Fast Detection of Overlapping Communities via Online Tensor Methods

Furong Huang
Electrical Engineering and Computer Science
University of California, Irvine
Irvine, CA 92697-2625.
furongh@uci.edu

Niranjan U N
Information and Computer Science
University of California, Irvine
Irvine, CA 92697-2625.
un.niranjan@uci.edu

Mohammad Umar Hakeem
Electrical Engineering and Computer Science
University of California, Irvine
Irvine, CA 92697-2625.
mhakeem@uci.edu

Animashree Anandkumar
Electrical Engineering and Computer Science
University of California, Irvine
Irvine, CA 92697-2625.
a.anandkumar@uci.edu

Abstract

We present a fast tensor-based approach for detecting hidden overlapping communities under the mixed membership stochastic block (MMSB) model. We present two implementations, viz., a GPU-based implementation which exploits the parallelism of SIMD architectures and a CPU-based implementation for larger datasets, where the GPU memory does not suffice. Our GPU-based implementation involves a careful optimization of storage and communication, while in our CPU-based implementation, we perform sparse linear algebraic operations to exploit the data sparsity. We use stochastic gradient descent for multilinear spectral optimization and this allows for flexibility in the tradeoff between node sub-sampling and accuracy of the results. We validate our results on datasets from Facebook, Yelp and DBLP, where ground truth is available, using notions of p-values and false discovery rates, and obtain high accuracy for membership recovery. We compare our results, both in terms of execution time and accuracy, to the state-of-the-art algorithms such as the variational method, and report many orders of magnitude gain in the execution time. For instance, for DBLP dataset with about a million nodes and 16 million edges, the execution time is about two minutes.

1 Summary of Contributions

Studying community formation is an important problem in social networks. A community generally refers to a group of individuals with shared interests or beliefs (e.g. music, sports, religion), or relationships (e.g. friends, co-workers). In a social network, we can typically observe and measure the interactions among the actors, but not the communities they belong to. A challenging problem is then to estimate the communities of the actors using only their observed interactions. In general, actors can participate in multiple communities, and detecting overlapping communities is even more challenging. Our goal is to design algorithms which can accurately detect overlapping communities, and yet be easily parallelizable for fast and scalable implementation on large graphs with millions of nodes. Moreover, we learn a probabilistic community model which allows us to carry out prediction tasks such as link classification.

In this work, we present a fast approach for detecting overlapping communities under the mixed membership stochastic block model (MMSB) [1]. It is based on estimating tensors from subgraph counts such as 3-stars in the observed network, and then performing linear algebraic operations (e.g. SVD), and an iterative stochastic gradient descent method for tensor decomposition using implicit trilinear operations. We present two implementations, viz., a GPU-based implementation which exploits the parallelism of SIMD architectures and a CPU-based implementation for larger datasets, where the GPU memory does not suffice.

The running time of our method is $O(n + k^3)$ using nk cores in the parallel computation model [8], where n is the number of nodes and k is the number of communities. Since $k \ll n$, we have a linear running time in the number of nodes, which makes it scalable for extremely large networks.

We first describe our GPU implementation below. We carry out a careful GPU-based code optimization and design of data structures for efficient heterogeneous storage on both CPU and GPU memories, and minimize CPU-GPU data transfers to obtain speedups. A naive implementation of the tensor method would result in a huge space requirement since it requires the manipulation of an $O(n) \times O(n) \times O(n)$ tensor, where n is the number of nodes, and can also result in poor scaling of running time due to communication costs between CPU and GPU. In contrast, we never explicitly construct the $O(n) \times O(n) \times O(n)$ tensor; we carry out dimensionality reduction in the preprocessing stage and then manipulate a $k \times k \times k$ tensor, where k is the number of communities. We carry out the decomposition of this tensor implicitly, without forming it, through stochastic gradient descent. Moreover, we convert the stochastic gradient update steps to matrix and vector operations, and implement them using the GPU *device interface*, so as to reduce the GPU-CPU transfer overhead, and obtain a bigger speed-up. Thus, we present an extremely fast community detection method on GPUs.

