Spring 2016

Variance of Clusterings on Graphs

Thomas Vlado Mulc
Rose-Hulman Institute of Technology, mulctv@rose-hulman.edu

Advisors:
John K. McSweeney

Variance of clusterings on graphs

by

Thomas Mulc

May 20, 2016

In partial fulfillment of the requirements of the course

Senior Thesis
Advisor: Professor John McSweeney, Ph.D.

DEPARTMENT OF MATHEMATICS ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

1 Introduction

Graphs that represent data often have structures or characteristics that can represent some relationships in the data. One of these structures is *clusters* or *community structures*. Most clustering algorithms for graphs are deterministic, which means they will output the same clustering each time. We investigated a few stochastic algorithms, and look into the consistency of their clusterings. Lastly, we address issues with using the true labeling of a graph as metric for clustering. For our test data, we use the Karate graph [14], an artificial dumbbell graph, and the games played in 2014 for NCAA Division 1 Football [9]. We used R for all the computations, and inside R we used the iGraph package for data visualization and comparisons of standard algorithms. The football data was cleaned using python.

2 Test Graphs

The Karate graph came about after a karate club had two members, each of which had different views on the prices of lessons. This led to fission in the club, and members would politically associate themselves with one of the leaders so that during club votes, it was as if individuals were voting on behave of their leader/party. A group studied the club members and took note when members we seen together outside of karate; when this occurred the two persons had a friendship. A graph was created to represent the social dynamics of the club. The nodes represent people, and the edges represent a friendship. There is clear community structure because those who were close to either leader tended to have similar ideological view and wouldn't socialize with people of opposing views.

The Dumbbell graph was generated in R using the iGraph package. First, two Erdős-Rényi random graphs were generate with parameters ($p = .99, |N| = 10$). Then, the graphs were connected by adding one edge from the first random graph to the second; the result was our Dumbbell graph. This graph was generated to have two communities of each of equal size with a high density of edges within communities and low density outside of communities.

The Football graph is generate by letting nodes represent teams, and letting edges represent games played. There is an inherent "true labeling" of

the vertices, by letting each vertex be labeled by its respective conference. A clustering algorithm may output a set of labels that disagrees with the conference labels. The true labeling might represent what the league wanted as far as who plays whom during the season, but a clustering algorithm might unveil the reality of the scheduling. Determining which is better–the intended labels or an algorithms output–is subjective, but it's important to recognize that an algorithm may be uncovering a more logical partition. On the other hand, it's equally as important to be wary of the output of an algorithm, because the algorithm may have a flaw or may have made assumptions that are unrealistic in the scenario.

2.1 Definitions

An undirected graph $G = (V, E)$ has a set of nodes or vertices V and a set of edges $E \subseteq V \times V$. The size of the graph can be described used the number of nodes or number of edges

$$\mid V \mid = n \text{ and } \mid E \mid = m.$$

An adjacency matrix A of G is an $n \times n$ matrix where each element is defined by

$$A_{ij} = \begin{cases} 1 & e_{ij} \in E \\ 0 & e_{ij} \notin E \end{cases}$$

where e_{ij} is an edge from node i to node j.
The degree of vertex $i \in V$ is

$$deg(i) = \sum_{j \in V} A_{ij}.$$

A partition $\mathcal{P}$ of the vertices of G is defined by

$$\mathcal{P} = \{C_1, C_2, ..., C_k\},$$

where

$$C_1 \cup C_2 \cup ... \cup C_k = V \text{ and } C_i \cap C_j = \emptyset \quad \forall i \neq j.$$

3 Clustering

Clustering is inherently a subjective way to organize data. Mathematically, there is a general agreed upon definition of a clustering, but no agreed upon definition of what constitutes a "good" clustering. In general, a good clustering has **many edges within each cluster** and **few edges between clusters**. In the context of graphs, we define a clustering θ on a graph a a set of groupings of the nodes. Thus, a clustering on θ on a graph G is a partition of V.

3.1 Quantifying clustering

Metrics exist for quantifying the goodness of a clustering.

