Improving Community Detection with Linguistic Information

Jean Mark Gawron and **Alex Dodge** and **Ming-Hsiang Tsou** and **Brian Spitzberg** and **Li An**
San Diego State University
gawron@mail.sdsu.edu

Abstract

We present a general method for using documents produced by a group for which a social network can be constructed to improve community detection. We augment bag-of-words document representations with "community words" based on the communities assigned by a a community detection algorithm, in counts weighted by the strength of evidence for community membership; we then run community detection again on the similarity graph of the modified documents. We show that combining the linguistic with network information improves significantly on systems using either kind of information alone.

1 Introduction

Real networks are not random; their edge distributions reveal high levels of organization, "with high concentrations of edges within special groups of vertices, and low concentration between these groups." (Fortunato, 2010). This clustering property is called **community structure** (Girvan and Newman, 2002). In this paper we target community detection in social networks, which display hierarchical community structure in abundance, exploring ways to refine the search using linguistic information.

2 Approach

We start with the situation of a group whose members produce documents, with some foundation for inferring social ties among the group members. The foundation could be co-occurrence on a membership list, such as those found on Live Journal, participation in the same mail thread, as on Linux lists, or it could be hyperlinks between Blogs. The inferred ties allow us to build a network, and we now have a network of documents. We will refer to this graph as the **link graph**. Anticipating the case study of science fiction blogs below, we refer to the document producers as bloggers, and to the evidence of social ties as links.

To bring linguistic (or arbitrary) features into play, we propose to run community discovery algorithms on a graph that is somewhat different than the link graph. We define the graph via an NxN similarity matrix **SM**, where $\mathbf{SM}_{ij}$ is the similarity of the feature vector of vertex i to that of vertex j. The use of a similarity matrix as a basis for clustering is familiar in a variety of application, for example in image segmentation, especially connection with normalized cuts (Von Luxburg, 2007).

The simplest way to combine linguistic and network information in **SM** would be to start by representing each document vertex with a classic bag–of-word vector (1 entry for each vocabulary item) and add one link feature for every vertex in the graph, defining the jth link feature for vector i to be 1 if there is a link between i and j.

There are three problems with combining the information from words and links in this way. Problem one is **sparsity**: Throwing a large set of links into the feature set adds another diffuse set of signals in to a very noisy background. Problem two is **signal strength**: treating links like other words does not properly weight link information in an unsupervised learning setting where the goal is community detection. Finally, there is the problem of **relevancy**. Words contain too much information, most of it irrelevant to the problem of community detection.

We turn to the last problem first. We approximate the set of features most likely to yield information about social ties by using proper names, limited to names of persons and organizations. The choice of proper names is motivated both by the fact that they can be reliably extracted and by anecdotal evidence. Adamic and Glance (2005) noted distinct patterns of name dropping in their data for conservative and liberal blogs. Certain names, particularly names of new organizations and political figuresm showed up and over and over in each group, and interestingly, these were often outgroup citations. Thus, for example, conservative bloggers were more likely to cite the liberal New York Times than were liberal bloggers.

We address the related problems of sparsity and signal strength with two strategies:

1. We concentrate the link signals by reducing all link signals to a single feature: Call this feature the **community word**. Thus if initial community detection discovers 3 communities, one of 3 community words is added to the bag of names of each blogsite s.

2. The count c assigned to the community word for blogsite s is equal either to 1 or to the number of s's neighbors in the same community, whichever is greater. [1]

Finally a few words about the community detection algorithms used in our experiments. We focus on two algorithms which rank community assignments using the **modularity** of the community-partition, the algorithms of Blondel et al. (2008) and Newman (2006). Modularity can be defined as a sum of edge weights, ignoring edges between communities; algorithms that are sensitive to edge weights are particularly attractive, since our approach seems to work best when the similarity graph is fairly dense graph (high values of k) with significant variability in weights. Among several definitions of modularity given in the literature, we choose the definition common to the two algorithms above (Newman, 2006):

