Prominence Ranking in Graphs
with Community Structure

Sibel Adalı
CS Department
Rensselaer Polytechnic Inst.
sibel@cs.rpi.edu

Xiaohui Lu
CS Department
Rensselaer Polytechnic Inst.
lux3@cs.rpi.edu

Malik Magdon-Ismail
CS Department
Rensselaer Polytechnic Inst.
magdon@cs.rpi.edu

Jonathan Purnell
CS Department
Rensselaer Polytechnic Inst.
purnej@cs.rpi.edu

Introduction

We study prominence ranking in heterogeneous social networks with actors who collaborate to create artifacts which display some homophily based community structure (Figure 1). For example, a paper is an artifact and multiple authors may collaborate on the paper. Papers appear in venues, which are communities containing papers on similar topics and of *similar quality*. An artifact conferes a social tie on its contributors, and homophily based groups of artifacts serves to identify the artifacts of similar stature. Past research on prominence has only used the social ties, ignoring the relationships between the artifacts (two artifacts with common authors are linked). We incorporate the artifact relationships [1].

Artifacts can belong to natural groupings. For example, conference venues have a topic and a prominence – papers in high impact conferences tend to be from prominent authors. This is true because of rigorous review cycles as well as self selection of submissions. Thus, artifact groupings convey valuable information on quality. On the other hand, conference venues can be a noisy source of information, containing peripherally related topics and papers of a wide range in quality. One needs robust measures to obtain information from the artifact groupings. We use the DBLP network of scientific publications to illustrate our algorithms, and our validation will be using citation counts (an external measure of prominence). We show that our algorithm, which considers the artifacts as well as the artifact groupings, provides a clear improvement over other well-known ranking methods.

What if artifact groupings, such as conference venues are not available? In this case, using the link structure between artifacts, we use clustering algorithms to find groupings for the artifacts (Clauset, Newman, and Moore 2004; Magdon-Ismail and Purnell 2010; Sun et al. 2009). Note
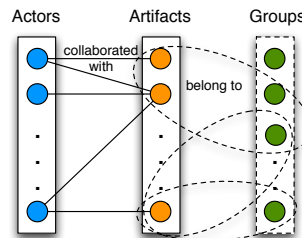
Figure 1: An example object graph

that, instead of clustering people, we cluster their artifacts to capture both the common topics and prominence of artifacts. For DBLP, we show that our clusters can be used as a substitute for "real" groups (conference venues). In fact, we show that a very efficient embedding based algorithm for clustering (Magdon-Ismail and Purnell 2010) produces paper groupings that perform *as well as or better than real venues*. The clustering is also more robust to the aforementioned noise that can be present in the real venues. We are able to show that when artifacts constitute a social tie between people, we can use the relationships between artifacts to improve the computation of prominence in social networks.

Prominence ranking has received significant attention in the literature (Faust and Wasserman 1994), with methods ranging from: centrality measures (Borodin et al. 2005); heterogeneous weights for heterogeneous links (Hotho et al. 2006; Balmin, Hristidis, and Papakonstantinou 2004), navigation based measures which take into account topic or trust (Haveliwala 2002; Gyongyi, Garcia-Molina, and Pedersen 2004). Our work, in contrast, ranks actors according to the artifacts they are associated with, and how groupings of the artifacts improves the ranking. Methods for analysing scientific literature typically consider the links between researchers based on the papers they have co-authored (Liu et al. 2005) instead of the linkages between the papers. RankClus (Sun et al. 2009) partitioning the actors ranking within each partition separately based on the claim that it does not make sense to compare actors from different communities into a single ranking. We do not dispute this fact. Instead, we augment it by showing that information from groupings of artifacts can be used to get a better ranking for

the actors themselves. Our contributions:

1. A framework for ranking in an (actor)-(artifact)-(artifact group) tri-partite social network (Figure 1). Our methods are applicable to collaborative social networks, and can be fine tunded to a particular dataset to tradeoff robustness with accuracy.

