Node Clustering in Graphs: An Empirical Study

Ramnath Balasubramanyan
Carnegie Mellon University
Pittsburgh, PA 15213
rbalasub@cs.cmu.edu

Frank Lin
Carnegie Mellon University
Pittsburgh, PA 15213
frank@cs.cmu.edu

William W. Cohen
Carnegie Mellon University
Pittsburgh, PA 15213
wcohen@cs.cmu.edu

Abstract

Modeling networks is an active area of research and is used for many applications ranging from bioinformatics to social network analysis. An important operation that is often performed in the course of graph analysis is node clustering. Popular methods for node clustering such as the normalized cut method have their roots in graph partition optimization and spectral graph theory. Recently, there has been increasing interest in modeling graphs probabilistically using stochastic block models and other approaches that extend it. In this paper, we present an empirical study that compares the node clustering performances of state-of-the-art algorithms from both the probabilistic and spectral families on undirected graphs. Our experiments show that no family dominates over the other and that network characteristics play a significant role in determining the best model to use.

1 Introduction

Much recent work on the problem of clustering graphs can be broadly classified into either the probabilistic family or the spectral clustering family. Probabilistic approaches posit a generative model for graphs or equivalently the adjacency matrix of the graph (an excellent survey on probabilistic models is found in [1]). Spectral clustering and other graph partition methods use insights about the nature of networks to tackle specific applications in graph analysis and work well empirically. While there has been some work [2] in constructing a theoretical connection between spectral clustering and stochastic block models, there has been relatively little work in comparing the strengths and weaknesses of the two broad approaches. Here we empirically compare these broad approaches by evaluating recent algorithms in each family on a range of sample datasets. From the spectral family, we use the normalized cut method (NCut) [3] and the Ng-Jordan-Weiss algorithm (NJW) [4] and the recently proposed power iteration clustering (PIC) method [5]. We compare these to the sparse block model of Parkkinen et al. (PSK) [6].

2 Spectral methods

Spectral clustering methods were introduced to the machine learning community as elegant solutions to graph partition problems, where the objective is to make a *graph cut* (a bisection of the graph). This is usually done by (1) defining a graph Laplacian with a graph cut objective in mind, (2) finding the "significant" eigenvector of the Laplacian (e.g., the second smallest eigenvector for the normalized cut objective), and (3) thresholding the eigenvector. Nodes corresponding to elements of the eigenvector above the threshold belong to one partition, and those below belong to the other.

A popular way to generalize these methods to create k partitions(clusters) is to find the k (or $k-1$ in the case of NCut) most significant eigenvectors of the graph Laplacian, embed the data points in the space spanned by these eigenvectors, and run k-means to produce the final clusters (partitions).

In this paper we pick two of the most popular spectral clustering methods for our purposes, NCut and NJW; their graph Laplacians are defined as $I - D^{-1}A$ and $I - D^{-1/2}AD^{-1/2}$, respectively.

A is the matrix form of the graph where $A(i,j)$ is the edge weight between node i and j; I is the identity matrix; D is the diagonal degree matrix where $D(i,i) = \sum_j A(i,j)$.

Similarly to NCut and NJW, Power iteration clustering (PIC) [5] also embeds nodes in a space defined by eigenvectors of the graph matrix, and produces clusters via k-means in the embedded space. However, individual eigenvectors are never explicitly calculated. Instead, PIC performs the power iteration *with early stopping* on an arbitrary initial vector using the normalized graph matrix $D^{-1}A$ to produce a vector that is a weighted combination of the significant eigenvectors as a one-dimensional embedding for the nodes. The early stopping criterion proposed in [5] is *acceleration*, based on the observation that the power iteration converges at an accelerated pace in the beginning, and later at a constant pace when non-significant eigenvectors (those with small corresponding eigenvalues) are no longer contributing to this weighted combination. The basic PIC algorithm is shown in Figure 1.

Input: A row-normalized matrix $W = D^{-1}A$ and the number of clusters k.
Output: Clusters $C_1, C_2, ..., C_k$.

1. Pick a random initial vector $\mathbf{v^0}$.
2. Set $\mathbf{v^{t+1}} \leftarrow W\mathbf{v^t}$ and $\boldsymbol{\delta}^{t+1} \leftarrow |\mathbf{v^{t+1}} - \mathbf{v^t}|$.
3. Increment t and repeat above step until $|\boldsymbol{\delta}^t - \boldsymbol{\delta}^{t-1}| \simeq \mathbf{0}$.
4. Use k-means to cluster points on $\mathbf{v^t}$ and return clusters $C_1, C_2, ..., C_k$.