3.1.1 Modularity

Perhaps the most agreed upon metric for clusters on graphs is Modularity. The modularity score, Q can be thought of as the fraction of edges that fall within either group minus the expected number of edges from and Erdős-Rényi random graph. It is calculated by

$$Q = \frac{1}{2m} \sum_{ij} \Big[A_{ij} - \frac{deg(i)deg(j)}{2m} \Big] I(i,j),$$

where

$$I(i,j) = \begin{cases} 1 & i \text{ and } j \text{ in same cluster} \\ 0 & \text{else.} \end{cases}$$

The modularity can be negative, and $Q \in [-\frac{1}{2}, 1]$. Typically, modularity scores of 0.3 and higher imply considerable community structure [6].

4 Clustering Algorithms

4.1 SC Method

The general idea of the SC algorithm [**?**] is generate clusterings by taking random walks at the nodes, and examining the walks to determine the clustering. The SC algorithm can be broken up into two algorithms that operate

in serial: Algorithm 1 and Algorithm 2. In Algorithm 1, the random walks are taken and form a similarity matrix, and in Algorithm 2 the similarity matrix is used for agglomerative clustering. When attempting to replicate [**?**] we found some errors in the stopping criteria, so the algorithms here are our best interpretation of how the SC Method is supposed to behave algorithmically.

4.1.1 Algorithm 1

Each node i: for $a \leq t$:

1. Take random step ($Pr_{ij} = \frac{A_{ij}}{d(i)}$.)
2. Create similarity matrix S using number of visits made on walks

4.1.2 Algorithm 2

1. Start with each node in its own cluster
2. Each iteration, find largest entry in S and merge nodes into same cluster
3. Continue until all nodes are in the same cluster or other criteria met

4.2 SC-Mod

We implemented a different form of SC, SC-Mod, that used modularity as its criterion for the final clustering. SC-Mod uses the same random walk algorithm as SC, but instead of outputting the clustering left when no more merges are possible or when the number of specified clusters had been formed, SC-Mod outputs whatever cluster has the highest modularity from all the clusterings formed during the merging algorithm.

4.3 Other stochastic methods

Most other clustering algorithms are deterministic, but we found *Spinglass* and *Label Propagation* are two other stochastic algorithms available in the iGraph package [12] [11]. We won't go into the details of these algorithms, but it's important to know that their results are random. We used these two methods to compare consistency of random clustering algorithms.

4.4 Walktrap

There are many deterministic method for graph clustering. But the one that is most relatable to SC is Walktrap. Walktrap [2] was inspired by the idea of random walks, but unlike SC, it does not actually take random walks, and is a deterministic method.

Walktrap runs in time $\mathcal{O}(mn^2)$ and space $\mathcal{O}(n^2)$, but can normally run in time $\mathcal{O}(n^2 \log n)$ and space $\mathcal{O}(n^2)$.
The general idea of this approach is to take a random walk from a node to determine the state after some number of steps.

A transition matrix P is created where $P_{i,j}$ is the probability of going from node i to node j:

$$P_{ij} = \frac{A_{ij}}{d(i)}.$$

A distance function r is defined by

$$r_{ij} = \sqrt{\sum_{k=1}^{n} (P_{ik} - P_{jk})^2 \frac{1}{d(k)}}.$$

The actual clustering algorithm is agglomerative. The initial partition is $\mathcal{P}_1 = \{\{v\}, v \in V\}$ so that each vertex is its own cluster. At each step k:

- Find $C_1, C_2 \in P_k$ s.t. they meet a minimum distance criterion
- Set $C_3 = C_1 \bigcup C_2$ and $\mathcal{P}_{k+1} = (P_k \setminus \{C_1, C_2\}) \bigcup \{C_3\}$
- update distances between adjacent communities.

After $n-1$ steps, $\mathcal{P}_n = \{V\}$. The final clustering of G is decided by picking one partition P_k that meets some criterion.

5 Consistency of Clusterings

Since the SC algorithm is stochastic by nature, we were interested in finding the consistency of the clusterings. We want to describe the spread of the clusterings and want a partition equivalent to a variance. Describing the variance for a set of clusterings is not straight forward. To do this, we first quantified how much two clusterings agree.

