Symmetrizations for Clustering Directed Graphs

Venu Satuluri
Dept. of Computer Science and Engineering
The Ohio State University
satuluri@cse.ohio-state.edu

Srinivasan Parthasarathy
Dept. of Computer Science and Engineering
The Ohio State University
srini@cse.ohio-state.edu

ABSTRACT

Graph clustering has generally concerned itself with clustering undirected graphs; however the graphs from a number of important domains are essentially directed, e.g. networks of web pages, research papers and Twitter users. This paper investigates various ways of symmetrizing a directed graph into an undirected graph so that previous work on clustering undirected graphs may subsequently be leveraged. Recent work on clustering directed graphs has looked at generalizing objective functions such as conductance to directed graphs and minimizing such objective functions using spectral methods.We show that more meaningful clusters (as measured by an external ground truth criterion) can be obtained by symmetrizing the graph using measures that capture in- and out-link similarity, such as bibliographic coupling and co-citation strength. However, direct application of these similarity measures to modern large-scale power-law networks is problematic because of the presence of hub nodes, which become connected to the vast majority of the network in the transformed undirected graph. We carefully analyze this problem and propose a Degree-discounted similarity measure which is much more suitable for large-scale networks. We show extensive empirical validation.

Categories and Subject Descriptors

G.2.2 [**Graph Theory**]: Graph Algorithms; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms,Performance

Keywords

Directed Graphs, Clustering, Graph Transformations

1. INTRODUCTION

A number of complex systems and applications can be modeled in the form of a relationship graph or network. Examples abound ranging from Protein Interaction Networks to Twitter networks, from Citation networks to technological networks such as the hyperlinked structure on the World Wide Web. Analyzing such networks can yield important insights about the domain problem in question. A common analysis tool here is to discover the community or cluster structure within such networks.

Directed graphs are essential in domains where relationships between the objects may not be recriprocal i.e., there may be an implicit or explicit notion of directionality in the context of the complex system being modeled. Most of the work to date on community discovery or clustering of graphs has targeted undirected networks and very little has focused on the thorny issue of community discovery in directed networks as noted in a recent survey on the topic[8].

A major challenge is that the nature of relationships captured by the edges in directed graphs is fundamentally different from that for undirected graphs. Consider a citation network where an edge exists from paper i to j if i cites j. Now i may be a paper on databases that cites an important result from the algorithmic literature (j). Our point is that paper i need not necessarily be similar to paper j. A common approach to handle directionality is to ignore it – i.e. eliminate directionality from edges and compute communities. In the above example that would not be the appropriate solution. Such a semantics of directionality is also evident in the directed social network of Twitter, where if a person i follows the feed of a person j, it tell us that i thinks the updates of j are interesting but says nothing about the similarity of i and j.

The central, and novel, insight driving our research is that groups of vertices which share similar in-links and out-links make meaningful clusters in directed graphs. This is in direct contrast to previous research (summarized in Section 2) on clustering directed graphs, which either simply ignores the directionality of the edges or concentrates on new objective functions for directed graphs which do not take into account in-link and out-link similarity of the nodes. For detecting clusters with homogenous in-link and out-link structure, we suggest a two-stage framework; in the first stage, the graph is *symmetrized* i.e. transformed into an undirected graph, and in the second stage, the symmetrized graph is clustered using existing state-of-the-art graph clustering algorithms. The advantages of the two-stage symmetrization framework are that (i) it is flexible - prior methods for directed graph clustering can also be equivalently expressed

EDBT 2011, March 22–24, 2011, Uppsala, Sweden.

in this framework, (ii) it makes the underlying assumptions about which kinds of nodes should be clustered together explicit (i.e. the implicit similarity measure being used in the clustering), and (iii) it allows us to leverage the progress made in (undirected) graph clustering algorithms. We propose two novel symmetrization methods, *Bibliometric* and *Degree-discounted*. Bibliometric symmetrization sets the similarity between a pair of nodes as being the number of common in- and out-links between the two nodes. However, this approach does not work well with large scale power-law graphs, since the hub nodes in such graphs introduce many spurious connections in the symmetrized graph. To alleviate this problem, we propose *Degree-discounted* symmetrization which discounts the contribution of nodes according to their degree, and therefore eliminates or downweights such hub-induced connections in the symmetrized graph.

We perform evaluation on four real datasets, three of which (Wikipedia, LiveJournal and Flickr) have million plus nodes, and two (Wikipedia,Cora) of which have dependable ground truth for evaluating the resulting clusters. We examine the characteristics of the different symmetrized graphs in terms of their suitableness for subsequent clustering. Our proposed Degree-discounting symmetrization approach achieves a 22% improvement in F scores over a state-of-the-art directed spectral clustering algorithm on the Cora dataset, and furthermore is two orders of magnitude faster. The degree-discounting symmetrization is also shown to enable clustering that is at least 4-5 times faster than other symmetrizations on our large scale datasets, as well as enabling a 12% qualitative improvement in Wikipedia. We also show examples of the clusters that our symmetrization enables recovery of in the Wikipedia dataset; such clusters validate our claim that interconnectivity is not the only criterion for clusters in directed graphs, and that in-link and out-link similarity is important as well. Ours is, to the best of our knowledge, the first comprehensive comparison of different graph symmetrization techniques.

In summary, the contributions of our paper are as follows:

1. We argue and provide evidence for the merits of an explicit symmetrization-based approach to clustering directed graphs. This is in contrast to recent work which attempted to design specialized spectral algorithms with limited scalability.

2. We propose the Bibliometric and Degree-discounted symmetrizations that take into account in-link and out-link similarities (which existing directed graph clustering approaches do not), with Degree-discounted also appropriately downweighting the influence of hub nodes.

3. We extensively compare the different symmetrizations, as well as a state-of-the-art directed graph clustering algorithm, on real world networks, providing empirical evidence for the usefulness of our proposed approaches.

2. PRIOR WORK

2.1 Normalized cuts for directed graphs

Many popular methods for clustering undirected graphs search for subsets of vertices with low *normalized cut* [11, 18, 21] (or *conductance*[11], which is closely related). The normalized cut of a group of vertices $S \subset V$ is defined as[21, 18]

$$Ncut(S) = \frac{\sum_{i \in S, j \in \bar{S}} A(i,j)}{\sum_{i \in S} degree(i)} + \frac{\sum_{i \in S, j \in \bar{S}} A(i,j)}{\sum_{j \in \bar{S}} degree(j)} \quad (1)$$

where A is the (symmetric) adjacency matrix and $\bar{S} = V - S$ is the complement of S. Intuitively, groups with low normalized cut are well connected amongst themselves but are sparsely connected to the rest of the graph.