Figure 1: The PIC algorithm.

A shortcoming of one-dimensional embeddings is that, as k becomes large, the one-dimensional embedding assigned to nodes from different clusters are increasingly likely be similar. One way to avoid such "collisions" is to set the initial vector such that nodes in different clusters likely to have different initial values–for instance, letting the initial vector map a node some function of its degree. Another way to avoid collisions is to run power iteration with early stopping d times ($d << k$) and embed the nodes in the d-dimensional space spanned by these vectors. In this paper we choose three variations of PIC for comparison: PIC_R (one random initial vector), PIC_D (initial vector set according to the diagonal of D), and PIC_{R4} (four random initial vectors).

3 A probabilistic model for sparse networks

In this section, we present the sparse network model which borrows concepts from topic models, based on the model introduced by Parkkinen et al. [6]. Figure 2 shows the plate figure for the model that generates a graph representing entity-entity links with an underlying block structure. Clusters in this model are represented as distributions over nodes. Linked entities (i.e. edges) are generated from cluster specific node distributions conditioned on the cluster pairs sampled for the edges. Cluster pairs for edges(links) are drawn from a multinomial defined over the Cartesian product of the cluster et with itself. Vertices in the graph representing nodes therefore have mixed memberships in clusters. Let K be the number of latent clusters(topics) we wish to recover. The generative process to obtain links in the graph is as follows.

1. Generate cluster distributions:

For each cluster $z \in 1, \ldots, K$, sample $\beta_z \sim$ Dirichlet(γ), the cluster specific node distribution.

2. Generate the link matrix of entities:

- Sample $\pi_L \sim$ Dirichlet(α_L) where π_L describes a distribution over the Cartesian product of clusters with itself, for links in the dataset.
- For every link $e_{i1} \rightarrow e_{i2}$, $i \in \{1 \cdots N_L\}$:
  - Sample a cluster pair $\langle z_{i1}, z_{i2} \rangle \sim$ Multinomial(π_L)
  - Sample $e_{i1} \sim$ Multinomial($\beta_{z_{i1}}$)
  - Sample $e_{i2} \sim$ Multinomial($\beta_{z_{i2}}$)

In contrast to MMSB[7], this model only generates realized links that are observed, making this model better suited to sparse graphs.

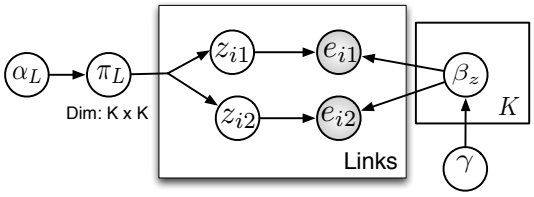

Figure 2: Sparse network model.

Given the hyperparameters α_L and γ, the joint distribution over the links, the cluster pair distribution and cluster assignments for edges is given by

$$p(\pi_L, \boldsymbol{\beta}, \langle \mathbf{z_1}, \mathbf{z_2} \rangle, \langle \mathbf{e_1}, \mathbf{e_2} \rangle | \alpha_L, \gamma) \propto \prod_{z=1}^{K} \text{Dir}(\beta_z | \gamma) \times \text{Dir}(\pi_L | \alpha_L) \prod_{i=1}^{N_L} \pi_L^{\langle z_{i1}, z_{i2} \rangle} \beta_{z1}^{e_{i1}} \beta_{z2}^{e_{i2}} \tag{1}$$

A commonly required operation when using latent variable models is to perform inference on the model to query the latent variable distributions and the cluster assignments of documents and links. Due to the intractability of exact inference in model, a collapsed Gibbs sampler is used to perform approximate inference. It samples a cluster pair for every link conditional on cluster pair assignments to all other links after collapsing π_L using the expression:

$$\begin{aligned} p(\mathbf{z_i} = \langle z_1, z_2 \rangle | \langle e_{i1}, e_{i2} \rangle, \mathbf{z}^{\neg \mathbf{i}}, \langle \mathbf{e_1}, \mathbf{e_2} \rangle^{\neg \mathbf{i}}, \alpha_L, \gamma) \\ \propto \left(n^{L \neg i}_{\langle z_1, z_2 \rangle} + \alpha_L \right) \times \frac{\left(n^{\neg i}_{z_1 e_{i1}} + \gamma \right) \left(n^{\neg i}_{z_2 e_{i2}} + \gamma \right)}{\left(\sum_e n^{\neg i}_{z_1} + |E|\gamma \right) \left(\sum_e n^{\neg i}_{z_2} + |E|\gamma \right)} \end{aligned} \tag{2}$$