6 Difference between two clusterings

A clustering of G is a partition on V. We describe the difference between two clusterings by finding the difference between two partitions. We use a function called the Rand Index to quantify this difference [10].

6.1 Rand Index

The Rand Index is a defined by

$$d(P_i, P_j) = \frac{a+b}{\binom{n}{2}}$$

where

- $a \equiv$ number of pairs that were in same cluster in P_i and were not in the same cluster in P_j
- $b \equiv$ number of pairs that were not in the same cluster in P_i and were in the same cluster in P_j

We have

$$0 \leq \frac{a+b}{\binom{n}{2}} \leq 1$$

so that a value of 0 means that two clusterings are identical and a value of 1 means that every possible pair of nodes disagrees. The Rand Index can be thought of as the fraction node-pairs that disagree. This notion is nice because it allows us to make comparisons of the same clustering algorithm on different graphs. Additionally, the Rand Index works well for our application because it allows us to compare clusterings that don't have the same number of clusters. Finally, the Rand Index is mathematically nice because it's a metric on P_k.

Let $v_1, v_2 \in V$. Let v_1 and v_2 agree if $\{v_1, v_2\} \subset C_a, C_b$ for some $C_a \in P_i$ and $C_b \in P_j$. Let v_1 and v_2 disagree if they don't agree.

Let $\gamma : V \times V \times \{1, ..., |P_k|\} \times \{1, ..., |P_k|\}$ be a function:

$$\gamma(v_1, v_2)_{i,j} = \begin{cases} 1 & v_1 \text{and} v_2 \text{ agree} \\ 0 & else \end{cases}$$

Proof. Let $P_h, P_i, P_j \in P_k$.
Separation:
$d(P_i, P_j) = \frac{a+b}{\binom{n}{2}}$ where $a, b, \binom{n}{2} \geq 0$. Thus $d(P_i, P_j) \geq 0$.

Symmetry:
$d(P_i, P_j) = \frac{a+b}{\binom{n}{2}}$ and $d(P_j, P_i) = \frac{a^*+b^*}{\binom{n}{2}}$. Note that a is the number of nodes that agree in P_i and disagree in P_j, while b^* is the number of nodes the disagree in P_j and agree in P_i. Thus $a = b^*$. Similarly for b and a^* we have $b = a^*$. Hence, $d(P_i, P_j) = \frac{b^*+a^*}{\binom{n}{2}} = d(P_j, P_i)$.

Identity of Indiscernibles:
If $d(P_i, P_j) = 0$, then $\frac{a+b}{\binom{n}{2}} = 0$. Since $n > 0$ then $a + b = 0$. Since $a, b > 0$ then every pair of nodes agrees. Thus, $P_i = P_j$.

If $P_i = P_j$ then $a = b = 0$. Thus $d(P_i, P_j) = 0$.

Triangle Inequality:
Assume $d(P_i, P_h) > d(P_i, P_j) + d(P_j, P_h)$. Then for $a, b \in V$,

$$\sum_{a<b}(a,b)_{i,h} > \sum_{a<b}(a,b)_{i,j} + \sum_{a<b}(a,b)_{j,h}.$$

Thus, $\exists(o,p) \in (V \times V)$ such that $(o,p)_{i,h} = 1$ and $(o,p)_{i,j} = (o,p)_{j,h} = 0$.

Case 1: (o,p) different in i and same in h.
Then for i, j, o, p are either

1. same in i and same in $j \Rightarrow$ Contradiction because can't be same in i
2. different i and different j.

Then for j, h, o, p are either

1. same j and same h
2. different j and different h. $\Rightarrow$ Contradiction because can't be same in h

Thus, it must be that o, p are different i and different j and same j and same h . This is a contradiction.

Case 2: (o, p) same in i and different in h.
Then for i, j, o, p are either

1. same in i and same in j
2. different i and different j. $\Rightarrow$ Contradiction because must be same in i

Then for j, h, o, p are either

1. same j and same $h \Rightarrow$ Contradiction because must be different in h
2. different j and different h. Thus, it must be that o, p are same in i and same in j and different j and different h. This is a contradiction.

