., social networks with friendship links, webpage networks with hyper-links. With the recent advance in data collection techniques, many real-world applications are facing large scale heterogeneous information networks [12], which involve multiple types of objects inter-connected through multiple types links.

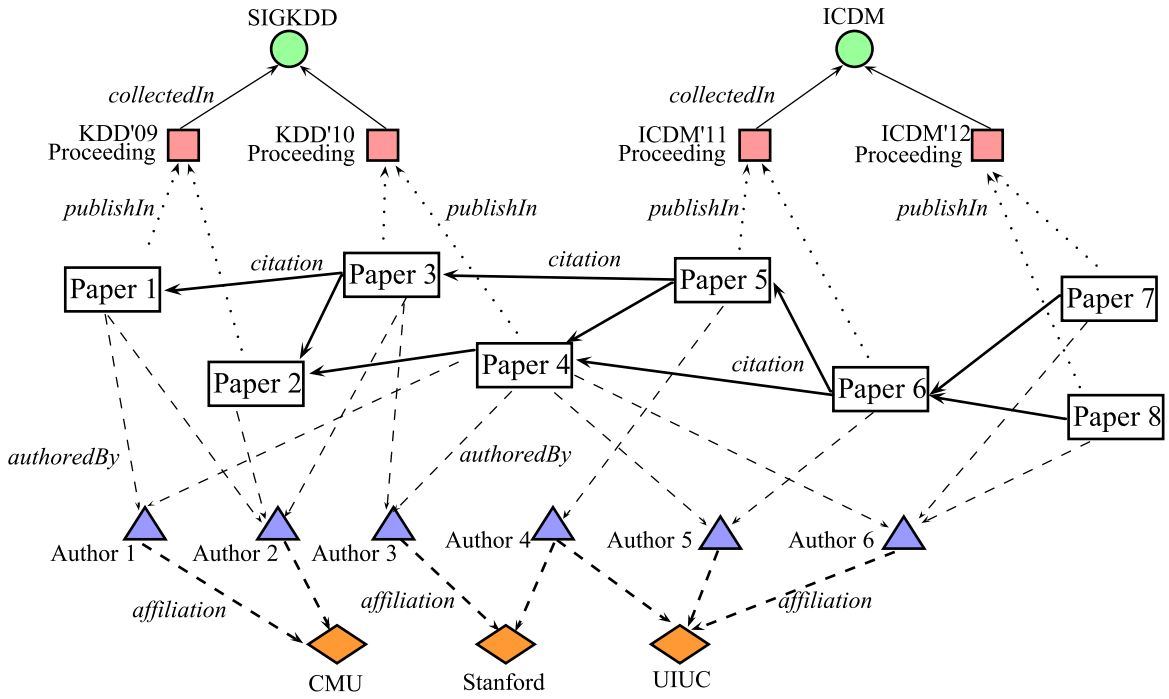


Fig. 1. An Example of Heterogeneous Information Network: bibliographic network

These networks are multi-mode and multi-relational networks, which involves large amount of information. For example, a bibliographic network in Figure 1 involves five types of nodes (papers, author, affiliations, conference and proceedings) and five types of links. This heterogeneous information network is more complex and contain more linkage information than its homogenous sub-network, *i.e.*, a paper network with only citation links.

Definition 1. **Heterogeneous Information Network**: A heterogeneous information network [14, 12] is a special kind of information network, which is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is the set of nodes, including t types of objects $\mathcal{T}_1 = \{v_{11}, \cdots, v_{1n_1}\}, \cdots, \mathcal{T}_t = \{v_{t1}, \cdots, v_{tn_t}\}$. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of links between the nodes in $\mathcal{V}$, which involves multiple types of links.

Example 1. **ACM conference network**: A heterogeneous information network graph is provided in Figure 1. This network involves five types of objects, *i.e.*, papers (P), authors (A), institutes (F), proceedings (V) and conferences (C), and five types of links, *i.e.*, *citation*, *authoredBy*, *affiliation*, *publishedIn* and *collectedIn*.