2. Improved ranking performance using the artifact groups as compared with ignoring these groups. We demonstrate on by using conference venues to significantly improve the ranking of DBLP authors, with ground truth defined using citation counts.

3. A novel algorithm which infers artifact clusters and can be used even when no natural artifact groupings are available. For DBLP, the infered clusters perform better than or as well as the "ground truth" conference venues.

Computing Prominence

Consider the tripartite graph $G = (V, E)$ in Figure 1; each node v has a type, $type(v)$. We consider three types: actors (A), artifacts (B), artifact groups (C), where the only edges are between nodes of type A and B, and B and C. Intuitively, A's are authors, B's are papers and C's are venues.

Let V_A, V_B, V_C denote the nodes of each type. The relationships between V_A and V_B, and V_B and V_C are many to many (actors can be associated with many artifacts, and artifacts can be associated with many actors; artifacts can belong to multiple groups and groups can have many artifacts). If the artifact groups are conferences, then they are disjoint. But, if the groups correspond to research topics, then a paper may belong to multiple groups. Assume the vertices in each of V_A, V_B, V_C are numbered from 1 to $|V_A|, |V_B|, |V_C|$ respectively. We compute scores for each node $a \in V_A$, $b \in V_B$ and $c \in V_C$, denoted by v_a, v_b and v_c.

We use a parametrized framework for computing the scores, in which the scores of nodes of type A,B and C depend on each other in a circular way (see Figure 2). The algorithm is specified through the functions f_1, f_2, f_3, f_4 (see the pseudo-code for Algorithm 1). Artifacts (papers) derive prominence from their actors (authors) and artifact groups (conferences) derive prominence from their artifacts. To close the loop, artifacts derive prominence by being in prominent groups and actors derive prominence by association with prominent artifacts. This framework provides us with a set of simultaneous equations to satisfy (given f_1, f_2, f_3, f_4); the unknowns are the $\{v_a, v_b, v_c\}$. When the algorithm converges, it provides a solution to the given equations. Some examples of functions f_i are given below.

- f_1: Artifact score depends on the scores of associated actors (good authors create good papers). Examples are: avg (each actor contributes equally); max (an artifact is as good as its best associated actor); sum (artifacts associated to more actors are better).

- f_2: The score of an artifact group depends on the scores of member artifacts (good papers are in good conferences). Examples are: avg (each member artifact contributes equally); $T_{(k,avg)}$ is the average of the best k% and worst k% of member artifacts.

$$
\begin{aligned}
v_b &= f_1(v_a) \\
v_c &= f_2(v_b) \\
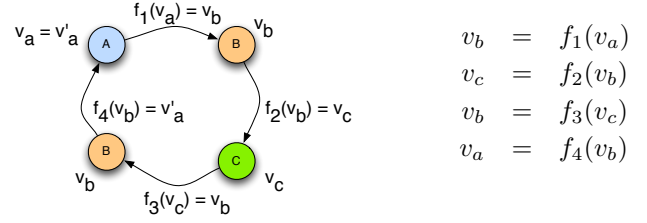v_b &= f_3(v_c) \\
v_a &= f_4(v_b)
\end{aligned}
$$

Figure 2: System of equations for computing prominence

- f_3: Artifact score depends on the groups it belongs to (good conferences contain good papers). An examples is avg (average score of groups containing the artifact).
- f_4: Actor score depends on the scores of associated artifacts (authors of prominent papers are prominent). Examples are: sum (more artifacts are better); sum_2: normalize artifact score by number of actors and sum (more artifacts with few co-actors is better); $T_{(k,sum)}$ (and $T_{(k,sum2)}$) use only the best and worst $k\%$ associated artifacts (prominent actors have some important artifacts and few dudds).