The connection between random walks and normalized cuts is as follows [18] : $Ncut(S)$ in Equation 1 is the same as the probability that a random walk that is started in the stationary distribution will transition either from a vertex in S to a vertex in $\bar{S}$ or vice-versa, in one step [18]

$$Ncut(S) = \frac{Pr(S \to \bar{S})}{Pr(S)} + \frac{Pr(\bar{S} \to S)}{Pr(\bar{S})} \quad (2)$$

Using the unifying concept of random walks, Equation 2 have been extended to directed graphs by Zhou et. al. [24] and Huang et. al. [10]. Let P be the transition matrix of a random walk on the directed graph, with π being its associated stationary distribution vector (e.g. PageRank vector) satisfyfing $\pi P = \pi$. The probability that a random walk started in the stationary distribution traverses a particular directed edge $u \to v$ is given by $\pi(u)P(u,v)$. The $Ncut$ of a cluster S is again the probability of a random walk transitioning from S to the rest of the graph, or from the rest of the graph into S in one step:

$$Ncut_{dir}(S) = \frac{\sum_{i \in S, j \in \bar{S}} \pi(i)P(i,j)}{\sum_{i \in S} \pi(i)} + \frac{\sum_{j \in \bar{S}, i \in S} \pi(j)P(j,i)}{\sum_{j \in \bar{S}} \pi(j)} \quad (3)$$

Meila and Pentney [17] introduce a general class of weighted cut measures on graphs, called $WCut$, parameterized by the vectors T, T' and the matrix A:

$$WCut(S) = \frac{\sum_{i \in S, j \in \bar{S_a}} T'(i)A(i,j)}{\sum_{i \in S} T(i)} + \frac{\sum_{j \in \bar{S}, i \in S} T'(j)A(j,i)}{\sum_{j \in \bar{S}} T(j)} \quad (4)$$

Different $NCut$ measures can be recovered from the above definition by plugging in different values for T, T' and A, including the definitions for $NCut$ and $NCut_{dir}$ given above.

All of the above work minimizes these various cut measures via spectral clustering i.e. by post-processing the eigenvectors of suitably defined *Laplacian* matrices. The Laplacian matrix $\mathcal{L}$ for $Ncut_{dir}$, e.g., is given by [24, 10, 4]

$$\mathcal{L} = I - \frac{\Pi^{1/2} P \Pi^{-1/2} + \Pi^{-1/2} P' \Pi^{1/2}}{2} \quad (5)$$

where P is the transition matrix of a random walk, and Π is a diagonal matrix with $diag(P) = \pi$, π being the stationary distribution associated with P.

2.1.1 Drawbacks of normalized cuts for directed graphs

A common drawback of the above line of research is that there exist meaningful clusters which do not necessarily have a low directed normalized cut. The prime examples here are groups of vertices which do not point to one another, but all of which point a common set of vertices (which may belong to a different cluster) We present an idealized example of such situations in Figure 1, where the nodes 4 and 5 can be legitimately seen as belonging to the same cluster, and yet the $Ncut_{dir}$ for such a cluster will be high (the probability that a random walk transitions out of the cluster $\{4, 5\}$

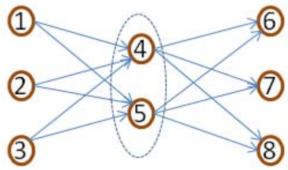


Figure 1: The nodes 4 and 5 form a natural cluster even though they don't link to one another, as they point to the same nodes and are also pointed to by the same nodes.

to the rest of the graph, or vice versa, in one step, is very high.) Such situations may be quite common in directed graphs. Consider, for example, a group of websites that belong to competing companies which serve the same market; they may be pointing to a common group of websites outside themselves (and, similarly be pointed at by a common group of websites), but may not point at one another for fear of driving customers to a competitor's website. Another example may be a group of research papers on the same topic which are written within a short span of time and therefore do not cite one another, but cite a common set of prior work and are also in the future cited by the same papers. We present real examples of such clusters in Section 5.7.

Another drawback of the above line of research is the poor scalability as a result of the dependence on spectral clustering (except for Andersen et. al. [1] who use local partitioning algorithms). We further discuss this issue in Section 3.2.

2.2 Bibliographic coupling and co-citation matrices

The *bibliographic coupling* matrix was introduced by Kessler [13], in the field of bibliometrics, for the sake of counting the number of papers that are commonly cited by two scientific documents. It is given by $B = AA^T$, and $B[i, j]$ gives the number of nodes that the nodes i and j both point to in the original directed graph with adjacency matrix A.

$$\begin{aligned} B(i,j) &= \sum_k A(i,k)A(j,k) \\ &= \sum_k A(i,k)A^T(k,j) \\ B &= AA^T \end{aligned}$$

The *co-citation* matrix was introduced by Small [22], again in the field of bibliometrics. It is given by $C = A^T A$, and $C[i, j]$ gives the number of nodes that commonly point to both i and j in the original directed graph.

3. GRAPH SYMMETRIZATIONS

We adopt a two-stage approach for clustering directed graphs, schematically depicted in Figure 2. In the first stage we transform the directed graph into an undirected graph (i.e. symmetrize the directed graph) using one of different possible symmetrization methods. In the second stage, the undirected graph so obtained is clustered using one of several possible graph clustering algorithms. The advantage of this approach is that it allows a practitioner to employ a graph clustering algorithm of their choice in the second stage. For example, spectral clustering algorithms are typically state-of-the-art quality-wise, but do not scale well as eigenvector computations can be very time-consuming [5]. Under such circumstances, it is useful to be able to plug in a scalable graph clustering algorithm of our own choice, such as Graclus [5], MLR-MCL [20], Metis [12] etc. Note that it is not the objective of this paper to propose a new (undirected) graph clustering algorithm or discuss the strengths and weaknesses of existing ones; all we are saying is that whichever be the suitable graph clustering algorithm, it will fit in our framework.

Of course, the effectiveness of our approach depends crucially on the the symmetrization method. If the symmetrization itself is flawed, even a very good graph clustering algorithm will not be of much use. But do we have reason to believe that an effective symmetrization of the input directed graph is possible? We believe the answer is yes, at least if the domain in question does indeed have some cluster structure. Fundamentally a cluster is a group of objects that are similar to one another and dissimilar to objects not in the cluster. If a domain admits of clusters, this means that there must exist some reasonable similarity measure among the objects in that domain. Since similarity measures are generally symmetric (i.e. $similarity(i, j) = similarity(j, i)$) and positive, defining a notion of similarity for a fixed set of input objects is equivalent to constructing an undirected graph among them, with edges between pairs of objects with non-zero similarity between them and the edge weight equal to the actual value of the similarity. In fact, our proposed degree-discounted symmetrization method can just as validly be thought of as measuring the similarity between pairs of vertices in the input directed graph.

We next discuss various ways to symmetrize a directed graph. In what follows, G will the original directed graph with associated (asymmetric) adjacency matrix A. G_U will be the resulting symmetrized undirected graph with associated adjacency matrix U.

3.1 $A + A^T$

The simplest way to derive an undirected graph from a directed one is via the transformation $U = A + A^T$. Note that this is very similar to the even simpler strategy of simply ignoring the directionality of the edges, except that in the case of pairs of nodes with directed edges in both directions, the weight of the edge in the symmetrized graph will be the sum of the weights of the two directed edges. It is important to empiricallly compare this scheme against other symmetrizations since this is the implicit symmetrization commonly used [15, 5, 17, 24].

The advantage of this method is, of course, its simplicity. On the other hand, this method will fare poorly with situations of the sort depicted in Figure 1; the nodes 4 and 5 will continue to remain unconnected in the symmetrized graph, making it impossible to cluster them together.

3.2 Random walk symmetrization

Is it possible to symmetrize a directed graph G into G_U such that the directed normalized cut of a group of vertices S, $NCut_{dir}(S)$ is equal to the (undirected) normalized cut of the same group of vertices in the symmetrized graph G_U? The answer turns out to be yes.