E refers to the set of nodes in the graph. The n's are counts of observations in the training set, where n_{ze} is the number of times an entity e is observed under cluster z and $n^L_{\langle z_1, z_2 \rangle}$ the count of links assigned to cluster pair $\langle z_1, z_2 \rangle$.

The cluster multinomial parameters and the cluster pair distributions of links are easily recovered using their MAP estimates after inference using the counts of observations.

$$\beta_z^{(e)} = \frac{n_{ze} + \gamma}{\sum_{e'} n_{ze'} + |E|\gamma}, \quad \pi_L^{\langle z_1, z_2 \rangle} = \frac{n^L_{\langle z_1, z_2 \rangle} + \alpha_L}{\sum_{z_1', z_2'} n^L_{\langle z_1', z_2' \rangle} + K^2 \alpha_L}$$

A de-noised form of the entity-entity link matrix can also be recovered from the estimated parameters of the model. Let B be a matrix of dimensions $K \times |E|$ where row $k = \beta_k$, $k \in \{1, \cdots, K\}$. Let Z be a matrix of dimensions $K \times K$ s.t $Z_{p,q} = \sum_{i=1}^{N_L} \mathbf{I}(z_{i1} = p, z_{i2} = q)$. The de-noised matrix M of the strength of association between the entities in E is given by $M = B^T Z B$.

For experiments with the probabilistic models, we place priors which favor diagonal blocks over off-diagonal blocks. In the sparse model, this is achieved by using a non-symmetric Dirichlet for α_L. In the MMSB model, the beta priors for the per-block binomials are set such that positive links are favored more in the diagonal blocks than in the off-diagonal blocks. During the experiments however, the MMSB model performed significantly worse than the other algorithms since the model is not suited for sparse graphs as it expends significant effort in modeling negative links between nodes. Airoldi et al. [7] propose variations of the MMSB model that take into account sparsity but an implementation for those variations with which to run experiments was unavailable. We therefore, do not report results with MMSB.

4 Datasets

We investigate the clustering properties of the two classes of approaches on three types of datasets. The nodes in the graphs in all the datasets studied have labels, which are used only to evaluate the accuracy of clustering.

The first type of datasets consists of social networks, citation networks, and similar networks that have been studied in the sociology literature. The nodes in the PolBook dataset[8] are political

Table 1: Dataset Statistics (N/E/C indicates Nodes / Edges / Clusters)

(a) Social network

Dataset	N/E/C	Dataset	N/E/C
karate	34 / 156 / 2	umbc	404 / 4764 / 2
polbooks	105 / 882 / 3	mgemail	280 / 1344 / 55
dolphin	62 / 318 / 2	citeseer	2114 / 7396 / 6
football	115 / 1226 / 10	cora	2485 / 10138 / 7
msp	4324 / 37254 / 2		
ag	1222 / 33428 / 2		
senate	98 / 9506 / 2		

(b) Author disambiguation

Dataset	N/E/C	Dataset	N/E/C
jsmith	4120 / 21452 / 30	jrobinson	686 / 2846 / 12
akumar	801 / 2476 / 14	ktanaka	827 / 2758 / 10
cchen	424 / 1558 / 16	mbrown	579 / 2112 / 13
djohnson	1381 / 5344 / 15	mmiller	2106 / 9918 / 12
jmartin	424 / 1558 / 16	jlee	5820 / 23110 / 100
agupta	2485 /10208 / 26	ychen	5472 / 25584 / 71
mjones	961 / 3450 / 13	slee	5963 / 23086 / 86