Graph classification in information networks have been studied intensively in the last decade. The task is to exploit the linkage information in networks to improve classification accuracies on the nodes. Different from conventional supervised classification approaches that assume data are independent and identically distributed, graph classification methods aim at exploiting the label autocorrelation among a group of inter-connected nodes and predict their class labels collectively, instead of independently. In many network data [15, 2], the nodes are inter-related with complex dependencies. For example, in bibliographic networks, if two papers both cite (or are cited by) some other papers (*i.e.*, bibliographic coupling or co-citation relationship) or one paper cites the other (*i.e.*, citation relationship), they are more likely to share similar research topics than those papers without such relations. These dependencies among the related nodes should be considered explicitly during classification process. Motivated by these challenges, graph classification problem in network data has received considerable attention in the literature [9, 15, 6].

The major research challenges for graph classification in heterogeneous networks can be summarized as follows:

Multi-Mode and Multi-Relational Data: One fundamental problem in classifying heterogeneous information networks is the complex network structure that involves multiple types of nodes and multiple types of links. For example, in Figure 1, one paper node can be linked *directly* with different types of objects, such as authors, conference proceedings and other papers, through different types of links, such as *citation*, *authoredBy*, *etc.* Different types of links have totally different semantic meanings. Trivial application of conventional methods by ignoring the link types and node types can not fully exploit the structural information within a heterogeneous information network.

Heterogeneous Dependencies: Another problem is that objects in heterogeneous information networks can be linked *indirectly* through different types of relational paths. Each types of relational path corresponds to different types of *indirect* relationships between objects. For example, in Figure 1, paper nodes can be linked with each other indirectly through multiple *indirect* relationships, such as, 1) the "paper-author-paper" relation indicates relationships of two papers sharing same authors; 2) the "paper-author-institute-author-paper" relation denotes relationship between papers that are published from the same institute. Heterogenous information networks can encode various complex relationships among different objects. Thus, ignoring or treating all relations equally will loss information dependence information in a heterogeneous information network. Exploring such heterogeneous structure information has been shown useful in many other data mining tasks, such as ranking [5, 4], clustering [13, 14] and classification tasks [3].

Historical Background

Heterogeneous information networks are special kinds of information networks which involve multiple types of nodes or multiple types of links. In a heterogeneous information network, different types of nodes and edges have different semantic meanings. The complex and semantically enriched network possesses great potential for knowledge discovery. In the data mining domain, heterogeneous information networks are ubiquitous in many applications, and have attracted much attention in the last few years [14, 13, 3]. Sun et al. [14, 12] studied the clustering problem and top-k similarity problem in heterogeneous information networks. Ming et al. studied a specialized classification problem on heterogeneous networks, where different types of nodes share a same set of label concepts [3].

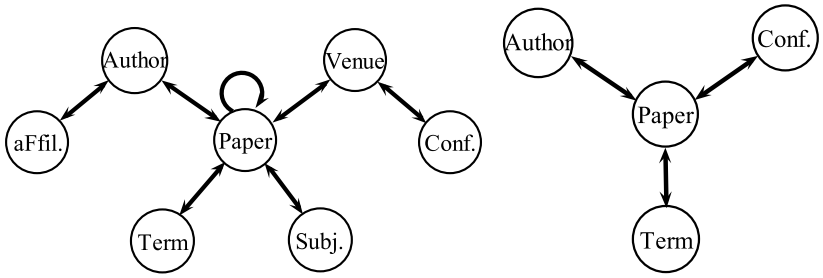
Graph classification in network data has been investigated by many researchers. The task is to predict the classes for a group of related instances simultaneously, rather than predicting a class label for each instance independently. In relational datasets, the class label of one instance can be related to the class labels (sometimes attributes) of the other related instances. Conventional graph classification approaches focus on exploiting the correlations among the class labels of related instances to improve the classification performances. Roughly speaking, existing graph classification approaches can be categorized into two types based upon the different approximate inference strategies: (1) Local methods: The first type of approaches employ a local classifier to iteratively classify each unlabeled instance using both attributes of the instances and relational features derived from the related instances. This type of approaches involves an iterative process to update the labels and the relational features of the related instances, *e.g.* iterative convergence based approaches [9, 6] and Gibbs sampling approaches [8]. Many local classifiers have been used for local methods, *e.g.* logistic regression [6], Naive Bayes [9], relational dependency network [10], *etc.* (2) Global methods: The second type of approaches optimizes global objective functions on the entire relational dataset, which also uses both attributes and relational features for inference [15]. For a detailed review of graph classification please refer to [11].