Let P be the transition matrix of the random walk, π its associated stationary distribution, and Π is the diagonal matrix with π on the diagonal.Let U be the symmetric matrix

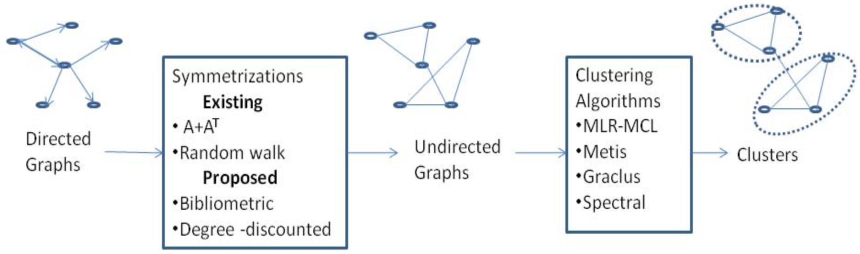


Figure 2: Schematic of our framework

such that

$$U = \frac{\Pi P + P^T \Pi}{2}$$

Gleich [9] showed that for the symmetrized graph G_U with associated adjacency matrix U, the (undirected) Ncut on this graph is equal to the directed Ncut on the original directed graph, for any subset of vertices S. This means that clusters with low directed ncut can be found by clustering the symmetrized graph G_U, and one can use any state-of-the-art graph clustering for finding clusters with low ncut in G_U, instead of relying on expensive spectral clustering using the directed Laplacian (given in Eqn 5) as previous researchers have [24, 10].

The matrix P can be obtained easily enough by normalizing the rows of input adjacency matrix A, and the stationary distribution π can be obtained via power iterations. However, the clusters obtained by clustering G_U will still be subject to the same drawbacks that we pointed out in Section 2.1.1. Also note that this symmetrization leads to the exact same set of edges as $A + A^T$, since P and P^T have the same non-zero structure as A and A^T and Π is a diagonal matrix. The actual weights on the edges will, of course, be different for the two methods.

3.3 Bibliometric symmetrization

One desideratum of the symmetrized graph is that edges should be present between nodes that share similar (in- or out-) links, and edges should be absent between nodes in the absence of shared (in- or out-)links. Both $A + A^T$ and Random walk symmetrization fail in this regard as they retain the exact same set of edges as in the original graph; only the directionality is dropped and, in the case of the Random walk symmetrization, weights are added to the existing edges.

The bibliographic coupling matrix (AA^T) and the co-citation strength matrix (A^TA) are both symmetric matrices that help us satisfy this desideratum. Recall that AA^T measures the number of common out-links between each pair of nodes, where as A^TA measures the number of common in-links. As there does not seem to be any obvious reason for leaving out either in-links or out-links, it is natural to take the sum of both matrices so as to account for both. In this case $U = AA^T + A^TA$, and we refer to this as bibliometric symmetrization. [1]

Setting $A := A + I$ prior to the symmetrization ensures that edges in the input graph will not be removed from the symmetrized version.

3.4 Degree-discounted symmetrization

As a consequence of the well-known fact that the degree distributions of many real world graphs follow a power law distribution [7, 3], nodes with degrees in the tens as well as in the thousands co-exist in the same graph. (This is true for both in-degrees and out-degrees.) This wide disparity in the degrees of nodes has implications for the Bibliometric symmetrization; nodes with high degrees will share a lot of common (in- or out-) links with other nodes purely by virtue of their higher degrees. This is the motivation for our proposed *Degree-discounted* symmetrization approach, where we take into account the in- and out-degrees of each node in the symmetrization process.

Another motivation for our proposed symmetrization is defining a useful similarity measure between vertices in a directed graph. As noted earlier in Section 3, a meaningful similarity measure will also serve to induce an effective symmetrization of the directed graph; ideally, we want our symmetrized graph to place edges of high weight between nodes of the same cluster and edges of low weight between nodes in different clusters.

How exactly should the degree of nodes enter into the computation of similarity between pairs of nodes in the graph? First we will consider how the computation of out-link similarity (i.e. the bibliographic coupling) should be changed to incorporate the degrees of nodes.

Consider the following two scenarios (see Figure 3(a)):

1. Nodes i and j both point to the node h, which has in-coming edges from many nodes apart from i and j. In other words, the in-degree of h, $D_i(h)$ is high.

2. Nodes i and j both point to the node k, but which has in-coming edges only from a few other nodes apart from i and j.

Intuition suggests that case 1 above is a more frequent (hence less informative) event than case 2, and hence the former event should contribute less towards the similarity between i and j than the latter event. In other words, *when two nodes i and j commonly point to a third node, say l, the contribution of this event to the similarity between i and j should be inversely related to the in-degree of l.*

Next we consider how the degree of two nodes should factor into the similarity computation of those two nodes themselves. Figure 3(b) illustrates the intuition here: sharing a common out-link k counts for less when one of the two nodes that are doing the sharing is a node with many out-links. In

[1]Meila and Pentney [17] compare against the A^TA symmetrization, but neither suggest nor compare against the $AA^T + A^TA$ symmetrization. To the best of our knowledge, this symmetrization is new to our work.

other words, *the out-link similarity between i and j should be inversely related to the out-degrees of i and j.*

We have determined qualitatively how we should take the in- and the out-degrees of the nodes into account, but the exact form of the relationship remains to be specified. We have found experimentally that discounting the similarity by the square root of the degree yields the best results; making the similarity inversely proportional to the degree itself turned out to be an excessive penalty.

With the above insights, we define the out-link similarity or degree-discounted bibliographic coupling between the nodes i and j as follows:(D_o is the diagonal matrix of out-degrees, and $D_o(i)$ is short-hand for $D_o(i,i)$. Similarly D_i is the diagonal matrix of in-degrees. α and β are the discounting parameters.)

$$\begin{aligned} B_d(i,j) &= \frac{1}{D_o(i)^\alpha D_o(j)^\alpha} \sum_k \frac{A(i,k)A(j,k)}{D_i(k)^\beta} \\ &= \frac{1}{D_o(i)^\alpha D_o(j)^\alpha} \sum_k \frac{A(i,k)A^T(k,j)}{D_i(k)^\beta} \end{aligned}$$

Note that the above expression is symmetric in i and j. It can be verified that the entire matrix B_d with its (i,j) entries specified as above can be expressed as:

$$B_d = D_o^{-\alpha} A D_i^{-\beta} A^T D_o^{-\alpha} \tag{6}$$

Our modification for the co-citation (in-link similarity) matrix is exactly analogous to the above discussion; we proceed to directly give the expression for the matrix C_d containing the degree-discounted co-citation or in-link similarities between all pairs of nodes.

$$C_d = D_i^{-\beta} A^T D_o^{-\alpha} A D_i^{-\beta} \tag{7}$$

The final degree discounted similarity matrix U_d is simply the sum of B_d and C_d.

$$U_d = B_d + C_d$$

Empirically we have found $\alpha = \beta = 0.5$ to work the best. Using $\alpha = \beta = 1$ penalized hub nodes excessively, while smaller values such as 0.25 were an insufficient penalty. Penalizing using the log of the degree (similar to the IDF transformation [16]) was also an insufficient penalty. Therefore, the final degree-discounted symmetrization is defined as follows:

$$U_d = D_o^{-1/2} A D_i^{-1/2} A^T D_o^{-1/2} + D_i^{-1/2} A^T D_o^{-1/2} A D_i^{-1/2} \tag{8}$$

We point out that the degree-discounting intuition has been found to be effective for solving other problems on directed graphs previously. In the context of node ranking, Ding et al. [6] combine the mutual re-inforcement of HITS with the degree-discounting of PageRank to obtain ranking algorithms that are intermediate between the two. In the context of semi-supervised learning, Zhou et al. [25] propose to regularize functions on directed graphs so as to force the function to change slowly on vertices with high normalized in-link or out-link similarity.