Since all cases resulted in a contradiction, our assumption is false

$$\therefore d(P_i, P_h) \leq d(P_i, P_j) + d(P_j, P_h)$$

□

6.2 Variance of clusterings

We calculate the variance similar to how we would calculate the variance of a set of numbers. For a sample of numerical data, $X = \{x_i : i \in \{1, ..., N\}\}$, the sample variance is

$$\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2$$

where $\bar{x}$ is the sample mean. This form can't be used to compute the variance of partitions, since there's no notion of a mean. The sample variance can be written pairwise as

$$\frac{1}{N^2} \sum_{i<j} (x_i - x_j)^2.$$

This form is convenient because it doesn't involve a mean. The term $(x_i - x_j)^2$ is the distance between two elements from X. We can use this notion to

quantify the spread of a set of partitions, $\theta = \{P_1, P_2, ..., P_N\}$. The variance of θ would be

$$\sigma_\theta^2 = \frac{1}{N^2} \sum_{i<j} d(P_i, P_j).$$

Note that σ_θ^2 is some what of an average number of pairs that disagree across all clusterings. The actual average number of pairs that disagrees is

$$\frac{1}{\binom{N}{2}} \sum_{i<j} d(P_i, P_j).$$

6.2.1 Run Time

Let $\theta = \{P_1, P_2, ..., P_N\}$, where P_i is a partition of V. To calculate the variance of θ, the Rand Index must be calculated for every pair of partitions. Calculating the Rand Index for a single partion pair takes $\mathcal{O}(\binom{n}{2})$. Thus, the total run time is

$$\mathcal{O}\Big(\binom{N}{2}\binom{n}{2}\Big) = \mathcal{O}\big((Nn)^2\big)$$

7 Methods

To test the variance formula, we simulated SC, SC-Mod, Spinglass, and Label Propagation on the following graphs: Karate, dumbbell, and 2014 NCAA Football. Since the Karate and dumbbell graphs were small (fewer than 50 nodes), we ran 100 simulations of each algorithm ($|\theta| = 100$). The Football graph was too large for the same sample size, so we only ran 30 simulation of each algorithm.

8 Results

The SC algorithm was run on 3 graphs: Karate, Dumbbell, and 2014 NCAA Football. The Dumbell graph, shown in Figure 2 was an artificial Erdős-Rényi random graph with 20 vertices.

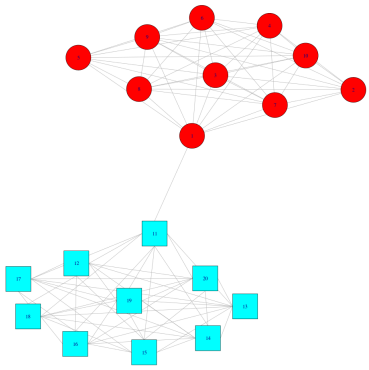
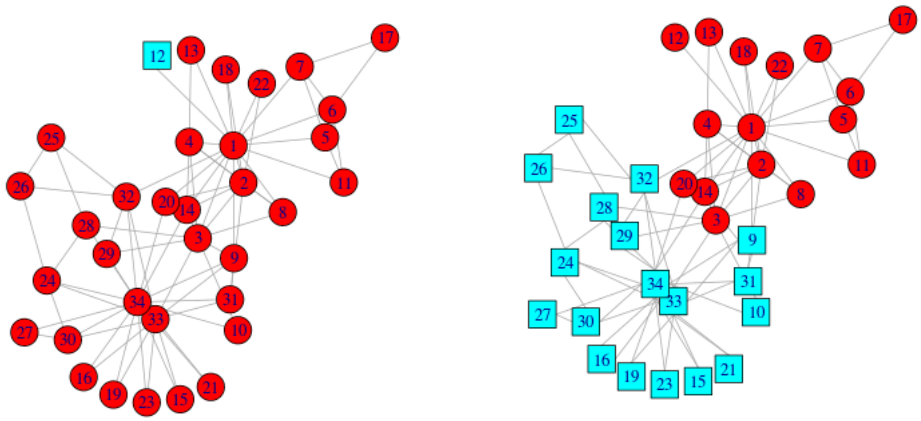


Figure 1: Artifical dumbell graph with clear community structure.