books, and edges represent co-purchasing behavior. Books are labeled "liberal", "conservative", or "neutral", based on their viewpoint. The nodes in the Karate dataset[9] are members of a karate club, and the edges are friendships. The labels are sub-communities, as defined by two subgroups that formed after a breakup of the original community. The Dolphin dataset[10] is a similar social network of associations between dolphins in a pod in Doubtful Sound, New Zealand, and labels correspond to sub-community membership after a similar breakup. The nodes in the Football dataset are Division IA colleges, the edges represent games in the 2000 regular season, and the labels represent conferences[11]. The nodes in the MGEmail corpus[12] are MBA students, organized in teams of four to six members, who ran simulated companies over a 14-week period as part of a management course at Carnegie Mellon University. The edges correspond to emails, and the true cluster labels correspond to teams. In the UMBC [13], AG [14], and MSP [13] datasets, the nodes are blogs, and an edge between two nodes represents hyperlinks. Blog sites are labeled either liberal or conservative. The MSP dataset also contains news cites, which are unlabeled. In the Cora and CiteSeer datasets, nodes are scientific papers, and links are citations. Node labels scientific subfield. The nodes in the Senate dataset are US Senators, and edges are agreement on congressional votes. The labels correspond to political party. Unlike other datasets, this is a complete graph.

The second type of datasets were used for author disambiguation [15]. Each of the dataset corresponds to a first name initial and a common last name. The datasets were constructed by extracting co-authorship information for papers authored by people with these ambiguous first initial-last name pairs. In each dataset, there are two types of nodes: (a) one node for each distinct name string, and (b)one node for each occurrence of a name in the list of authors of a paper. Edges link a name occurrence with the corresponding name string, and also link the name occurrence nodes for co-authors of a paper. The label of name occurrence nodes correspond to the id of the person associated with this name occurrence, and name string nodes are unlabeled.

The third type of datasets are synthetic. We generated datasets similarly to the planted partition model [16], as follows. (1) The size of the clusters are drawn from a Gaussian distribution with a mean at n=k. (2) Within-cluster edges are generated according to the Erdos-Renyi random graph model in one variant (ER) and the Barabasi-Albert scale-free network model (BA), respectively in the second. We also link all the nodes in the ER in a chain to avoid trivial, uninteresting clusters. (3) Inter-cluster edges are drawn at random according to a noise parameter, which defines the ratio of the probability of a within-cluster edges and to the probability of an inter-cluster edges. For each of E-R and B-A cluster models, variant models, we first set the noise parameter to 0.05 and vary the number of clusters; and then we fix the cluster size at 3 and vary the noise parameter. With clusters sizes of of 2,3,5,8 and 13 and noise values of 1%, 5%, 10%, 20%, and 30%, we obtain a total of 20 synthetic datasets.

5 Results and analysis

Clustering using the sparse model presented can be performed using two methods. In the first method, the number of clusters K is set to the number of known clusters in the dataset. After inference, each node e is assigned to a cluster as determined by $\arg\max_z \beta_z^e$. The clusters are then aligned with known class labels such that the alignment provides the best accuracy in predicting the cluster label (the optimal alignment can be efficiently determined using the Hungarian algorithm). In the second method, each node is associated with a distribution over clusters by normalizing β_z^e and the 1-NN algorithm is used to assign labels to nodes by using the Jensen-Shannon distance between cluster distributions as the metric to measure the distance between two nodes.

Table 2: Node clustering quality

(a) NMI: Social networks

Dataset	PSK	PIC_D	PIC_R	PIC_{R4}	NCut	NJW
Karate	**0.98**	0.65	0.75	0.76	0.76	0.78
Dolphin	0.61	**0.89**	0.86	**0.89**	**0.89**	**0.89**
UMBC	0.72	0.69	0.74	0.75	0.75	**0.76**
AG	**0.73**	0.67	0.70	0.72	0.02	0.00
MSP	**0.51**	0.00	0.00	0.00	0.00	0.02
Senate	0.88	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**
PolBook	0.54	0.46	0.55	**0.59**	0.57	0.56
Football	**0.83**	0.53	0.57	0.72	0.80	0.77
MGEmail	0.62	0.69	0.70	**0.83**	0.83	0.82
CiteSeer	0.12	0.28	0.24	0.31	0.25	**0.32**
Cora	0.29	**0.38**	0.30	0.36	0.04	0.30
Average	**0.62**	0.56	0.58	0.62	0.53	0.56