3.5 Pruning the symmetrized matrix

One of the main advantages of Degree-discounted symmetrization over Bibliometric symmetriation ($AA^T + A^T A$) is that it is much easier to prune the resulting matrix. $AA^T + A^T A$ and the Degree-discounted similarity matrix U_d share the same non-zero structure, but the actual values are, of course, different. For big real world graphs, the full similarity matrix has far too many non-zero entries and clustering the entire resulting undirected graph is very time-consuming. For this reason, it is critical that it be possible for us to pick a threshold so as to be able to retain only those entries in the matrix which pertain to sufficiently similar pairs of nodes. However, picking a threshold for $AA^T + A^T A$ can be very hard; as the degrees of nodes are not taken into account, the hub nodes in the graph generate a large number of non-zero entries with high non-zero values (this is because hubs will tend to share a lot of out-links and in-links with a lot of nodes just by virtue of their having high degrees). When we set a high threshold so as to keep the matrix sparse enough to be able to cluster in a reasonable amount of time, many of the rows corresponding to the other nodes become empty. When we lower the threshold in response, the matrix becomes very dense and it becomes impractical to cluster such a dense matrix.

This problem is considerably reduced when applying Degree-discounted symmetrization. This is because the matrix entries involving hub nodes no longer are the largest; this lets us choose a threshold such that when we retain only matrix entries above the threshold, we have a matrix that is sufficiently sparse and at the same time covers the majority of nodes in the graph.

3.6 Complexity analysis

The time complexity in general for multiplying dense matrices is $O(n^{2.8})$ using Strassen's algorithm, where n is the number of rows/columns. However, since our matrices are sparse, we can do significantly better than that. Each node i that has d_i connections (either through in-links or out-links), contributes to the similarity between each of the $\binom{d_i}{2}$ pairs of nodes it connects to. Therefore, the total number of similarity contributions that will need to be computed and added up is $\sum_i d_i^2$, which means that a new upper bound on the time complexity of the similarity computation is $O(\sum_i d_i^2)$. We can further improve upon this by exploiting the fact that we only want to compute those entries in the similarity matrix which are above a certain prune threshold. Bayardo et. al. [2] outline approaches for curtailing similarity computations that will provably lead to similarities lower than the prune threshold, and which can enable significant speedups compared to computing all the entries in the similarity matrix.

In terms of space complexity, the similarity computation requires no extra space in addition to that required to store the similarities themselves.

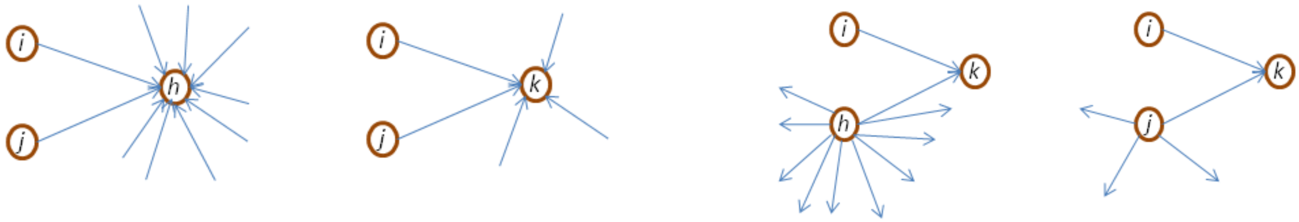
4. EXPERIMENTAL SETUP

4.1 Datasets

We perform experiments using four real datasets, detailed below. Also see Table 1.

1. Wikipedia: This is a directed graph of hyperlinks between Wikipedia articles. We downloaded a snapshot of the entire Wikipedia corpus from the Wikimedia foundation [2] (Jan–2008 version). The corpus has nearly 12 million articles, but a lot of these were insignificant or noisy articles

[2]http://download.wikimedia.org/



(a) If the nodes i and j both point to a hub node h with many incoming edges (**left**), that should contribue lesser to their similarity than if they commonly point to a non-hub node k (**right**)

(b) All else equal, the node i should be less similar to the hub node h which has many out-going edges (**left**) when compared to the non-hub node j (**right**).

Figure 3: Scenarios illustrating the intuition behind degree-discounting.

Dataset	Vertices	Edges	Percentage of symmetric links	No. of ground truth categories
Wikipedia	1,129,060	67,178,092	42.1	17950
Cora	17,604	77,171	7.7	70
Flickr	1,861,228	22,613,980	62.4	N.A.
Livejournal	5,284,457	77,402,652	73.4	N.A.

Table 1: Details of the datasets

that we removed as follows. First, we retained only those articles with an abstract, which cut the number of articles down to around 2.1 million. We then constructed the directed graph from the hyperlinks among these pages and retained only those nodes with out-degree greater than 15. We finally obtained a directed graph with 1,129,060 nodes and 67,178,092 edges, of which 42.1% are bi-directional.

Pages in Wikipedia are assigned to one or more categories by the editors (visible at the bottom of a page), which we used to prepare ground truth assignments for the pages in our dataset. We removed the many categories that are present in Wikipedia for housekeeping purposes (such as "Articles of low significance", "Mathematicians stubs"). We further removed categories which did not have more than 20 member pages in order to remove insignificant categories. We obtained 17950 categories after this process. Note that these categories are not disjoint, i.e. a page may belong to multiple categories (or none). Also, 35% of the nodes in the graph do not have any ground truth assignment.

2. Cora: This is a directed graph of CS research papers and their citations. It has been collected and shared by Andrew McCallum [3]. Besides just the graph of citations, the papers have also been manually classified into 10 different fields of CS (such as AI, Operating Systems, etc.), with each field further sub-divided to obtain a total of 70 categories at the lowest level. Again, 20% of the nodes have not been assigned any labels.We utilize the classifications at the lowest level (i.e. 70 categories) for the sake of evaluation. This graph consists of 17,604 nodes with 77,171 directed edges. Note that although symmetric links are, strictly speaking, impossible in citation networks (two papers cannot cite one another as one of them will need to have been written before the other), there is still a small percentage (7.7%) of symmetric links in this graph due to noise.

3. Flickr and **4. Livejournal**:These are large scale directed graphs, collected by the Online Social Networks Research group at The Max Planck Institute [19]. The number of nodes and edges for these datasets can be found in Table 1. We use these datasets only for scalability evaluation as we do not have ground truth information for evaluating effectiveness of discovered clusters.

4.2 Setup

We compare four different graph symmetrization methods described in Section 3. For Random walk symmetrization, the stationary distribution was calculated with a uniform random teleport probability of 0.05 in all cases. We clustered the symmetrized graphs using MLR-MCL [20], Metis [12] and Graclus [5]. We are able to show the results of Graclus only on the Cora dataset as the program did not finish execution on any of the symmetrized versions of the Wikipedia dataset. Note that the number of output clusters in MLR-MCL can *only be indirectly controlled* via changing some other parameters of the algorithm; for this reason there is a slight variation in the number of clusters output by this algorithm for different symmetrizations.