Graph Classification in Heterogeneous Networks

Different from conventional networks, heterogeneous information networks involve different types of objects (e.g., papers and conference) that are connected with each other through multiple types of links. Each type of links represents an unique binary relation R from node type i to node type j, where $R(v_{ip}, v_{jq})$ holds iff object v_{ip} and v_{jq} are related by relation R. R^{-1} denotes the inverted relation of R, which holds naturally for $R^{-1}(v_{jq}, v_{ip})$. Let $dom(R) = \mathcal{T}_i$ denote the domain of relation R, $rang(R) = \mathcal{T}_j$ denotes

Table 1. Important Notations.

Symbol	Definition
$\mathcal{V} = \bigcup_{i=1}^{t} \mathcal{T}_i$	the set of nodes, involving t types of nodes
$\mathcal{E} = \{e_i \in \mathcal{V} \times \mathcal{V}\}$	the set of edges or links
$\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{n_1}\}$	the given attribute values for each node in target type $\mathcal{T}_1$
$\mathcal{Y} = \{Y_1, \cdots, Y_{n_1}\}$	the set of variables for labels of the nodes in $\mathcal{T}_1$, and $Y_i \in \mathcal{C}$
$\mathcal{L}$ and $\mathcal{U}$	the sets for training nodes and testing nodes, and $\mathcal{L} \cup \mathcal{U} = \mathcal{T}_1$
y_i	the given label for node $v_{1i} \in \mathcal{L}$, and $Y_i = y_i$
$\mathcal{S} = \{\mathcal{P}_1, \cdots, \mathcal{P}_m\}$	the set of meta paths
$\mathcal{P}_j(i) = \{k \vert \mathcal{P}_j(v_{1i}, v_{1k})\}$	the index set of all related instances to $\boldsymbol{x}_i$ through meta path $\mathcal{P}_j$



(a) ACM data (b) DBLP data

Fig. 2. Examples of bibliographic network schema.

its range. $R(a) = \{b : R(a, b)\}$. For example, in Figure 1, the link type "*authorBy*" can be written as a relation R between paper nodes and author nodes. $R(v_{ip}, v_{jq})$ holds iff author v_{jq} is one of the authors for paper v_{ip}. For convenience, we can write this link type as "$paper \xrightarrow{authoredBy} author$" or "$\mathcal{T}_i \xrightarrow{R} \mathcal{T}_j$".

Different from homogeneous networks, two objects in a heterogeneous network can be connected via different paths and these paths have different meanings. For example, in Fig. 2(a), conferences and authors can be connected via "Author-Paper-Venue-Conference" ($APVC$) path, "Author-Paper-Subject-Paper-Venue-Conference" ($APSPVC$) path, and so on. It is clear that semantics underneath these paths are different. The $APVC$ path means that papers written by authors are published in con-

Table 2. Semantics of Meta Paths among Paper Nodes

	Notation	Meta Path	Semantics of the Dependency
1	P→P	Paper $\xrightarrow{cite}$ Paper	Citation
2	P←P→P	Paper $\xrightarrow{cite^{-1}}$ Paper $\xrightarrow{cite}$ Paper	Co-citation
3	P→P←P	Paper $\xrightarrow{cite}$ Paper $\xrightarrow{cite^{-1}}$ Paper	Bibliographic coupling
4	PVP	Paper $\xrightarrow{publishIn}$ Proceeding $\xrightarrow{publishIn^{-1}}$ Paper	Papers in the same proceeding
5	PVCVP	Paper $\xrightarrow{publishIn}$ Proceeding $\xrightarrow{collectIn}$ Conference $\xrightarrow{collectIn^{-1}}$ Proceeding $\xrightarrow{publishIn^{-1}}$ Paper	Papers in the same conference
6	PAP	Paper $\xrightarrow{write^{-1}}$ Author $\xrightarrow{write}$ Paper	Papers sharing authors
7	PAFAP	Paper $\xrightarrow{write^{-1}}$ Author $\xrightarrow{affiliation}$ Institute $\xrightarrow{affiliation^{-1}}$ Author $\xrightarrow{write}$ Paper	Papers from the same institute