Initially, we were getting incredibly different clusters for each implementation of SC on the Dumbbell graph. To give the algorithm the best possible chance to return the two communities that most algorithms agree on, we then implemented the criterion for SC to return the clustering where $|P| = 2$. Figure 3 shows two clusterings generated using the SC algorithm with these constraints on the Karate graph.



(a) One cluster only has one node

(b) Clusters with appropriate number of nodes

Figure 2: Forcing a clustering to stop at 2 clusters doesn't guarantee a good clustering

The clustering on the left is trivial, and resulted from the SC merging algorithm. To remove these clusterings from our results, we created a threshold, α, such that if

$$\min\{|C| : \forall C \in P\} \leq \alpha$$

then P was thrown out and a new clustering was generated. This process was repeated until

$$\min\{|C| : \forall C \in P_i\} > \alpha, \quad \forall P \in \theta.$$

Figure 4 shows the relationship between variance and threshold on Karate.

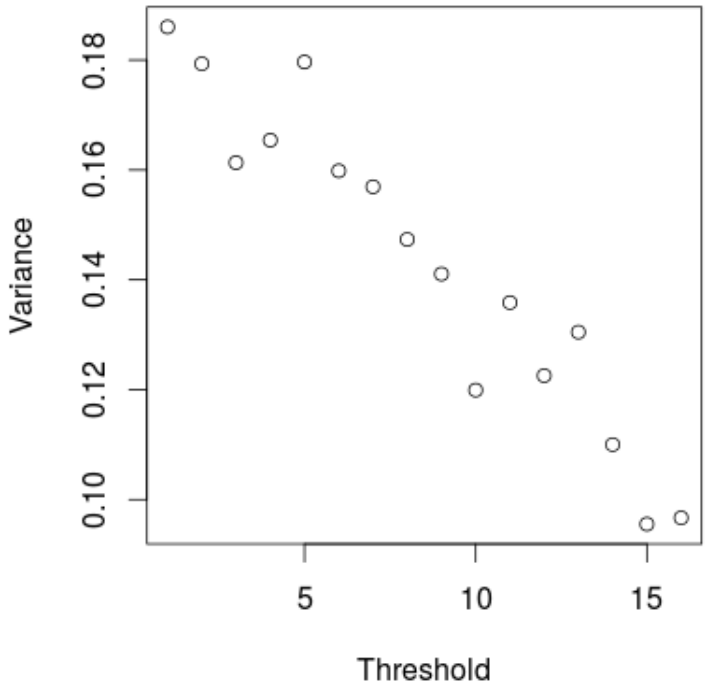


Figure 3: The relationship between the variance σ^2_θ and threshold α for SC on Karate

It's important to note the the maximum value for t is 16 since $|V(G)| = 34$ for the Karate graph. Thus, even when we forced the clusters to be of equal sizes, there was still an average 10% of possible pairs that didn't agree between two different partitions.

To account for these occurrences, clusterings that had one group with t or fewer nodes were not included. Also, we forced the algorithm to stop at 2 clusters. The variance for 1000 trials with $t = 3$ and walk size of 4 on Karate was 0.1738035; we can interpret this as the "average percentage of pairs of nodes that did not agree was 17.38%." As expected, the variance under the same condition for Walktrap is 0, since the clustering is deterministic.

	SC	SC-Mod	Spinglass	Label Propagation
Dumbbell	0.014	0.008	0.007	0.000
Karate	0.182	0.150	0.006	0.103
Football*	0	0.106	0.005	0.023

Table 1: Comparison of variances for stochastic clustering algorithms

Although SC had a variance of 0, that was because it gave a trivial cluster where every node was in the same cluster. The Football test were run with only 30 trials because the graph was much larger.