Algorithm 1 PAG $(G = (V, E))$

1: Input: Tri-partite $G = (V, E)$ with node types A, B, C.
2: For $a \in V_A$, let $v_a = degree(a) / \sum_{a \in V_A} degree(a)$.
3: **while** not converged **do**
4:     **for all** $b \in V_B$ **do** $v_b = f_1(\{v_a \mid (a,b) \in E\})$
5:     normalize v_b values to sum up to 1
6:     **for all** $c \in V_C$ **do** $v_c = f_2(\{v_b \mid (b,c) \in E\})$
7:     normalize v_c values to sum up to 1
8:     **for all** $b \in V_B$ **do** $v_b = f_3(\{v_c \mid (b,c) \in E\})$
9:     normalize v_b values to sum up to 1
10:    **for all** $a \in V_A$ **do** $v_a = f_4(\{v_b \mid (a,b) \in E\})$
11:    normalize v_a values
12: **end while**
13: return $\{v_a\}, \{v_b\}, \{v_c\}$

Our algorithm is similar in formulation to the Hits algorithm (Kleinberg 1999), except for a tri-partite graph; we compute the scores of each type of node separately. The ranking performance can depend on the choice of the functions, and this choice can benefit from domain knowledge. For example, a group may contain many artifacts, but we might only care about the quality of the top artifacts in evaluating the group. We introduce the T_k method. Let X be the set of scores of the papers in a conference; $top(X, k)$ and $bottom(X, k)$ are the average score of the top and bottom $k\%$ scores in X. Then, $T_{(k,f)}(X) = \lambda \cdot top(X, k) + (1 - \lambda) \cdot bottom(X, k)$. When $\lambda < 1$, the best score asks for good scores in X with low deviations.

Inferring Groups

When external artifact groups are not known, we infer groups using clustering. For $b \in V_B$, let $A(b)$ be the set of actors associated with b. Construct the artifact similarity

graph $G^*(V_B, E^*)$ on vertex set V_B with $(b_1, b_2) \in E^*$ iff $A(b_1) \cap A(b_2) \neq \emptyset$; in this case, we give the edge a weight (distance) $w(b_1, b_2) = |A(b_1) \cup A(b_2)|/|A(b_1) \cap A(b_2)|$.

We compare results with different clustering algorithms: (1) FC: FastCommunity (Clauset, Newman, and Moore 2004) finds densely connected subgroups of the graph and produces disjoint clusters of varying sizes. (2) DE: Distance Embedding, SSDE-Clustering (Magdon-Ismail and Purnell 2010) based on metric embeddings followed by Gaussian Mixture Model clustering to find n (an input) of overlapping clusters. (3) RC: RankClus (Sun et al. 2009) takes the dataset a with predefined venues and partitions these venues into n disjoint clusters where n is an input to the algorithm.

Clusters from FC and DE capture two types of information: common topics and collaborators. Clusters from RC correspond to common groupings of existing venues into topics. DE allows a paper to belong to multiple groups, which none of the other methods do.

Experimental Setup

To test our prominence framework, we used the RDF tagged version DBLP from August 2007 called SwetoD-BLP(lsdis.cs.uga.edu/projects/semdis/swetodblp/) containing publications from Computer Science (books, conferences and journals). The data contains 1,004,959 publications (book chapters, journal articles and conference papers). which we partition into 5 subsets:

	Description	Size
ALL	All publications in all venues	1,004,959
III	Books & conference w/ proceedings	495,159
C	All conferences	611,910
J	All Journals	382,499
M	Conferences merged across years	993,651

In the first four subsets, every year of a conference is considered a different venue. To test the ranking performance, we used the citation record of authors for a sample of 89K authors, gathered using Google Scholar. We considered two measures: (1) *h-index* is the largest x for which the author has x papers with at least x citations each (Hirsch 2005). (2) *tc-10* measures the average number of citations of the top 10 most cited papers. The tc-10 measure is useful for capturing the authors who are not prolific but impactful. We used four measures to compare the performances of various ranking algorithms. The first two measure the prominence of the top 20 ranked actors. The second compares the full rank ordering with the ground truth as defined by the citation based measures.