ferences, while the *APSPVC* path means that papers having the same subjects as the authors' papers are published in conferences. In order to categorize these paths, we extend the definition of link types to "path types", which are named as *meta path*, similar to [12, 5].

Definition 2. **Meta Path**: A meta path $\mathcal{P}$ represents a sequence of relations $R_1, \cdots, R_\ell$ with constrains that $\forall i \in \{1, \cdots, \ell-1\}, rang(R_i) = dom(R_{i+1})$. The meta path $\mathcal{P}$ can also be written as $\mathcal{P} : \mathcal{T}_1 \xrightarrow{R_1} \mathcal{T}_2 \xrightarrow{R_2} \cdots \xrightarrow{R_\ell} \mathcal{T}_{\ell+1}$, *i.e.*, $\mathcal{P}$ corresponds to a composite relation $R_1 \times R_2 \times \cdots \times R_\ell$ between node type $\mathcal{T}_1$ and $\mathcal{T}_{\ell+1}$. $dom(P) = dom(R_1)$ and $rang(P) = rang(R_\ell)$. The length of P is ℓ, *i.e.*, the number of relations in $\mathcal{P}$.

Different meta paths usually represent different semantic relationships among linked objects. In Table 2, we show some examples of meta paths with their corresponding semantics. Most conventional relationships studied in network data can naturally be captured by different meta paths. For example, the paper *co-citation* relation [1] can naturally be represented by meta path "*paper* $\xrightarrow{cite^{-1}}$ *paper* $\xrightarrow{cite}$ *paper*", and the co-citation frequencies can be written as the number of path instances for the meta path. Here a *path instance* of $\mathcal{P}$, denoted as $p \in \mathcal{P}$, is an unique sequence of nodes and links in the network that follows the meta path constrains. For convenience, we use the node type sequence to represent a meta path, *i.e.*, $\mathcal{P} = \mathcal{T}_1\mathcal{T}_2 \cdots \mathcal{T}_{l+1}$. For example, we use PAP to represent the meta path "*paper* $\xrightarrow{authoredBy}$ *author* $\xrightarrow{authoredBy^{-1}}$ *paper*".