We now describe our second implementation on the CPU. This was carried out to overcome the memory limitations of the GPU for extremely large datasets consisting of millions of nodes. We manipulate the data in the sparse format, consisting of sparse multiplications and sparse Lanczos SVD. Furthermore, we implement randomized methods for dimensionality reduction [6]. We obtain tremendous gains in terms of both the running time and the memory required to run on datasets with millions of nodes. We observe that while our GPU implementation is efficient for denser graphs (e.g. Facebook) with larger number of communities, our CPU implementation is extremely efficient for sparse graphs (e.g. Yelp and DBLP) with large number of nodes.

We recover hidden communities across many real and synthetic datasets with high accuracy. When ground-truth communities are available, we propose a new error score based on the hypothesis testing methodology involving p-values and false discovery rates [15] to validate our results. Although these notions are standard in statistics (and mainly bio-statistics), its use in social network analysis is limited. The use of p-values eliminates the need to carefully tune the number of communities output by our algorithm, and we obtain a flexible tradeoff between the fraction of communities recovered and their estimation accuracy. We also provide arguments to show that the normalized mutual information (NMI) and other scores, previously used for evaluating recovery of overlapping community, can underestimate the errors and in fact, the overlapping version of NMI does not reduce to the information-theoretic (non-overlapping version of) NMI, and is thus incorrect.

We find that our method has very good accuracy on a range of datasets: Facebook, Yelp and DBLP. For the Facebook friendship dataset, consisting of around 0.7 million edges, 20000 nodes and 360 communities, the average error in estimating the community memberships is below 5% and the running time is 35 seconds. For the dataset consisting of Yelp reviews, consisting of around 0.6 million edges, 40000 nodes and 159 communities, the error is below 10% and the method runs under 10 seconds. On a much larger DBLP collaborative dataset, consisting of 16 million edges, 1 million nodes and 250 communities, with an error of about 10% and the method runs in about 2 mins, excluding the 80 minutes taken to read the edge data from the files.

Compared to the state-of-art method for learning MMSB models using stochastic variational inference algorithm of [7], we obtain several orders of magnitude speedup in running times on multiple real datasets. This is because our method consists of efficient matrix operations which are *embarassingly parallel* and can also be manipulated in sparse format, which is especially efficient for social network settings involving sparse graphs. Moreover, our code is flexible to run on a range of graphs such as directed, undirected and bipartite graphs, while the code of [7] is designed for homophilic networks, and cannot handle bipartite graphs, needed for us to recover communities in a recommendation setting such as the Yelp dataset.

Although there have been fast implementations for community detection before [14, 9], these methods are not statistical and do not yield descriptive statistics such as bridging nodes [11], and cannot perform predictive tasks such as link classification, which are the main strengths of the MMSB model. With the implementation of our tensor-based approach, we record huge speed-ups compared to existing approaches for learning the MMSB model. Thus, we obtain good accuracy as well as fast running times with our method.

2 Main results

We describe the results on real datasets(summary of datasets in Table 1) in details below and in Table 2.

	Facebook	Yelp	DBLP sub	DBLP
$\|E\|$	766,800	672,515	5,066,510	16,221,000
$\|V\|$	18,163	10,010+28,588	116,317	1,054,066
GD	0.004649	0.000903	0.000749	0.000029
k	360	159	250	6,003
AB	0.5379	0.4281	0.3779	0.2066
ADCB	47.01	30.75	48.41	6.36

Table 1: Summary of real datasets used in our paper. $|V|$ is the number of nodes in the graph, $|E|$ is the number of edges, GD is the graph density given by $\frac{2|E|}{|V|(|V|-1)}$, k is the community number, AB is the average bridgeness and ADCB is the average degree-corrected bridgeness.

Data	Method	$\widehat{k}$	Thre	$\mathcal{E}$	$\mathcal{R}(\%)$	Time(s)
FB	Ten(sparse)	10	0.10	0.063	13	35
	Ten(sparse)	100	0.08	0.024	62	309
	Ten(sparse)	100	0.05	0.118	95	309
	Ten(dense)	100	0.100	0.012	39	190
	Ten(dense)	100	0.070	0.019	100	190
	Variational	100	–	0.070	100	$10,795$
	Ten(dense)	500	0.020	0.014	71	468
	Ten(dense)	500	0.015	0.018	100	468
	Variational	500	–	0.031	100	$86,808$
YP	Ten(sparse)	10	0.10	0.271	43	10
	Ten(sparse)	100	0.08	0.046	86	287
	Ten(dense)	100	0.100	0.023	43	$1,127$
	Ten(dense)	100	0.090	0.061	80	$1,127$
	Ten(dense)	500	0.020	0.064	72	$1,706$
	Ten(dense)	500	0.015	0.336	100	$1,706$
DB sub	Ten(dense)	100	0.15	0.072	36	$7,664$
	Ten(dense)	100	0.09	0.260	80	$7,664$
	Variational	100	–	7.453	99	$69,156$
	Ten(dense)	500	0.10	0.010	19	$10,157$
	Ten(dense)	500	0.04	0.139	89	$10,157$
	Variational	500	–	16.38	99	$558,723$
DB	Ten(sparse)	10	0.30	0.103	73	4716
	Ten(sparse)	100	0.08	0.003	57	5407
	Ten(sparse)	100	0.05	0.105	95	5407