We also compare against the BestWCut algorithm described by Meila and Pentney [17], but on the Cora dataset alone, as the algorithm did not finish execution on the Wikipedia dataset. It bears emphasis that BestWCut is not a symmetrization method. The directed spectral clustering of Zhou et. al. [24] did not finish execution on any of our datasets.

All the experiments were performed on a dual core machine (Dual 250 Opteron) with 2.4GHz of processor speed and 8GB of main memory. However, the programs were single-threaded so only one core was utilized. The software for each of the undirected graph clustering algorithms as well as BestWCut [17] was obtained from the authors' respective webpages. We implemented the different symmetrization methods in C, using sparse matrix representations.

4.3 Evaluation method

The clustering output by any algorithm was evaluated

[3]http://www.cs.umass.edu/ mccallum/code-data.html

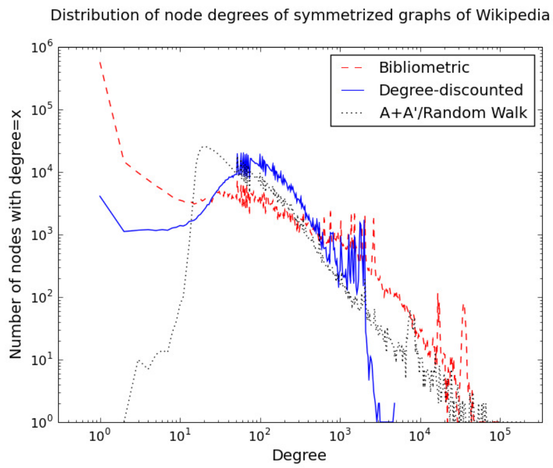


Figure 4: Distributions of node degrees for different symmetrizations of Wikipedia.

with respect to the ground truth clustering as follows. Let the output clustering be $\mathcal{C} = \{C_1, C_2, \ldots, C_i, \ldots, C_k\}$. For any output cluster C_i, the precision and recall of this cluster against a given ground truth category, say G_j, are defined as: $Prec(C_i, G_j) = \frac{|C_i \cap G_j|}{|C_i|}$ and $Rec(C_i, G_j) = \frac{|C_i \cap G_j|}{|G_j|}$. The F-measure $F(C_i, G_j)$ is the harmonic mean of the precision and the recall. We match each output cluster C_i with the ground truth cluster G_j for which $F(C_i, G_j)$ is the highest among all ground truth clusters. This is the F-measure that is subsequently associated with this cluster, and is referred to as $F(C_i)$; i.e.

$$F(C_i) = \max_j F(C_i, G_j)$$

The average F-measure of the entire clustering $\mathcal{C}$ is defined as the average of the F-measures of all the clusters, weighted by their sizes (i.e. we compute the micro-averaged F-measure).

$$Avg.F(\mathcal{C}) = \frac{\sum_i |C_i| * F(C_i)}{\sum_i |C_i|}$$

5. RESULTS

5.1 Characteristics of symmetrized graphs

The number of edges in the resulting symmetrized graph for each symmetrization method for the different datasets is given in Table 2, along with the pruning thresholds used. To obtain more insight into the structure of the symmetrized graphs, we analyze the distribution of node degrees in the case of Wikipedia (see Figure 4). Note that $A + A^T$ and Random Walk have the same distributions, as they have the same set of edges. The Degree-discounted method ensures that most nodes have medium degrees in the range of 50-200 (which is about the size of the average cluster [15]), and completely eliminates hub nodes. These properties enable subsequent graph clustering algorithms to perform well. The Bibliometric graph, on the other hand, has many nodes with both very low degrees, as well as many hub nodes, making clustering the resulting graph difficult. The $A + A^T$ graph also has more hub nodes than the Degree-discounted graph.

5.2 Results on Cora

Results pertaining to cluster quality as well as clustering time on the Cora dataset are shown in Figures 5 and 6.

Figure 5 (a) compares the Avg. F scores obtained using MLR-MCL with different symmetrizations. For all symmetrizations, the performance reaches a peak at 50-70 clusters, which is close to the true number of clusters (70). With fewer clusters, the precision is adversely impacted, while a greater number of clusters affects the recall. Degree-discounted symmetrization on the whole yields better F scores than the other methods, and also achieves the best overall F-value of 36.62. Bibliometric symmetrization also yields good F-scores with a peak of 34.92, and marginally improves on Degree-discounted for higher number of clusters. $A + A^T$ and Random walk perform similarly and are relatively poor compared to the other two methods. Figure 5 (b) shows the effectiveness of different symmetrizations, this time using a different clustering algorithm, Graclus. Degree-discounted symmetrization clearly delivers improvements over the other symmetrizations in this case as well. This shows that multiple clustering algorithms can benefit from the proposed symmetrizations.

Figure 6 (a) fixes the symmetrization to Degree discounted and compares MLR-MCL, Graclus and Metis with Meila and Pentney's BestWCut [17]. The peak F score achieved by BestWCut is 29.94, while the peak F-scores for MLR-MCL, Graclus and Metis are 36.62, 34.69 and 34.30 respectively. Therefore Degree-discounted symmetrization combined with any of the three clustering algorithms - either MLR-MCL, Graclus or Metis - comfortably outperforms BestWCut. Using MLR-MCL, Degree-discounted symmetrization improves upon BestWCut by 22%.

Figure 6 (b) compares cluster times of MLR-MCL, Graclus and Metis with Degree-discounted symmetrization against the time taken by BestWCut. All three are much faster than BestWCut. The slow performance of BestWCut is because of the need for expensive eigenvector computations, which none of the other three algorithms involve.

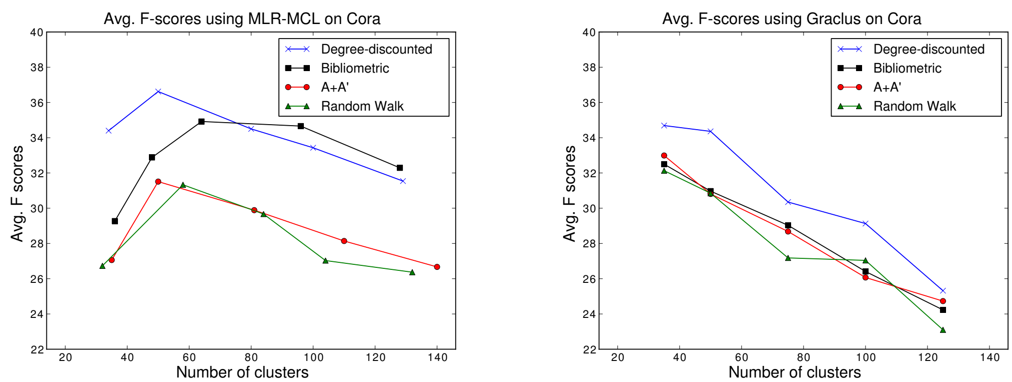
5.3 Results on Wikipedia

We next turn to cluster quality and timing results on Wikipedia, depicted in Figures 7 and 8. In general, this dataset was harder to cluster than the Cora dataset, with an overall peak Avg. F score of 22.79, compared to 36.62 for Cora. Note that we do not have any results from BestWCut [17] on this dataset as it did not finish execution.

