Randomization Techniques for Statistical Significance Testing on Graphs

Sami Hanhijärvi SAMI.HANHIJARVI@TKK.FI
Helsinki University of Technology, Finland

Gemma C. Garriga GEMMA.GARRIGA@TKK.FI
Helsinki University of Technology, Finland

Kai Puolamäki KAI.PUOLAMAKI@TKK.FI
Helsinki University of Technology, Finland

Studying the patterns and properties of graph data is important in many application areas. A crucial question remains still largely ignored: how significant are the data mining results found on the graph data? Currently, the results are mostly justified by the optimal or near optimal value of the defined objective function. We study randomization techniques for testing the statistical significance of graph analysis results.

1. Preliminaries

We study unweighted undirected graphs G. We consider the following statistics to describe a part of the structure of a graph: *degree distribution*, *average clustering coefficient* and *characteristic path length*. The clustering coefficient of a node $v \in V(G)$ is the fraction of links the neighbors of the node have among them with respect to all possible such links. The characteristic path length of a graph is calculated as the mean of all pairs shortest paths between the nodes of a graph.

We base our analysis on *statistical hypothesis testing*. We denote a test statistic as $S = S(\mathcal{A}(G)) \in \mathbb{R}$, where $\mathcal{A}$ is the used data mining algorithm and $\mathcal{A}(G)$ is the result of the algorithm on graph G. The statistic could be defined, for example, as the value of the objective function: the value of the minimum cut in graph clustering. However in principle, it can be any function from the space of results to a real number. The null hypothesis H_0 is that for all graphs G that satisfy the given constraints, the values of $S(\mathcal{A}(G))$ follow the same null distribution Π_0. We find Π_0 by randomization, where the basic idea is to perturb the original data and carry out the experiments with the randomized version of the data. The randomized data can be thought to be sampled from a distribution, defined such that the chosen properties of the original data are maintained with a sufficient precision. When the randomization is performed several times, the experiments with the random data yield a set of values for the test statistic, which follow the null distribution Π_0. These are used to define an *empirical p-value*, which is the fraction of test statistic values that are more extreme than the test statistic value for the original data. The significance test entails a definition of a significance level α, which is the maximum p-value allowed to reject the null hypothesis. We use $\alpha = 0.05$.

2. Graph Randomization

All our randomizations preserve exactly the number of nodes and edges, as well as the degree distribution or individual node degrees, which are intuitive descriptors of a graph. We also study cases where the user may select additional graph statistics that are approximately preserved, such as the characteristic path length or the average clustering coefficient.

Let $\rho_0(G_s)$ be the distribution such that all graphs with a given number of nodes and edges and a certain degree distribution are equally likely. Our solution is to allow the user to define a distribution $\rho(G_s)$ from which the random samples will be drawn. In this paper, we will use a Gaussian distribution centered at the value of the preserved statistic of the original graph,

$$\rho(G_s) \propto N(R(G_s) - R(G), \sigma^2)\rho_0(G_s), \qquad (1)$$

where $N(\cdot, \cdot)$ denotes a Gaussian probability density function with a given mean and variance, and $R(G_s) \in \mathbb{R}$ and $R(G) \in \mathbb{R}$ describe the value of the graph statistic in the sampled and original graphs, respectively. We denote by Uniform the randomization that preserves only the degree distribution, $\rho(G_s) = \rho_0(G_s)$. The additional constraints that preserve the average clustering coefficient and the characteristic path length are denoted by AvgCC and CPL, respectively. In other words, $R(G_s)$ is defined as the average clustering coefficient of G_s for AvgCC, and as characteristic path length for CPL.

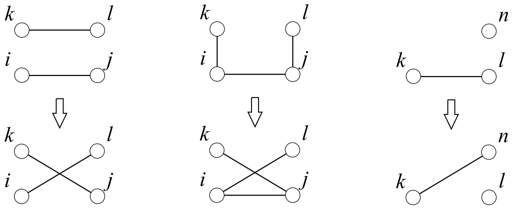
3. Markov Chain with Swaps

In this section, we describe three MCMC methods of backward-forward sampling (Besag & Clifford, 1989) for the randomization to obtain the samples from $\rho(G_s)$. This sampling method guarantees that the p-values we obtain are conservative, that is, even if the MCMC has not converged we should obtain p-values that are no less than the true p-values.

Let a swap be a distortion in a graph consisting of the exchange of an edge between two nodes. We propose to use Markov chains to construct the samples of randomized graphs by means of swaps. The idea of our solution is that we start the Markov chain from the original graph, and make small random distortions that will affect at most two edges. These swapping distortions are designed to preserve the degree distribution of the original graph at all times. By means of applying these distortions as long as is needed for the chain to mix, we will arrive to a randomized graph, different from the original one.