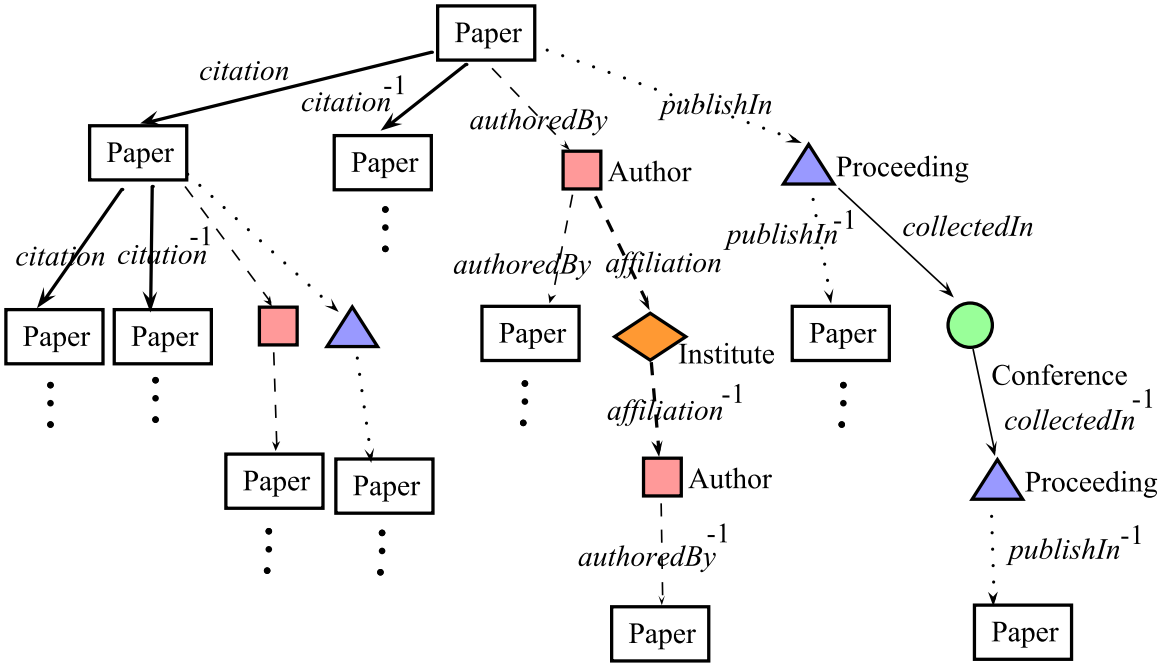


Fig. 3. An example of dependence tree for meta path-based dependencies. Each paper node corresponds to a unique type of path-based dependencies in the network.

Note that for meta paths involving *citation* links, we explicitly add arrows to represent the link directions, *e.g.*, the paper *co-citation* path can be written as $P \leftarrow P \rightarrow P$.

Meta Path-based Graph Classification

For classifying target nodes in a heterogeneous information network, the most naïve approach is to assume that all instances (*e.g.*, the paper nodes) are independent from each other. However, this approach can be detrimental to their performance for many reasons. This is particularly troublesome when nodes in heterogeneous networks have very complex dependencies with each other through different meta paths.

In this section, we discuss a simple and effective algorithm for meta path-based graph classification in heterogeneous information networks.

Step 1. We first consider how to extract all meta paths in a heterogeneous information network of bounded length ℓ_{max}. When ℓ_{max} is small, we can easily generate all possible meta paths as follows: We can organize all the type-correct relations into

a prefix tree, called *dependence tree*. In Figure 3, we show an example of dependence tree in ACM conference networks. The target nodes for classification are the paper nodes, and each paper node in the dependence tree corresponds to an unique meta path, indicating one type of dependencies among paper instances. However, in general the number of meta paths grows exponentially with the maximum path length ℓ_{max}. As it has been showed in [12], long meta paths may not be quite useful in capturing the linkage structure of heterogeneous information networks. In this paper, we only exploit the instance dependences with short meta paths ($\ell_{max} = 4$).

In many really world network data, exhaustively extracting all meta paths may result in large amount of redundant meta paths, *e.g.*, $PVPVP$. Including redundant meta paths in a classification model can result in overfitting risks, because of additional noisy features. Many of the redundant meta paths are constructed by combining two or more meta paths, *e.g.*, meta path $PVPVP$ can be constructed by two PVP paths. In order to reduce the model's overfitting risk, we extract all meta paths that cannot be decomposed into shorter meta paths (with at least one *non-trivial* meta paths). Here non-trivial meta paths refer to the paths with lengths greater than 1. For example, in ACM conference network, meta paths like $P{\rightarrow}PAP$ can be decomposed into $P{\rightarrow}P$ and PAP, thus will be excluded from our meta path set. We refer the meta path set extract process as the "Initialization" step of our method. By breadth-first search on the dependence tree, our model first select shortest meta paths from the network. Then longer meta paths are incrementally selected into path set $\mathcal{S}$ until we reach a meta path that can be decomposed into shorter meta paths in $\mathcal{S}$.

Step 2. After the meta path set $\mathcal{S}$ is extracted from the heterogeneous information network, we then show how to use these meta paths to perform classification effectively. Conventional graph classification based on iterative inference process, *e.g.* ICA (Iterative Classification Algorithm) [9, 6], provide a simple yet very effective method

for graph classification in homogeneous networks. Inspired by the success of these iterative inference methods, we discuss a similar framework for meta path-based graph classification method.