Table 2: Yelp, Facebook and DBLP main quantitative evaluation of ”tensor method” vs ”variational method” : $\widehat{k}$ is the community number specified to our algorithm, Thre is the threshold for picking significant estimated membership entries, $\mathcal{E}$ is the classification error per node per community, $\mathcal{R}$ is the fraction of ground truth communities recovered. Refer to Table 1 for statistics of the datasets.

Recovery ratio is defined as $\mathcal{R} := \frac{1}{k} \sum\limits_{(i,j)\in E_{\{\mathsf{P}_{\text{val}}\}}} 1$ for all $i \in [k]$ and $j \in [\widehat{k}]$, where P_{val} denotes the p-value and $E_{\{\mathsf{P}_{\text{val}}\}}$ denotes the pairings between ground truth and estimated communities which has p-values smaller than 0.01. The perfect case is that all the memberships have at least one significant overlapping estimated membership, giving a recovery ratio of 100%. The average error function is defined as $\mathcal{E} := \frac{1}{k} \sum\limits_{(i,j)\in E_{\{\mathsf{P}_{\text{val}}\}}} \left\{ \frac{1}{n} \sum\limits_{x\in|X|} \left| \widehat{\Pi}_i(x) - \Pi_j(x) \right| \right\}$,

$\forall i \in [k],\ j \in [\widehat{k}]$, where P_{val} denotes the p-value. The results are presented in Table 2. We note that our method in both dense and sparse implementations performs very competitively compared to the state-of-art variational method. For the Yelp dataset, we have a bipartite graph where the business nodes are on one side and user nodes on the other and use the review stars as the edge weights. In this bipartite setting, the variational code provided by Gopalan et. al [7] does not work on since it is not applicable to non-homophilic models. Our approach does not have this restriction. Note that we employ our dense implementation (on GPU) to run experiments with large number of communities k, since the GPU is SIMD architecture thus is fast at the STGD step . On the other hand, sparse implementation on CPU is significantly faster and memory efficient in sparse graphs (and under small k), while dense format on GPU is faster for denser graphs such as Facebook. Note that data reading time for DBLP is 4700 seconds, which is not negligible compared to other datasets(usually within a few seconds), our method actually executes within 2 minutes for $k = 10$ and within 10 minutes for $k = 100$.

Business	**RC**	**Categories**
Four Peaks Brewing Co	735	Restaurants, Bars, American (New), Nightlife, Food, Pubs, Tempe
Pizzeria Bianco	803	Restaurants, Pizza, Phoenix
FEZ	652	Restaurants, Bars, American (New), Nightlife, Mediterranean, Lounges, Phoenix
Matt's Big Breakfast	689	Restaurants, Phoenix, Breakfast& Brunch
Cornish Pasty Company	580	Restaurants, Bars, Nightlife, Pubs, Tempe
Postino Arcadia	575	Restaurants, Italian, Wine Bars, Bars, Nightlife, Phoenix
Cibo	594	Restaurants, Italian, Pizza, Sandwiches, Phoenix
Phoenix Airport	862	Hotels & Travel, Phoenix
Gallo Blanco Cafe	549	Restaurants, Mexican, Phoenix
The Parlor	489	Restaurants, Italian, Pizza, Phoenix

Table 3: Top 10 bridging businesses in Yelp and categories they belong to. "RC" denotes review counts for that particular business.

We select top 10 categories recovered with the lowest error and report the business with highest weights in $\widehat{\Pi}$. Among the matched communities, we find the business with the highest membership weight (Table 4). We can see that most of the "top" recovered businesses are rated high. Many of the categories in the top-10 list are restaurants as they have a large number of reviewers. Our method can recover restaurant category with high accuracy, and the specific restaurant in the category is a popular result (with high number of stars). Also, our method can also recover many of the categories with low review counts accurately like hobby shops, yoga, churches, galleries and religious organizations which are the "niche" categories with a dedicated set of reviewers, who mostly do not review other categories.