(b) NMI: Author disambiguation

Dataset	PSK	PIC_D	PIC_R	PIC_{R4}	NCut	NJW
AGupta	0.13	0.35	0.33	**0.49**	0.21	0.48
AKumar	0.18	0.32	0.31	0.38	0.37	**0.41**
CChen	0.39	0.54	0.57	**0.68**	0.30	0.63
DJohnson	0.11	0.27	0.32	0.41	0.28	**0.44**
JLee	0.26	0.45	0.50	**0.69**	0.35	0.67
JMartin	0.36	0.53	0.56	**0.67**	0.30	0.63
JRobinson	0.24	0.39	0.44	**0.55**	0.25	0.55
JSmith	0.12	0.34	0.30	0.48	0.30	**0.52**
KTanaka	0.10	0.34	0.36	**0.40**	0.36	0.34
MBrown	0.21	0.41	0.46	0.56	0.53	**0.56**
MJones	0.12	0.27	0.31	0.40	**0.44**	0.38
MMiller	0.07	0.30	0.29	0.40	0.25	**0.46**
SLee	0.20	0.42	0.48	0.65	0.38	**0.65**
YChen	0.20	0.47	0.54	**0.71**	0.35	0.68
Average	0.19	0.39	0.41	**0.53**	0.33	0.53

(c) Best alignment: Social networks

Dataset	PSK	PIC_D	PIC_R	PIC_{R4}	NCut	NJW
Karate	**1.00**	0.91	0.93	0.95	0.95	0.95
Dolphin	0.90	**0.98**	0.98	**0.98**	**0.98**	**0.98**
UMBC	0.95	0.93	0.95	0.95	0.95	**0.96**
AG	**0.95**	0.91	0.94	0.94	0.52	0.51
MSP	**0.88**	0.63	0.63	0.63	0.63	0.64
Senate	0.98	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
PolBook	0.78	0.80	0.81	**0.83**	0.82	0.80
Football	**0.76**	0.47	0.51	0.66	0.72	0.67
MGEmail	0.28	0.39	0.40	**0.64**	0.59	0.56
CiteSeer	0.33	0.51	0.48	**0.55**	0.48	0.52
Cora	**0.47**	0.46	0.40	0.45	0.29	0.42
Average	0.75	0.73	0.73	**0.78**	0.72	0.73

(d) Best alignment: Author disambiguation

Dataset	PSK	PIC_D	PIC_R	PIC_{R4}	NCut	NJW
AGupta	0.13	0.26	0.24	**0.37**	0.26	0.34
AKumar	0.20	0.29	0.31	0.37	0.35	**0.40**
CChen	0.30	0.43	0.44	**0.53**	0.24	0.50
DJohnson	0.15	0.24	0.33	0.46	**0.47**	0.35
JLee	0.11	0.20	0.23	**0.41**	0.17	0.39
JMartin	0.28	0.42	0.43	**0.53**	0.25	0.49
JRobinson	0.26	0.37	0.42	**0.49**	0.26	0.48
JSmith	0.11	0.22	0.21	0.41	0.31	**0.42**
KTanaka	0.19	0.36	0.41	0.45	**0.45**	0.43
MBrown	0.21	0.35	0.41	**0.52**	0.47	0.50
MJones	0.19	0.29	0.34	0.38	**0.38**	0.35
MMiller	0.14	0.30	0.41	0.52	0.52	**0.53**
SLee	0.08	0.19	0.23	**0.41**	0.23	0.39
YChen	0.10	0.23	0.28	**0.47**	0.23	0.46
Average	0.18	0.30	0.34	**0.45**	0.33	0.43

(e) 1-NN: Social networks

Dataset	PSK	PIC_D	PIC_R	PIC_{R4}	NCut	NJW
Karate	**1.00**	**1.00**	0.99	0.99	1.00	0.97
Dolphin	0.89	0.95	0.95	0.95	0.95	**0.98**
UMBC	0.92	0.93	0.93	0.93	0.92	**0.94**
AG	0.92	**0.94**	0.93	0.93	0.88	0.89
MSP	0.84	0.76	0.73	**0.86**	0.64	0.59
Senate	0.97	1.00	**1.00**	**1.00**	**1.00**	**1.00**
PolBook	0.79	0.68	0.76	0.80	**0.84**	0.78
Football	0.89	0.43	0.45	0.85	0.94	**0.95**
MGEmail	0.22	0.27	0.26	0.72	0.80	**0.81**
CiteSeer	0.34	0.55	0.54	**0.71**	0.69	0.66
Cora	0.45	0.56	0.51	**0.80**	0.47	0.75
Average	0.75	0.73	0.73	**0.87**	0.83	0.85