The general idea is as follows: we model the joint probability based on the following assumption: if instance v_{1i} and v_{1j} are not connected via any meta path in $\mathcal{S}$, the variable Y_i is conditional independent from Y_j given the labels of all v_{1i}'s related instances, *i.e.*, $\{v_{1j}|j \in \bigcup_{k=1}^{m} \mathcal{P}_k(i)\}$. Hence the local conditional probability each instance's label can be modeled by a base learner with extended *relational features* built upon the predicted Y_j's ($j \in \bigcup_{k=1}^{m} \mathcal{P}_k(i)$). And the joint probability can be modeled based on these local conditional probabilities by treating the instances as being independent.

In graph classification, each instance may be linked with different number of instances through one meta path. In order to build a fixed number of relational features for each instance, we employs *aggregation* functions to combine the predictions on the labels of related instances. Many aggregation functions can be used here, such as COUNT and MODE aggregators [6]. In this paper, we use the *weighted label fraction* of the related instances as the relational feature for each meta path. We calculate the average fraction of each label appearing in the related instances. Each related instance in re-weighted by the number of path instances between from the current node, e.g., for meta path PAP, the papers that share more authors in their author lists are more likely to share similar topics than those only share one author. In detail, given an aggregation function, we can get one set of relational features from the labels of related instances for each meta path.

Step 3. Inspired by the success of ICA framework [6, 7, 8] in network classification, we designed a similar inference procedure. (1) For inference steps, the labels of all the unlabeled instances are unknown. We first *bootstrap* an initial set of label

estimation for each instance using content attributes of each node. In our current implementation, we simply set the relational features of unlabeled instances with zero vectors. Other strategies for *bootstrap* can also be used in this framework. (2) *Iterative Inference*: we iteratively update the relational features based on the latest predictions and then these new features are used to update the prediction of local models on each instance. The iterative process terminates when convergence criteria are met. In our current implementation, we update the variable Y_i in the $(r+1)$-th iteration (say $\hat{Y}_i^{(r+1)}$) using the predicted values in the r-th iteration ($\hat{Y}_j^{(r)}$) only.

Cross-References

Bibliometrics, Network Analysis, and Knowledge Generation

Classical Algorithms for Social Network Analysis: Future and Current Trends

Collective Classification

Collective Classification, Structural Features

Data Mining

Sources of Network Data

Social Network Datasets

References

1. Y. Ding, E. Yan, A. Frazho, and J. Caverlee. PageRank for ranking authors in co-citation networks. *Journal of the American Society for Information Science and Technology*, 60(11):2229–2243, 2009.
2. J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. Community outliers and their efficient detection in information networks. In *KDD*, pages 913–822, 2010.
3. M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *KDD*, pages 1298–1306, San Diego, CA, 2011.
4. N. Lao and W. Cohen. Fast query execution for retrieval models based on path-constrained random walks. In *KDD*, pages 881–888, Washington, DC, 2010.

5. N. Lao and W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(2):53–67, 2010.
6. Q. Lu and L. Getoor. Link-based classification. In *ICML*, pages 496–503, 2003.
7. L. K. McDowell, K. M. Gupta, and D. W. Aha. Cautious inference in collective classification. In *AAAI*, pages 596–601, 2007.
8. L. K. McDowell, K. M. Gupta, and D. W. Aha. Cautious collective classification. *Journal of Machine Learning Research*, 10:2777–2836, 2009.
9. J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI'10 Workshop on Learning Statistical Models from Relational Data*, 2000.
10. J. Neville and D. Jensen. Collective classification with relational dependency networks. In *KDD'03 Workshop on Multi-Relational Data Mining*, pages 77–91, 2003.
11. P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.
12. Y. Sun, J. Han, X. Yan, P. Yu, and T. Wu. PathSim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, Seattle, WA, 2011.
13. Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. RankClus: integrating clustering with ranking for heterogeneous information network analysis. In *EDBT*, pages 565–576, Saint-Petersburg, Russia, 2009.
14. Y. Sun, Y. S. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, pages 797–806, Paris, France, 2009.
15. B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, pages 482–492, 2002.