Figure 7 (a) and (b) compares the Avg. F scores with different symmetrizations using MLR-MCL and Metis.Degree-discounted symmetrization yields the best Avg F scores, with a peak F value of 22.79. $A + A^T$ gives the next best results, with a peak F value of 20.31. These peak scores were obtained using MLR-MCL. Metis on Degree-discounted symmetrization achieves a peak F-value of 20.15, a significant 27% improvement on the next best F-value of 15.95, achieved using $A + A^T$. Therefore Degree-discounted symmetrization benefits both MLR-MCL and Metis. The performance of Random Walk is slightly worse than $A + A^T$ but is otherwise similar. We do not report Metis combined with Random Walk symmetrization as the program crashed when run with this input. Bibliometric performs very poorly, with F scores barely touching 13%. The main reason for the poor performance of Bibliometric is that explained in Section 3.5 - even though we pruned the outputs of both Bibliometric

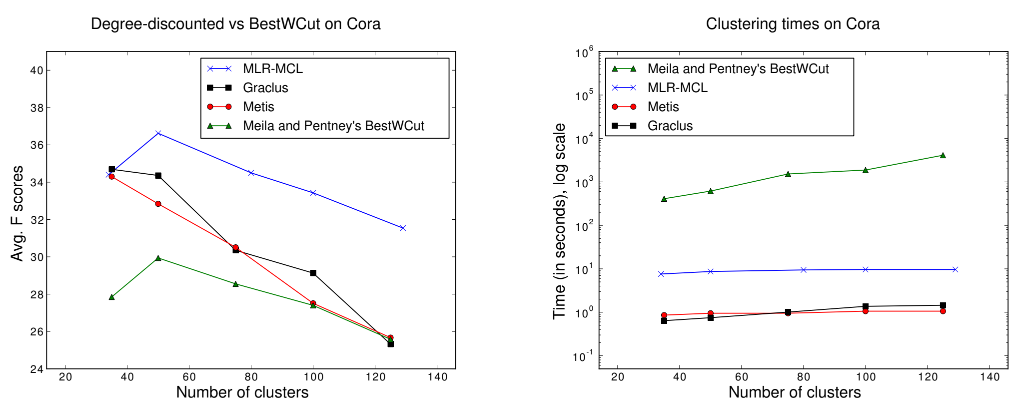
Dataset	$A + A^T$ / Random Walk	Bibliometric		Degree-discounted	
	Edges	Edges	Threshold	Edges	Threshold
Wikipedia	53,017,527	85,035,548	25	80,373,184	0.01
Flickr	15,555,041	79,765,961	20	45,167,216	0.01
Cora	74,180	986,444	0	986,444	0
Livejournal	51,352,001	143,759,001	5	91,624,309	0.025

Table 2: Number of edges for various symmetrization strategies, and the pruning thresholds used.



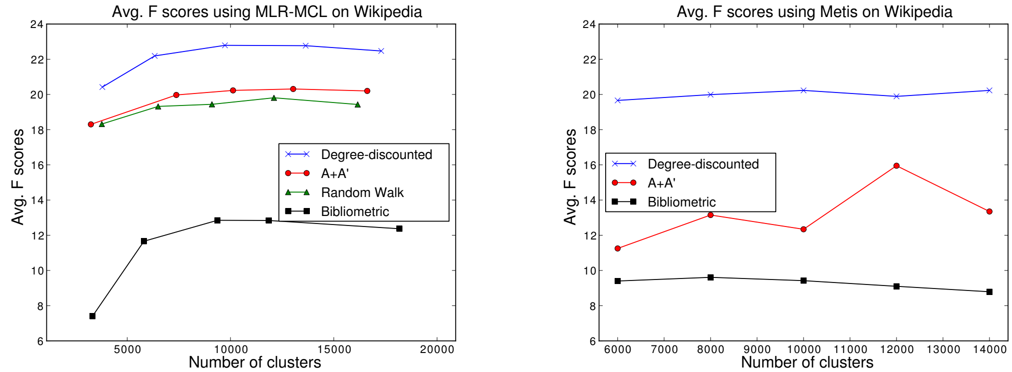
(a) (b)

Figure 5: Effectiveness of symmetrizations on Cora using (a) MLR-MCL and (b) Graclus, as the clustering algorithms.



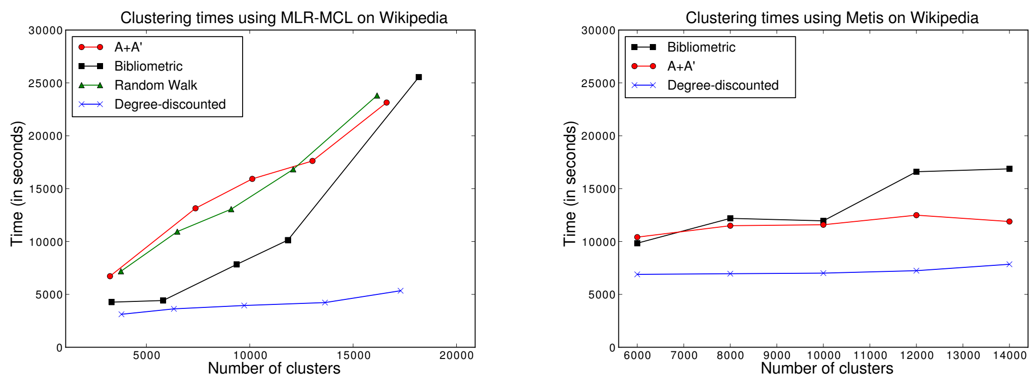
(a) (b)

Figure 6: Comparison of Degree-discounted symmetrization vs. Meila and Pentney's BestWCut w.r.t. (a) Effectiveness and (b) Speed.



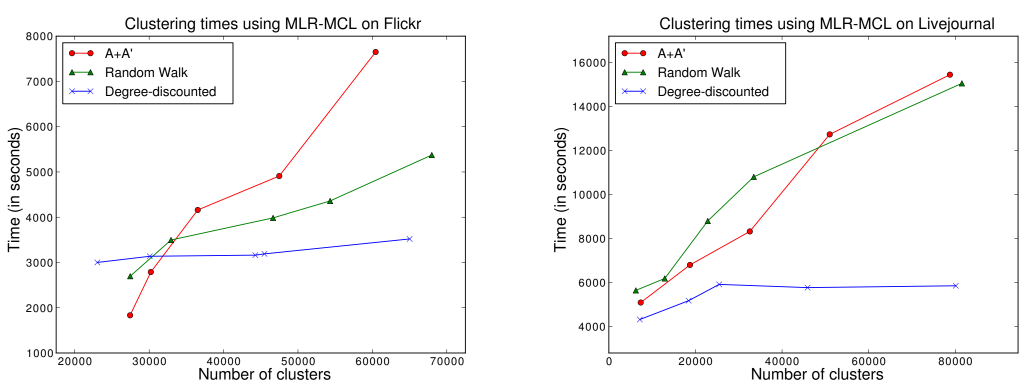
(a) (b)

Figure 7: Effectiveness of symmetrizations on Wikipedia using (a) MLR-MCL and (b) Metis, as the clustering algorithms.



(a) (b)

Figure 8: Clustering times on Wikipedia using (a) MLR-MCL and (b) Metis, as the clustering algorithms.