(f) 1-NN: Author disambiguation

Dataset	PSK	PIC_D	PIC_R	PIC_{R4}	NCut	NJW
AGupta	0.68	0.74	0.72	**0.95**	0.79	0.91
AKumar	0.82	0.69	0.74	**0.85**	0.79	0.81
CChen	0.77	0.73	0.74	**0.89**	0.75	0.85
DJohnson	0.81	0.81	0.83	**0.95**	0.85	0.92
JLee	0.55	0.61	0.68	**0.92**	0.79	0.91
JMartin	0.77	0.73	0.73	**0.88**	0.75	0.85
JRobinson	0.86	0.75	0.80	**0.92**	0.83	0.85
JSmith	0.65	0.75	0.67	**0.93**	0.85	0.91
KTanaka	0.81	0.84	0.86	**0.95**	0.90	0.90
MBrown	0.83	0.78	0.82	**0.93**	0.86	0.89
MJones	0.79	0.69	0.71	**0.91**	0.90	0.89
MMiller	0.81	0.83	0.81	**0.99**	0.97	0.98
SLee	0.59	0.69	0.77	**0.92**	0.85	0.92
YChen	0.57	0.73	0.79	**0.95**	0.84	0.94
Average	0.74	0.74	0.76	**0.92**	0.84	0.90

The quality of clustering can also be evaluated by measuring the normalized mutual information(NMI) between the most likely clusters for each node and the known cluster labels. If $\hat{Z}$ is the random variable denoting the cluster assignments and Z, the random variable denoting the true class labels, then NMI is defined as $\frac{I(\hat{Z};Z)}{(H(\hat{Z})+H(Z))/2}$ where $I(\hat{Z};Z)$ is the mutual information be-