Category	Business	Star(B)	Star(C)	RC(B)	RC(C)
Latin American	Salvadoreno	4.0	3.94	36	93.8
Gluten Free	P.F. Chang's	3.5	3.72	55	50.6
Hobby Shops	Make Meaning	4.5	4.13	14	7.6
Mass Media	KJZZ 91.5FM	4.0	3.63	13	5.6
Yoga	Sutra Midtown	4.5	4.55	31	12.6
Churches	St Andrew Church	4.5	4.52	3	4.2
Art Galleries	Sette Lisa	4.5	4.48	4	6.6
Libraries	Cholla Branch	4.0	4.00	5	11.2
Religious	St Andrew Church	4.5	4.40	3	4.2
Wickenburg	Taste of Caribbean	4.0	3.66	60	6.7

Table 4: Most accurately recovered categories and businesses with highest membership weights for the Yelp dataset. "Star(B)" denotes the review stars that the business receive and "Star(C)", the average review stars that businesses in that category receive. "RC(B)" denotes the review counts for that business and "RC(C)" , the average review counts in that category.

The top bridging nodes recovered by our method for Yelp dataset are given in the Table 3. The bridging nodes have multiple attributes: typically the type of business and location. In addition, the categories may also be hierarchical:

within restaurants, different cuisines such as Italian, American or Pizza are recovered by our method. Moreover, restaurants which also function as bars or lounges are also recovered as top bridging nodes in our method. Thus, our method can recover multiple attributes for the businesses efficiently.

For Facebook dataset, top 10 communities recovered with lowest error consist of certain high schools, second majors and dorms/houses. We see that high school attributes are easiest to recover and second major and dorm/house are reasonably easy to recover by looking at the friendship relations in Facebook. This is reasonable: college students from the same high school have a high probability of being friends; so do colleges students from the same dorm. For the DBLP subsampled dataset[1], the performance of our algorithm is summarized in Table 2. For the full DBLP dataset, we used Eigen Sparse library and scaled up to a million nodes.

References

[1] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, June 2008.

[2] A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A Tensor Spectral Approach to Learning Mixed Membership Community Models. In *Conference on Learning Theory (COLT)*, June 2013.

[3] Raman Arora, Andrew Cotter, Karen Livescu, and Nathan Srebro. Stochastic optimization for pca and pls. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 861–868, 2012.

[4] Grey Ballard, Tamara Kolda, and Todd Plantenga. Efficiently computing tensor eigenvalues on a gpu. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 1340–1348. IEEE, 2011.

[5] Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. *arXiv:1210.3335*, 2012.

[6] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. *CoRR*, abs/1207.6365, 2012.

[7] P. Gopalan, D. Mimno, S. Gerrish, M. Freedman, and D. Blei. Scalable inference of overlapping communities. In *Advances in Neural Information Processing Systems 25*, pages 2258–2266, 2012.

[8] Joseph JáJá. *An introduction to parallel algorithms*. Addison Wesley Longman Publishing Co., Inc., 1992.

[9] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[10] F. McSherry. Spectral partitioning of random graphs. In *FOCS*, 2001.

[11] Tamás Nepusz, Andrea Petróczi, László Négyessy, and Fülöp Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77(1):016107, 2008.

[12] Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of mathematical analysis and applications*, 106(1):69–84, 1985.

[13] Martin D Schatz, Tze Meng Low, Robert A van de Geijn, and Tamara G Kolda. Exploiting symmetry in tensors for high performance. *arXiv:1301.7744*, 2013.

[14] Jyothish Soman and Ankur Narang. Fast community detection algorithm with gpus and multicore architectures. In *Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International*, pages 568–579. IEEE, 2011.

[15] Korbinian Strimmer. fdrtool: a versatile r package for estimating local and tail area-based false discovery rates. *Bioinformatics*, 24(12):1461–1462, 2008.

[16] Yu Zhang and Dit-Yan Yeung. Overlapping community detection via bounded nonnegative matrix tri-factorization. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 606–614, 2012.

[1] `http://dblp.uni-trier.de/xml/Dblp.xml`