$$\mathrm{Q} = \frac{1}{4m} \sum_{i,j} \left[\mathrm{A}_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where A_{ij} represents the weight of the edge between vertices i and j; k_i is the weighted degree of vertex i (the sum of the weights of the edges attached to vertex i), c_i, c_j are the communities to which vertex i and j are assigned; and the δ-function $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise; and m is the number of edges in the graph. To search for maximally modular partitions of the graph into two communities, Newman defines a modularity matrix $\mathbf{B}$, whose i, j entry contains the contribution of edge i, j to the modularity sum:

$$\mathbf{B}_{ij} = \mathbf{A}_{ij} - \frac{k_i k_j}{2m}$$

with the modularity then computable as

$$\frac{1}{4m} \mathbf{s}^T \mathbf{B} s,$$

if s is a vector such that $s_i = 1$ if node i belongs to community 1, and -1 if it belongs to community 2. Newman observes that a good approximation of the maximum modularity value of s is acheived by choosing it to maximize the dot product with the first eigenvector of B. The resulting split is then refined by using it as the starting partition for a Kernighan-Lin like procedure (Kernighan and Lin, 1970) which settles on a reassignment of communities that maximizes ΔQ. Further splits are then attempted recursively on each community until overall modularity can no longer be improved. Like the original Kernighan-Lin algorithm, the procedure is sensitive to the order in which vertices are considered.

The algorithm of Blondel et al. (which we will, following convention, refer to as the Louvain algorithm) maximizes the same quantity but through a two-phase agglomerative approach. In the first phase, nodes all start out in separate communities, then the mergers that maximize ΔQ are made, and merging continues until modularity can no longer be improved. In the second phase, communities are disovered in the first phase are converted into nodes in a new graph, with recomputed weights. Then the

[1] A closely related approach is to weight c by the "community centrality" score of site s, as defined for Newman's algorithm. This approach achieves comparable results, but is not described in this study.

process repeats until the first phase can make no improvement.

Attractive features of the algorithms include:

1. No k is chosen in advance. The algorithms decide on the number of communities based on a procedure which terminates when modularity can no longer be increased. Both algorithms may return a trivial solution in which there is only one community. They are thus appropriate for *community discovery*. This is in contrast to many clustering procedures such as K-means, for which k must be chosen in advance.

2. In Newman's algorithm, the leading Eigenvector of modularity matrix $\mathbf{B}$ contains information. The positions with the highest absolute values correspond to vertices concentrating the greatest mass of community evidence. If e is the leading eigenvector, we refer to $|e_i|$ as the **community centrality** of site i.

3 Data

The data for our case study was collected in July of 2015 from 218 sites in the science fiction blogging community with the aim of tracking an ongoing fracture in the community known as **puppygate**. Fourteen seed webpages were chosen by internet search on the term "puppygate"; the dataset was extended by following links. Sites were labeled by hand according to which one of two sides in the debate they supported. We will refer to the two sides as the "puppies" and the "others".

The "Puppygate" controversy concerns the alleged hijacking of the science fiction community's annual award (the "Hugo" Award) by a group of conservative writers, fans, and publishers who circulated slates of eligible works, urging their readers and followers to read and consider nominating those works. Two slates in particular, known as the "sad" and "rabid" puppies slates, attracted considerable attention in 2014, earning a number of nominations; and in 2015, after a similar campaign, 18 of the 20 Hugo finalists came from one of the two "puppy" slates. Because the "puppies" (in particular "sad puppy" author Larry Correia) claimed they were reacting against a nomination process already firmly in the control of a cabal of liberal "Social Justice Warriors", the controversy immediately took on a "GamerGatish" color and polarizing tone, with political correctness being one of the main topics of discussion. Many of the texts collected are vitriolic, and the fact that the controversy also attracted the attention of major media players such as Entertainment Weekly, the Telegraph (in the U.K.), and Salon, probably made things worse.

This dataset provides an ideal test case for our approach. It contains online texts produced by a community in fracture, with hyperlinks that can be used to infer social ties; moreover, the blog documents themselves very clearly establish community affiliation. An additional feature of interest is that cross-community links are quite common, often because bloggers embroiled in the debate are quoting sentiments they abhor. Almost exactly 2/3 of the edges are between nodes on the same side of the controversy (758 out of 1142), but that means 1/3 span between communities, making it more difficult for a community detection algorithm operating on link information alone to identify the fracture. This may be one reason why linguistic information was able to make a substantial contribution in this case.