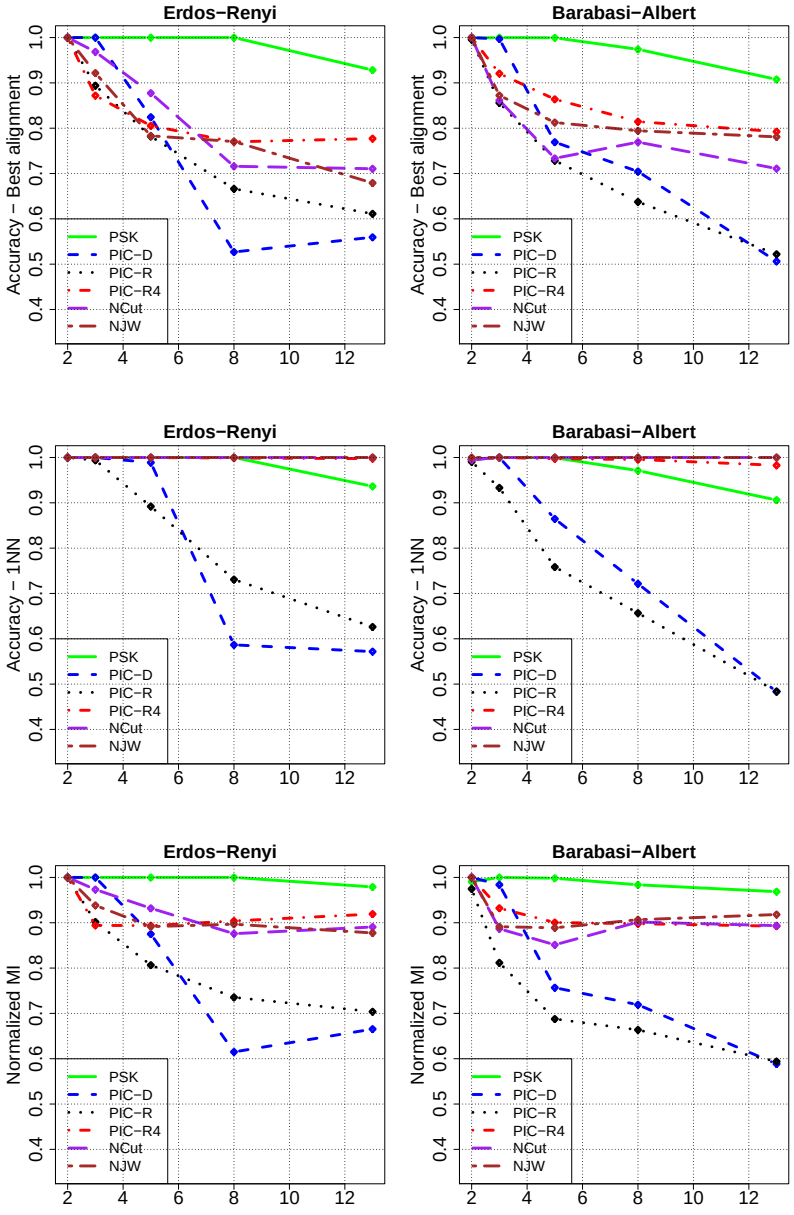

Figure 3: Varying number of clusters in synthetic datasets

tween $\hat{Z}$ and Z and H denotes entropy. NMI values range from 0 to 1 with higher values indicating better clustering.

A similar approach is used with the spectral techniques after replacing Jensen-Shannon divergence with Euclidean distance. In all our experiments, we set the number of partitions or clusters to be the number of known clusters in the dataset.

Table 2 shows the clustering performance by the different algorithms presented on the social network and author disambiguation datasets. The performance is measured using 1-NN accuracy, best alignment accuracy and using normalized mutual information as described above. Bold entries in each row highlight the score of the best algorithm for the given metric and dataset. For all the data sets except the Senate dataset, we ignore edge weights during probabilistic modeling. Since the senate vote dataset has an edge between every pair of nodes, we eliminate those edges with weights below a threshold and use the remaining edges in an unweighted fashion.