(a) (b)

Figure 9: Clustering times using MLR-MCL on (a) Flickr and (b) LiveJournal

Threshold	No. of edges	MLR-MCL		Metis	
		F score	Time	F score	Time
0.010	80,373,184	22.47	4225	20.15	7010
0.015	73,273,127	22.45	3615	20.06	4488
0.020	50,801,885	22.27	1912	20.04	1399
0.025	37,663,652	21.72	1039	19.86	547

Table 3: Effect of varying pruning threshold

and Degree-discounted symmetrizations so that they contained a similar number of edges (around 80 milion), the Bibliometric graph still ended up with nearly 50% of the nodes as singletons. It is worth mentioning that there was no such problem with Degree-discounted.

Figure 8 (a) and (b) show the time to cluster different symmetrizations using MLR-MCL and Metis. We find that both MLR-MCL and Metis execute faster with Degree-discounted, than any of the other symmetrizations. The difference becomes more pronounced with increasing number of clusters; MLR-MCL executes nearly 4.5 to 5 times faster on Degree-discounted as compared to the other symmetrizations in the high clusters range (16000-18000). We believe that the absence of hub nodes (as can be seen in Fig 4), coupled with clearer cluster structures in the Degree-discounted graph explains its better performance. It is also interesting to note that on this dataset MLR-MCL is on average significantly faster (2000s) than Metis on the degree-discounted transformation.

5.3.1 *Varying the prune threshold*

How does the performance of Degree-discounted symmetrization change as we change the pruning threshold i.e. as more or fewer edges are retained in the graph? We experimented with four different thresholds. The obtained Avg F scores as well as times to cluster are given in Table 5.3.1, for both MLR-MCL and Metis. The trends depicted in the table accord very well with our intuition; as we raise the threshold, there are fewer edges in the graph, and there is a gradual drop in the cluster quality, but which is compensated by faster running times. In fact, even with a threshold of 0.025, and having only 60% as many edges as $A + A^T$, Degree-discounted+MLR-MCL still yields an F score of 21.72 (compared to 20.2 for $A+A^T$) and clusters in 1039 seconds (compared to nearly 23000 seconds for $A + A^T$). The trends are very similar for Metis as well.

These results also suggest that there is no single "correct" pruning threshold. Lower prune thresholds retain more edges in the symmetrized graphs and result in higher clustering accuracies, but on the flip side, take longer to cluster. Higher prune thresholds mean the accuracy may be lower, but the graph is also clustered faster. The user may therefore select a prune threshold according to their computational constraints. One can compute all the similarities corresponding to a small random sample of the nodes, and choose a prune threshold such that the average degree when this threshold is applied to the random sample approximates the final average degree that the user desires. For many real networks, an average degree of 50-150 in the symmetrized graph seems most reasonable, since this is the size of typical clusters in such networks [15].

5.4 Results on Livejournal and Flickr

In Figure 9(a) and (b), we show clustering times using MLR-MCL on the Livejournal and Flickr datasets. We could not evaluate cluster quality for lack of ground truth data. We do not report results on Bibliometric, since it is clear from the number of singletons for that transformation (see Table 2) that it is not viable for such large scale graphs. The trends for these datasets closely mimic the trends in Wikipedia, with Degree-discounted symmetrization once again proving at least two times as fast to cluster as the others at the higher range of the number of clusters. Similar to Wikipedia, the main reason for the faster performance of Degree-discounted symmetrization is the absence of hub nodes in the symmetrized graphs and a clearer cluster structure (the normalized cuts [21, 5] obtained from clustering the Degree-discounted symmetrized graphs are much lower than those obtained using the original graph, indicating the presence of well-separated clusters in the former).

α	β	F-score on Cora	F-score on Wiki
0	0	28.48	9.42
log	log	30.92	19.43
0.25	0.25	30.79	18.13
0.5	0.5	**31.66**	**20.15**
0.75	0.75	29.82	19.97
1.0	1.0	30.58	18.70
0.25	0.50	30.42	19.79
0.25	0.75	31.42	19.52
0.50	0.25	30.51	18.65
0.50	0.75	30.93	20.04
0.75	0.25	30.07	18.42
0.75	0.50	31.07	19.38

Table 4: Effect of varying α, β (Metis). The best results are indicated in bold.

5.5 Effect of varying α and β

We next examine the effect of varying the out-degree discount parameter α and the in-degree discount parameter β. The Avg. F-scores obtained by clustering the symmetrized graph using Metis for a specific configuration of α and β is shown in Table 5.5 (for ease of comparison, the number of clusters is fixed at 70 for Cora and 10,000 for Wikipedia). In both the datasets, the best F-scores are obtained using $\alpha = \beta = 0.5$. However, doing degree-discounting using some configuration of α and β is better than doing no degree discounting at all (shown as $\alpha = \beta = 0$ in Table 5.5).

In fact, using $\alpha = \beta = 0.5$ is similar to using L2-norms for normalizing raw dot-products, as done when computing cosine distance. Spertus et. al. [23] empirically compared six different similarity measures for the problem of community recommendation and found that L2-normalization performed the best. Hence, it is not suprising that $\alpha = \beta = 0.5$ should similarly work well for us across different datasets.

5.6 Significance of obtained improvements

We emphasize that the improvements obtained using Degree-discounted symmetrizations are significant, both in the practical and the statistical sense. MLR-MCL, Graclus and Metis are quite different clustering algorithms, and combining *any* of them with the Degree-discounted symmetrization resulted in significant improvements over baseline approaches in terms of quality (in the range of 10-30%), as well as in terms of clustering time (2-5 times speedup on million-plus node graphs). We also found that the improvements obtained using Degree-discounted symmetrization over the baseline approaches were highly statistically significant. We used the very general *paired binomial sign test* to test the null hypothesis that there is no improvement. The sign test makes no assumptions about the underlying test distribution, and hence is suitable in our situation since we do not actually know the underlying test distribution. It was applied as follows. We count the number of graph nodes that were correctly clustered in one clustering but not in the other clustering (in a paired fashion i.e. each node in one clustering is compared with the same node in the other clustering), and also the other way around. The probability of the obtained counts (or more extreme counts) arising from the null hypothesis, calculated using the binomial distribution with p=0.5, gives us the final p-value. Small p-values tell us that the observed improvements are unlikely to have occured by random chance.

The improvements in clustering accuracy reported above are all highly statistically significant. On Cora, MLR-MCL's improvement using Degree-discounted symmetrization over using $A+A^T$ is significant with p-value 1.0E-312, and the improvement over BestWCut is significant with p-value 1.0E-112. Similarly, the improvement of Graclus using Degree-discounted symmetrization over using $A+A^T$ is significant with p-value 1.0E-36, and the improvement over using BestWCut is significant with p-value 1.0E-44. The improvement of Metis over using BestWCut is significant with p-value 1.0E-79. Coming to Wiki, MLR-MCL's improvement when using Degree-discounted over $A+A^T$ is significant with p-value 1.0E-3367. The improvement for Metis is also significant with p-value 1.0E-22767.

5.7 A case study of Wikipedia clusters

Why exactly does Degree-discounted symmetrization outperform other methods? We give some intuition on this question using examples of Wikipedia clusters that were successfully extracted through this method but not with the other symmetrizations. Note that these example clusters were recovered by both MLR-MCL as well as Metis and is thus independent of the clustering algorithms.