9 "True Labeling"

Steinhauser and Chawla recommended that modularity not be used as a metric for clustering, and instead, the true labeling of the vertices should be used as the benchmark. By true labeling we mean a labeling of the vertices, equivalently a partition of the vertices into clusters, provided a priori, which is taken to be the ground truth for the true community structure.

The true labeling of the vertices of graph sounds nice, because it defines an absolute truth to use in comparisons. One issue with using the true labeling of a graph when dealing with clustering is that not all graphs have a true labeling. Another issue is that even if the graph did have a true labeling, it may be flawed. For example, an interesting issue with true labels was discovered by Miller when doing an AMS topic on Nation Football League (NFL) center and math enthusiast, John Urschel [13]. Teams in the NFL are split among two conferences, the AFC and NFC, and with each conference, a team belongs to one of four divisions (denoted by the cardinal directions North, South, East, and West). A graph of 2015 NFL season can be generated by letting each team be represented by a node and letting edges represent games played between two teams; this is shown in Figure 1 taken from [13].

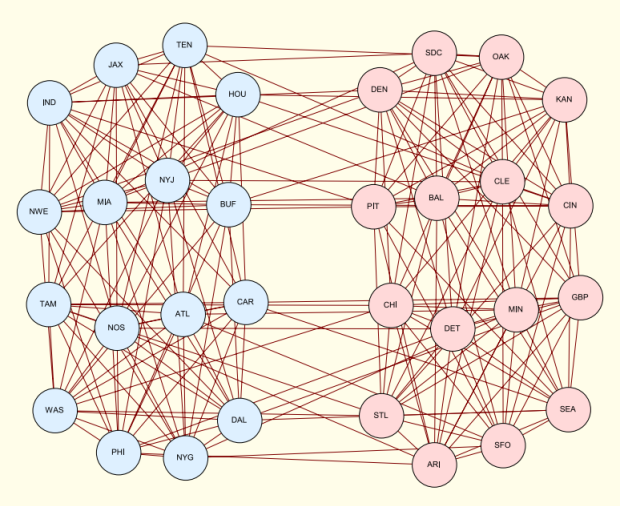


Figure 4: Spectral clustering on 2015 NFL graph

The teams on the upper-half of the graph belong to the AFC while the lower-half represents the NFC. In order to make sense of the Figure 1 its necessary to understand how the NFL scheduling.

In each season there are 192 intra-conference edges and 64 inter-conference edges. Each team will play:

- 2 games against each team in its division (12 intra-division edges per division)
- 1 game against each of the 4 teams from another other division within the same conference (call the two divisions an intra-conference pair)
- 1 game against each of the 4 teams one division that is inter-conference (the the two divisions an inter-conference pair)
- 1 game against one team from the 2 other intra-conference divisions.

It seems like since there are 96 intra-conference per conference, and only 64 inter-conference edges, naturally the best partition for splitting the NFL graph into 2 clusters would be to use each teams given conference.

However, the spectral clustering found by Miller assigned 50% of the teams to their incorrect respective conference. Miller went on to show that due

to the structure in the 2015 schedule, an intra-division pair in the AFC was assign an inter-division pair in the NFC. Thus, there was more community structure across conferences than within conferences according to The UrschelZikatanov Theorem that Miller used to generate the clusters. The UrschelZikatanov Theorem uses the eigen-value of the the Lapcian of the NFL graph to determine the clusters; this ultimately found the clustering on the NFL graph that split the teams into two clusters, where the number of edges between the clusters was minimum. In other words, the clustering shown in figure 1 was (objectively) determined by partitioning the vertices so that the number of edges between partitions was minimized.