Name	Description
avgh	Average h-index of Top20
avgt	Average tc-10 of Top20
kth	Kendall-τ between h-index & algorithm ranking
ktt	Kendall-τ between tc-10 & algorithm ranking

For all measures, higher is better. The Kendall-τ correlation is for all 89K authors: for two rankings, it is number of pairs with the same ordering minus the number of pairs with flipped ordering, all divided by the total number of pairs. A Kendall-τ of 1 corresponds to identical orderings whereas -1 corresponds to reversal. Note that we have

Dataset: ALL

Algorithm	avgh	avgt	kth	ktt
Indegree	44.5	385.58	0.44	**0.39**
Pagerank	44.3	435.5	0.36	0.33
PAG/NoGroups	47.3	534	**0.45**	**0.39**
PAG	**48**	**695**	**0.45**	**0.39**

Dataset: III

Algorithm	avgh	avgt	kth	ktt
Indegree	36.75	319	**0.4**	**0.35**
Pagerank	42.5	352	0.32	0.29
PAG/NoGroups	35.4	421	**0.4**	**0.35**
PAG	**56**	**876**	**0.4**	**0.35**

Figure 3: The performance of different algorithms

run tests to determine the significance of Kendall-τ using randomly ordered nodes, and the deviation from zero is of order 0.003. The measures *avgh* and *avgt* capture how well the algorithm does in finding the top ranks, while *kth* and *ktt* capture how well the algorithm performs overall. To our knowledge, there is no study that uses DBLP with such an extensive evaluation against the ground truth.

We also compare with benchmark algorithms that use the author-author graph with weights equal to the number of common papers. We tested: INDEGREE; PAGERANK (eigenvalue centrality with $\alpha = 0.85$) and our algorithm (PAG-NOGROUP) without any artifact groups.

Results

Our first goal is to show (using DBLP) that venue information (artifact groupings) are useful (Figure 3). We report the best case behavior of our PAG algorithm based on the different functions we have tested (approximately 80 test cases). In our tests, indegree outperforms pagerank; PAG outperforms all the algorithms for capturing top people (*avgh*, *avgt*) when venue information is incorporated but indegree is competitive with our algorithm for *ktt* and *kth*. It means that for a suitable choice of functions $f_1, \ldots, f_4$, venue information can add significant value.

Next, we observe that the usefulness of the venue information (as measured by the performance of the PAG algorithm) varies significantly across different datasets (Figure 5). Some venue types are more noisy than others. Overall, conference proceedings appear to be the best and the most robust data set which makes sense as in the Computer Science fields represented in this dataset – conferences are typically more important than journals. Adding journals or merging conference years has a negative impact for capturing the top actors. Furthermore, we note that the best functions for each data set tend to differ, but in all test cases we study, the top-k function $T_{(k,f)}$ appears to outperform sum and average especially for large sets.

Our main goal is to study what happens when you don't have access to the artifact groupings (conference venues in this case), and have to infer them with clustering (Figure 4). In particular, we find that since the REAL venues are noisy, the performance using clustering is better than that using the real venues in almost all cases and accross all metrics.

Groupings	*avgh*			*avgt*			*kth*			*ktt*		
	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
FC	26.7	19.8	**53.3**	307	224	661	0.35	0.26	0.41	0.33	0.24	0.38
DE10_10	43.49	39.7	45.85	460	384	510	**0.43**	0.35	**0.46**	**0.39**	0.32	**0.41**
DE50	**47.56**	**43.65**	**53.3**	**576**	**396**	**737**	0.41	**0.37**	0.43	0.38	**0.35**	**0.41**
RC	15.6	5.67	46	120	25	469	0.39	0.28	0.43	0.35	0.26	0.38
REAL	13.8	4.92	48	124	12.13	695	0.38	0.33	0.42	0.35	0.31	0.39

Figure 4: Performance of the PAG algorithm using different clustering algorithms vs. real venues for all venues.

Dataset	*avgh*	*avgt*	*kth*	*ktt*
ALL	22.7	284	0.38	0.34
III	34.6	403	0.35	0.31
C	**43.5**	**502**	0.38	0.35
J	18.6	240	0.29	0.27
C+J	20.5	256	0.37	**0.37**
M	27.6	224	**0.40**	0.36

Figure 5: Average behavior of PAG across different data sets

Furthermore, the performance is more robust across different choices for the ranking model using clusters than real venues (as can be verified by looking at the worst case performance). This is especially true for the DE algorithm.