A typical example is the cluster consisting of the plant species belonging to the genus *Guzmania*. The in-links and out-links of this group is shown in Figure 10. Example cluster consisting of plants belonging to the *Guzmania* family. The first notable fact about this cluster is that *none of the cluster members* links to one another, but they all point to some common pages - e.g. "Poales", which is the Order containing the *Guzmania* genus; "Ecuador", which is the country that all of these plants are endemic to; and so on. All group members are commonly pointed to by the *Guzmania* node as well as point to it in return.

Note that this cluster is not an isolated example. Clusters involving lists of objects particularly were found to satisfy a similar pattern to the *Guzmania* cluster. Other examples include *Municipalities in Palencia, Irish cricketers, Lists of birds by country* etc.

These examples provide empirical validation of our hypothesis - laid out in Section 3 and Figure 1 - that in-link and out-link similarity, and not inter-linkage, are the main clues to discovering meaningful clusters in directed graphs.

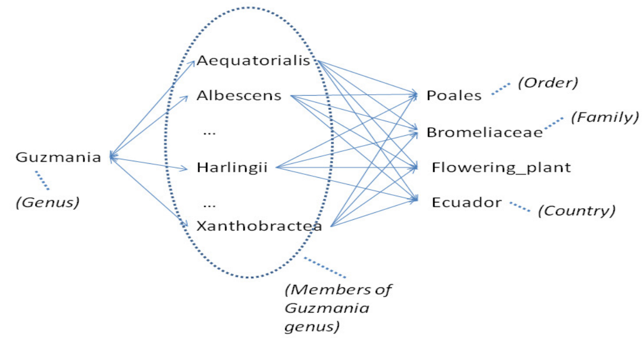


Figure 10: The subgraph of Wikipedia consisting of plant species of the genus *Guzmania* and their in-links and out-links.

5.8 Top-weight edges in Wikipedia symmetrizations

We pick the top-weighted edges in the different symmetrizations of Wikipedia to gain a better understanding into their workings. The top 5 edges from Degree-discounted, Bibliometric and Random Walk symmetrizations are shown in Table 5. Bibliometric heavily weights edges involving hub nodes such as 'Area', 'Population density' etc ('Area' has an in-degree of 71,146, e.g.), as expected. Similarly, Random walk heavily weights edges involving nodes with high Page Rank, which also typically tend to be hub nodes. The top-weighted edges of Degree-discounted, on the other hand, involve non-hub nodes with specific meanings; the particular examples listed in Table 5 are almost duplicates of one another.

6. CONCLUSION

In this article, we have investigated the problem of clustering directed graphs through a two-stage process of symmetrizing the directed graph followed by clustering the symmetrized undirected graph using an off-the-shelf graph clustering algorithm. We presented Random Walk and Bibliometric symmetrizations, drawing upon previous work, and based on an analysis of their weaknesses, presented the Degree-discounted symmetrization. We compared the different symmetrizations extensively on large scale real world datasets w.r.t. both quality and scalability, and found that Degree-discounted symmetrization yields significant improvements in both areas. In future work, we would like to investigate the performance of our proposals in large-scale web scenarios involving the possibilities of spam and link fraud. Extending our approaches to bi-partite and multi-partite graphs also seems to be a promising avenue. Similarly, in addition to evaluation on real data we would like to validate results on synthetically controlled datasets. Unfortunately, we are aware of no synthetic graph generators for producing realistic directed graphs with known ground truth clusters. For instance the Kronecker graph generator [14] allows the generation of realistic directed networks - but does not come associated with a set of clusters with ground truth.

Acknowledgments: We thank the reviewers and Stratis Viglas for their thoughtful comments and suggestions. This work is supported in part by the following grants: NSF CAREER IIS-0347662, RI-CNS-0403342,CCF-0702587 and IIS-

Symmetrization method	Node 1	Node 2	Edge weight
Random walk	Area	Square mile	3354848
	Mile	Square mile	2233110
	Geocode	Geographic coordinate system	1788953
	Degree (angle)	Geographic coordinate system	1766339
	Area	Octagon	1457427
Bibliometric	Area	Population density	2465
	Record label	Music genre	2423
	Population density	Geographic coordinate system	2301
	Square mile	Population density	2129
	Area	Time zone	2120
Degree-discounted	Cyathea	Cyathea (Subgenus Cyathea)	68
	Roman Catholic dioceses in England & Wales	Roman Catholic dioceses in Great Britain	57
	Sepiidae	Sepia (genus)	55
	Szabolcs-Szatmár-Bereg	Szabolcs-Szatmár-Bereg-related topics	53
	Canton of Lizy-sur-Ourcq	Communauté de communes du Pays de l'Ourcq	52

Table 5: Edges with highest weights for different symmetrization methods on the Wikipedia dataset. Note that the edge weights in the rightmost column are normalized by the lowest edge weight in the graph, as the non-normalized weights are incommensurable.

0917070.

7. REFERENCES

[1] R. Andersen, F. R. K. Chung, and K. J. Lang. Local partitioning for directed graphs using pagerank. In *WAW*, pages 166–178, 2007.

[2] R. Bayardo, Y. Ma, and R. Srikant. Scaling up all pairs similarity search. In *Proceedings of the 16th international conference on World Wide Web*, pages 131–140. ACM, 2007.

[3] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2, 2006.

[4] F. Chung. Laplacians and the Cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1):1–19, 2005.

[5] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted Graph Cuts without Eigenvectors: A Multilevel Approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957, 2007.

[6] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon. Pagerank, hits and a unified framework for link analysis. In *SIAM Conference on Data Mining*, 2003.

[7] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, 1999.

[8] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

[9] D. Gleich. Hierarchical Directed Spectral Graph Partitioning. 2006.

[10] J. Huang, T. Zhu, and D. Schuurmans. Web communities identification from random walks. *Lecture Notes in Computer Science*, 4213:187, 2006.

[11] R. Kannan, S. Vempala, and A. Veta. On clusterings-good, bad and spectral. In *FOCS '00*, page 367. IEEE Computer Society, 2000.

[12] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20, 1999.

[13] M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14:10–25, 1963.

[14] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research*, 11:985–1042, 2010.

[15] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *CoRR*, abs/0810.1355, 2008.

[16] C. Manning, P. Raghavan, and H. Schutze. *An introduction to information retrieval*. 2008.

[17] M. Meila and W. Pentney. Clustering by Weighted Cuts in Directed Graphs. In *SDM*, 2007.

[18] M. Meila and J. Shi. A random walks view of spectral segmentation. In *Artificial Intelligence and Statistics AISTATS*, 2001.

[19] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, October 2007.

[20] V. Satuluri and S. Parthasarathy. Scalable graph clustering using stochastic flows: applications to community discovery. In *KDD '09*, pages 737–746, New York, NY, USA, 2009. ACM.

[21] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[22] H. Small. Co-citation in the scientific literature: A new measure of the relationship between documents. *Journal of the American Society for Information Science*, 24:265–269, 1973.

[23] E. Spertus, M. Sahami, and O. Buyukkokten. Evaluating similarity measures: a large-scale study in the orkut social network. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 678–684, New York, NY, USA, 2005. ACM.

[24] D. Zhou, J. Huang, and B. Scholkopf. Learning from labeled and unlabeled data on a directed graph. In *ICML '05*, pages 1036–1043, 2005.

[25] D. Zhou, B. Scholkopf, and T. Hofmann. Semi-supervised learning on directed graphs. *Advances in neural information processing systems*, 17:1633–1640, 2005.