Now, imagine an algorithm that measures performance based on true labels. This fictitious algorithm would perform poorly–under its own standards–on the 2015 NFL graph, rather than declaring the results anomalous! This example reveals the importance graph structure has in clustering. If only the graph structure is used, then the results of clustering tell you something about the graph. If, instead, the accuracy of labeling is used, interpreting the results is less clear. If an algorithm that used true label clustering criteria was created and used on the NFL graph for years other than 2015, the algorithm might appear to do well during training, but then in 2015, you have results that you are extremely confident in but in reality they are incorrect! This leads into the final concern with true labels. An algorithm that used true labeling as its basis for performance only has historical information as a means of credibility. This can become problematic if you'd like to test an algorithm on a graph where there has never been a true labeling assigned. For example, consider an algorithm such as SC that has performed extremely well on the standard graphs (i.e. Karate, NFL, etc.). Now, apply the algorithm to a graph that has nothing to do with Karate clubs. How confident are you that the clustering is correct, and why are you confident?

This issue is related to the divide of unsupervised and supervised learning in Machine Learning. An unsupervised algorithm takes data that does not contains labels and organized that data points so that each data point is assigned a label. In contrast, supervised learning takes data with labels (known as training data) and trains the algorithm to gives output for new data based on the previously seen training. In Machine Learning, clustering is categorized as unsupervised learning because it should work without training data. In general, clustering on a graph seems to follow the unsuper-

vised mold, because we want to give an output (i.e. clusters) only based on data (i.e. edges and nodes). The SC method uses unsupervised tactics and validates results using supervised tactics.

Most importantly, going back to the definition of a cluster, it doesn't make sense to validate an algorithm by using accuracy when the true labeling may have no relation to the structure of the graph.

10 Conclusions

Stochastic methods of clusterings on graphs exhibit variance. If consistency in the clustering is important, one should stick with deterministic methods.

Clustering falls into the category of unsupervised learning in the area of machine learning. As interest in machine learning and its applications continues to grow, the following areas related to this work warrant more research:

- Stochastic clustering methods
- Other metrics on P_k
- A different method for computing σ^2
- A fast method to estimate σ^2

Lastly, true labeling should not be used as a benchmark for clustering algorithms. (expand to paragraph)

References

[1] J. McAuley, R. Pandey, J. Leskovec, "Inferring networks of substitutable and complementary products," *Knowledge Discovery and Data Mining*, 2015

[2] P. Pons, M. Latapy, "Computing communities in large networks using random walks," *LIAFA - CNRS*, Dec. 2005.

[3] M. Clauset et al., "Finding community structure in very large networks," *Phys. Rev. E* 70, 6 Dec. 2004. doi: `10.1103/PhysRevE.70.066111`

[4] M. E. J. Newman, "Finding community structure in networks using eigenvectors of matrices," *Department of Physics and Center for the Study of Complex Systems*, Jun. 2006

[5] M. E. J. Newman, "Detecting community structure in networks," *Eur Phys. Rev. J. B.* 38,pp 321-330, Mar. 2004.

[6] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physics Review E*, vol. 69, 18 June 2004. DOI: 10.1103/PhysRevE.69.066133

[7] K. Steinhaeuser K, N. V. Chawla, "Is Modularity the Answer to Evaluating Community Structure in Networks?" 26 June 2008.

[8] K. Steinhaeuser K, N. V. Chawla, "Identifying and evaluating community structure in complex networks" Pattern Recognition Lett. 2009, DOI:10.1016/j.patrec.2009.11.001

[9] Sports Reference LLC. "2014 College Football," *Sports Reference*, Oct. 2016, http://www.sports-reference.com/cfb/years/2014-schedule.html

[10] W. M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association*, Vol. 66, No. 336, pp.846-850, Dec. 1971

[11] U.N. Raghavan, R. Albert, S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks" *Phys Rev E 76*, 2007, DOI:10.1103/PhysRevE.76.036106

[12] J. Reichardt, S. Bornholdt, "Statistical Mechanics of Community Detection," 3 February 2008, DOI:10.1103/PhysRevE.74.016110

[13] S.D. Miller, "'I plan to be a great mathematician': An NFL Offensive Lineman Shows Hes One of Us,"Notices of the AMS, DOI: http://dx.doi.org/10.1090/noti1331

[14] W.W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," *Journal of Anthropological Research*, Vol. 33, No. 4, pp. 452-473, 1977

[15] M. Matsumoto, T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," 1998.