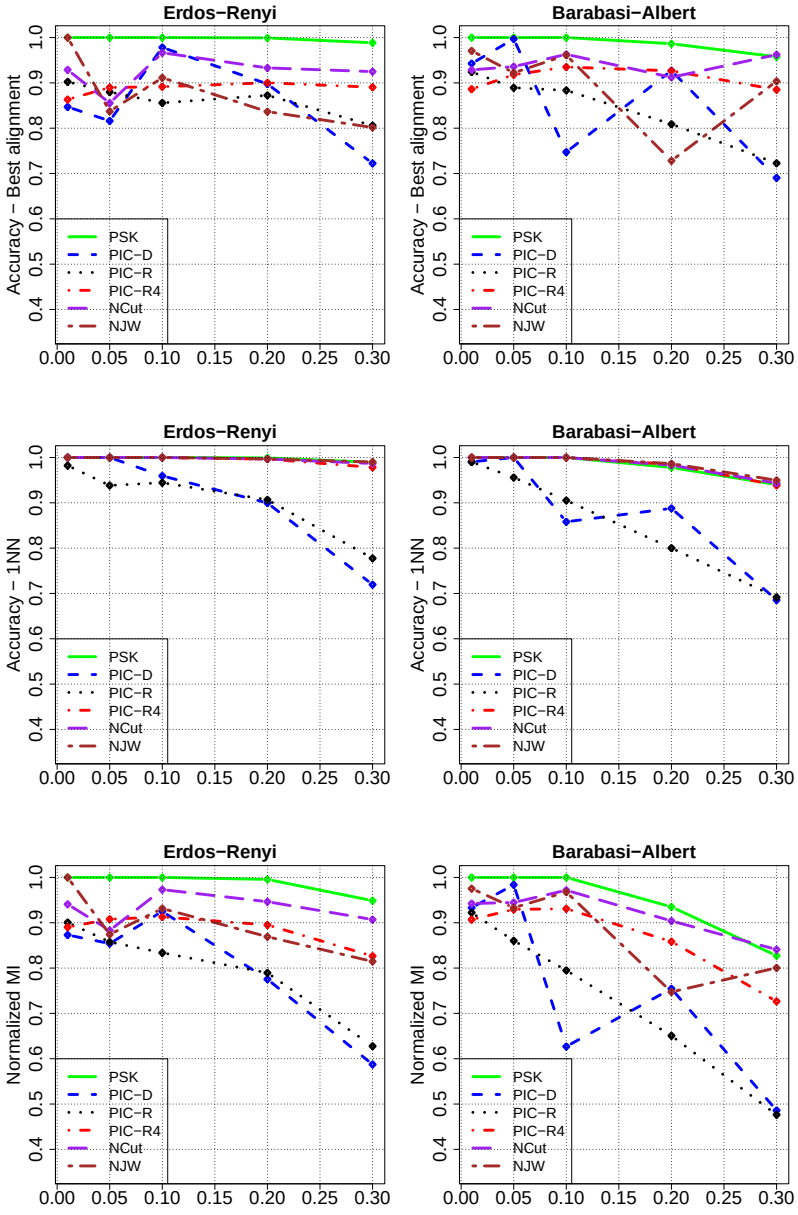

Figure 4: Varying noise in synthetic datasets

Table 2 shows that the spectral methods perform better consistently for the author disambiguation tasks. These datasets on average have a larger number of nodes, edges and classes than the social network datasets. The probabilistic model considered has parameters that scale quadratically with the number of clusters which tends to lead to over-fitting problems and slower convergence with approximate inference techniques. For the social network datasets however, especially with datasets with fewer classes, the probabilistic techniques are competitive and often work better than the spectral techniques, especially when the network intrinsically has many edges between nodes from different clusters. This difference is studied further in the experiments with synthetic datasets. Within the spectral techniques, the most competitive algorithms are PIC with 4 random initial vectors and the NJW technique.

A noteworthy result to point out is PSK's strong performance compared to spectral methods on the MSP dataset—all spectral methods failed completely ($\sim$0 NMI) to reproduce clusters resembling

class labels on this dataset. This network dataset is peculiar in that it is strongly bipartite—it consists of blog sites that points to news articles but rarely to another blog. This results in an off-diagonal block structure in the graph matrix, which may not be what spectral methods expect (spectral clustering can be explain via matrix perturbation of block-diagonal matrices). While a bipartite graph can be "folded" into a unipartite graph [17] if such underlying structures are known *a priori*, this result suggests that since PSK make less assumptions about the graph structure, it may be more robust to various types of graph data compared to spectral methods.

Figures 3 and 4 show the performance of clustering using the same three metrics on the synthetic datasets. The first six subplots study the effect of varying the noise parameter in the datasets while the last six plots study the effect of cluster size. As noted, it can seen that PIC methods drop accuracy at a faster rate than the probabilistic method when the noise parameter is increased. This behavior is less pronounced with PIC_{R4} than with PIC_{R} and PIC_{D}. A similar behavior is observed when the number of clusters is increased. PIC_{R4} exhibits a small drop but with PIC_{R} and PIC_{D}, the accuracies and NMI drop significantly. With all the synthetic datasets, the sparse network model exhibits better performance than the spectral techniques due to the low number of classes in the datasets.

We can therefore see that in cases where there is a higher incidence of inter-cluster edges(noise), probabilistic techniques with the right priors are more suitable than spectral techniques. However as the number of nodes and edges increase, the increase in the number of parameters to fit leads to over-fitting making the spectral techniques more attractive.

6 Conclusion

We presented an empirical study of node clustering algorithms from the spectral and probabilistic model families. Our experiments show that the neither class completely dominates the other and that spectral techniques, the best of which was PIC_{R4}, works better for larger graphs and that the sparse network model works well with smaller number of clusters and is not sensitive to inter cluster noise.

References

[1] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models: Blockmodels, stochastic and discovery, community and models, latent space. 2(2):129–233, 2010.

[2] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional Stochastic Block Model. July 2010.

[3] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[4] Andrew Y. Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.

[5] Frank Lin and William W. Cohen. Power iteration clustering. In *ICML*, 2010.

[6] Juuso Parkkinen, Janne Sinkkonen, Adam Gyenge, and Samuel Kaski. A block model suitable for sparse graphs. In *The 7th International Workshop on Mining and Learning with Graphs*, Leuven, 2009. Poster.

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

[8] Books about us politics.

[9] W.W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[10] David Lusseau, Karsten Schneider, Oliver J. Boisseau, Patti Haase, Elisabeth Slooten, and Steve M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[11] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, June 2002.

[12] R. Kraut, S.R. Fussell, F.J. Lerch, and A. Espinosa. A. coordination in teams: Evidence from a simulated management game. *Under Review*.

[13] Frank Lin and William W. Cohen. Semi-supervised classification of network data using very few labels. In *ASONAM*, 2010.

[14] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 U.S. election: Divided they blog. In *LinkKDD*, 2005.

[15] *Name Disambiguation in Author Citations Using a K-way Spectral Clustering Method*. ICDL, 2005.

[16] Anne Condon and Richard M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

[17] Frank Lin and William W. Cohen. A very fast method for clustering big text datasets. In *ECAI*, 2010.