In our tests, FC produced 2737 clusters of sizes beween 2 and 111,628. Due to memory restrictions, we ran RC for authors with more than 9 papers and grouped them into 20 clusters. For DE, we used two settings: (1) DE_50: Cluster the main data set into 50 clusters. (2) DE_10_10: Cluster the data set into 10 main clusters, then within each cluster find 10 subclusters, resulting in a total of 100 clusters. The main clusters corresponds to the specific research areas and each subcluster is customized to the underlying graph induced within the main research area; one might argue that this is a better representation of the grouping structure within the artifacts (papers in this case).

We also note that the DE_10_10 gives the highest Kendall-tau performance of any test case and any subset of DBLP we studied. We conclude that not only clusters may be substituted for real venues, it also reduces the necessity of tuning the algorithm considerably. Our hypothesis is that clusters are not as noisy as real venues, hence robustness and tuning is not as crucial.

Conclusions

We developed a novel algorithm for ranking in a social network where the "semantics of the ties" can add significant value. The main message is that when the social tie between actors is inferred by their participation in some artifact, the properties and relations between those artifacts can significantly improve the ranking (as opposed to only using the social ties among the actors). This is especially true when the artifacts display strong grouping or community structure.

One important aspect of our algorithm is that it does not need to know how the artifacts are organized into groups, because these artifact groups can be inferred. In fact performance was enhanced by the more robust organization of the artifacts deduced using the overlapping clustering algorithm in (Magdon-Ismail and Purnell 2010).

Many systems rely on ratings of information to build a credibility score for users. However, in many cases, there is no social link between the raters. By making the rating scheme collaborative (e.g. through the use of budgets) and organizing the rated artifacts into groups, our algorithm can be used to provide better credibility scores.

Several avenues require further investigation: How do different choices for the functions f_1, f_2, f_3, f_4 relate to the type of network? Should all the artifact-groups be used in the algorithm? One might conjecture that only the "important" groups should be used, which is similar to the notion of trust-rank, where the structure of the graph with respect to important nodes can be more robust (Gyongyi, Garcia-Molina, and Pedersen 2004).

References

Balmin, A.; Hristidis, V.; and Papakonstantinou, Y. 2004. Objectrank: Authority-based keyword search in databases. In *Proc. 13TH VLDB*, 564–575.

Borodin, A.; Roberts, G.; Rosenthal, J.; and Tsaparas, P. 2005. Link analysis ranking: Algorithms, theory, and experiments. *ACM TOIT* 5(1):231–297.

Clauset, A.; Newman, M.; and Moore, C. 2004. Finding community structure in very large networks. *Phys. Rev. E* 70(6):066111.

Faust, K., and Wasserman, S. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press.

Gyongyi, Z.; Garcia-Molina, H.; and Pedersen, J. 2004. Combating web spam with trustrank. In *Proc 30th VLDB*.

Haveliwala, T. 2002. Topic-sensitive pagerank. In *Proc. ACM WWW*, 517–526.

Hirsch, J. 2005. An index to quantify an individual's scientific research output. *PNAS* 46:16569–16572.

Hotho, A.; Jaschke, R.; Schmitz, C.; and Stumme, G. 2006. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*.

Kleinberg, R. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46(5):604–632.

Liu, X.; Bollen, J.; Nelson, M. L.; and de Sompel, H. V. 2005. Co-authorship networks in the digital library research community. *Information Processing & Management* 41:14621480.

Magdon-Ismail, M., and Purnell, J. 2010. Fast overlapping clustering of networks using sampled spectral distance embedding and gmms. Technical Report 10-06, RPI.

Sun, Y.; Han, J.; Zhao, P.; Yin, Z.; Cheng, H.; and Wu, T. 2009. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *Proc. 12th EDBT*, 565–576.