4 Methodology

We compare the following three systems:

Link: This **baseline system** uses only community detection on the original link graph.

Ling: This system uses the full pipeline described below, but only with proper names features

LingLink: This full pipeline system combines name features with community features found by running community detection on the link graph.

Ling and LingLink share the following pipeline:

1. Names are extracted using the Stanford Named Entity Extractor (Finkel et al., 2005). Only person and organization names are kept, with each site represented as bag of names. If community word features are being used. they are added to the name bags.

2. Feature weights are assigned using Pointwise Mutual Information (Church and Hanks, 1990). Features are filtered using PMI scores

to the *FeatNum* best features, resulting in an Nx*FeatNum* data matrix DM.

3. Dimensionality reduction using SVD maps from Nx*FeatNum* matrix DM to Nx*NumDim* matrix RDM. We had our best success using the scikit-learn implementation (Pedregosa et al., 2011) of the stochastic SVD algorithm of Halko et al. (2009).
4. A similarity matrix SM is constructing from the reduced data matrix RDM using cosine as the similarity measure. For the associated graph SG, an edge exists between i and j if either j is among the *top_n* most similar nodes to i or i is among the *top_n* most similar nodes to j, with the proviso that no edge can exist between vertices with similarity 0.
5. Dectect communities on similarity graph SG.

5 Results

Table 1 gives the results of our experiments for the Ling and LingLink systems using Newman and Louvain for community detection. Numbers shown are all AMI scores x 1000, and ±-values show the 95% confidence half-intervals after 10 runs of each system. Indeterminacy was introduced by our stochastic SVD implementation, as well as by Newman, because of the Kernighan-Lin like refinement step.

Table 1 shows that a system combining Linguistic and network information improves significantly on the baseline; note that almost all of the LingLink systems improve on their Ling counterparts, demonstrating fairly robustly that these linguistic features alone cannot converge on a solution of the community problem in an unsupervised setting.

			Newman	
TopN	NFeat	NDim	Ling	LingLink
50	2K	3	118±05	158±10
		7	94±10	157±28
		25	88±14	146±16
	5K	3	127±23	120±17
		7	102±18	138±18
		25	65±13	103±19
75	2K	3	110±13	**292±40**
		7	106±11	180±46
		25	38±34	131±24
	5K	3	114±18	151±26
		7	123±09	145±37
		25	10±22	56±33

			Louvain	
TopN	NFeat	NDim	Ling	LingLink
50	2K	3	117±12	133±19
		7	79±08	126±17
		25	57±09	91±13
	5K	3	83±17	128±11
		7	110±14	129±10
		25	51±13	51±09
75	2K	3	129±11	118±25
		7	86±8	129±30
		25	65±13	96±12
	5K	3	127±21	182±28
		7	107±13	124±9
		25	58±14	58±11

Table 1: Upper table: AMI scores for the full pipeline systems with Newman. The baseline Link system achieved an AMI of .170 with a 95% confidence half-interval of 0, because the Newman algorithm found the same community on all runs. Lower table: same systems with Louvain.

6 Conclusion

Our results show that the best Newman-algorthm based system gives a significant performance boost over the baseline system in reconstructing the communities in this fracture.

It is worth noting that the use of a community detection algorithm does more than reconstruct communities; it also discovers new subcommunities and, in the case of Newman's algorithm, yields centrality centrality assignments, potentially moving us toward a better understanding of the the players, the issues, and future fractures. Many of our high-scoring runs produce 3-community solutions of considerable interest, most of which divided the puppies up into two groups, corresponding to early and late arrivals in the controversy. In addition, high-centrality participants were mostly well-known figures in the debates, with some interesting surprises. These are analytical opprotunities well worth refining.

References

Lada A. Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the third international workshop on Link discovery*, pages 36–43. ACM.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):10008.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370. Association for Computational Linguistics.

S. Fortunato. 2010. Community detection in graphs. *Physics Reports*, 486(3):75–174.

Michelle Girvan and Mark EJ Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

N. Halko, J. A. Tropp, and P. G. Martinsson. 2009. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. *ACM Report*, 5.

Brian W Kernighan and Shen Lin. 1970. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307.

M.E.J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, October.

Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.