Since the swaps are small changes, we call two graphs *adjacent* if they can be reach from one another by a single swap. Using this definition of adjacency, each graph corresponds to a state in the Markov chain. The chain is reversible, in that for each single swap, we can perform a corresponding reverse swap. However, the chain is not regular, since the number of graphs one can arrive to varies among different graphs. We make the chain regular by considering all illegal swaps as being a self-loop to the current state (Gionis et al., 2006). A swap is illegal, for instance, if it would result in duplicate edges. If we define all swaps equally likely and each state has equal number of swaps, be it legal or not, the degree of each state is constant because of the self-loops, and hence the chain is regular.

We propose three different edge swapping methods. The first method, XSwap, follows the idea in (Gionis et al., 2006) and is used in bioinformatics (Sharan et al., 2005). XSwap selects two random edges and swaps two endpoints together, as illustrated in Figure 1a. The swap has the property that it maintains the individual node degrees as well as being very general. XSwap does not maintain connected components in the graph, and therefore, we propose another swap, called LocalSwap, that does not mix edges between connected components and respects the locality of connections. Figure 1b illustrates the LocalSwap. In some situations the individual node degrees do not matter and preserving them could be excessively restrictive. Therefore, we propose a third swap called Flip, which is illustrated in Figure 1c. A random edge and a node are selected. Then either endpoint is selected at random and the swap is done *if* the degree of the endpoint differs by one of the degree of the single selected node. This operation maintains the degree distribution, but changes the individual node degrees. Flip allows the graph to change more freely.



(a) XSwap (b) LocalSwap (c) Flip

Figure 1. Different swaps for randomization. The Flip in c) is further conditioned with $|\delta(n) - \delta(l)| = 1$, where δ is the degree of the node.

If $\rho(G_s)$ is defined as Uniform, using one of these swaps is all that is required for the Markov chain will converge to that distribution. However, if $\rho(G_s)$ is not uniform, the swapping needs to be further controlled. We use Metropolis-Hastings (Hastings, 1970) approach to define state transition probabilities to make the Markov chain have the required steady state distribution. The swap from a graph G to a graph G' is performed with the Metropolis-Hastings probability $\min(1, \frac{\rho(G')}{\rho(G)})$, where $\rho(G)$ is the user defined distribution. Because of this, the Markov chain will have the steady state distribution $\rho(G)$, which is the distribution we want to sample from.

Related Work: Statistical significance testing on graphs is not a new discovery. Bioinformaticians use constructive graph models to define p-values for graphs (Koyutürk et al., 2007). Some use Monte Carlo swapping methods to sample graphs, and to define some empirical probabilities (Sharan et al., 2005). Our work extends these methods by different swappings that can preserve graph statistics.

4. Experiments

We use five different real datasets: Zachary, Adjnoun, Football, Power and Compound. Our experiments focus on the applications of graph clustering and graph pattern mining, but the proposed methods are not limited to these examples. For each setting, the convergence of the Markov Chain was determined by carrying out swaps long enough to see the Frobenius distance between the current and original graph settle.

Using the Zachary, Adjnoun, Football and Power datasets, 100 random graphs were generated for all pairs of an algorithm and a statistic; except for Power and CPL due to the dataset being too large to use with CPL in a reasonable time. After all the random samples were generated, the graphs were clustered with Graclus, readily available from the authors' site[1] as well as a spectral graph clustering method (Yu & Shi, 2003). The algorithms were used to cluster individual graphs from two to 15, 30, and 50 clusters, for Zachary, Adjnoun and Football, and Power datasets, respectively. For each clustering, the minimum cut value, the value of the objective function selected as the test statistic, was stored. Finally, the p-value for each combination of algorithm, dataset and graph statistic was calculated by taking the fraction of minimum cut values for randomized graphs that were *less* than the minimum cut value for the original graph. Results for Zachary and Adjoun are depicted in Figures 2, 3. All results were significant for the Power dataset, whatever the setting.

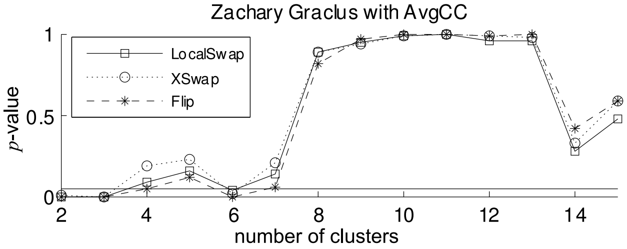


Figure 2. Results for Zachary dataset, all randomization algorithms and AvgCC as graph statistic. The continuous horizontal line signifies the $\alpha = 0.05$ confidence threshold. The p-value rapidly increases to around one at 8 clusters and stays there for several number of clusters. The reason for this is that when the original data is clustered, there is a limit to how many reasonable number of clusters the graph can be divided to. When this value is exceeded, the algorithm has to produce cluster borders within clusters, which results in a high value for the objective function. Since the random graphs do not have this structure, adding one cluster more makes no big difference. Hence the rapid incline. Additionally, p-values for spectral clustering did not exceed the α threshold around 5 clusters. Graclus seems to have trouble clustering to around five clusters, which could be caused by the approximate nature of the algorithm in combination with this dataset.

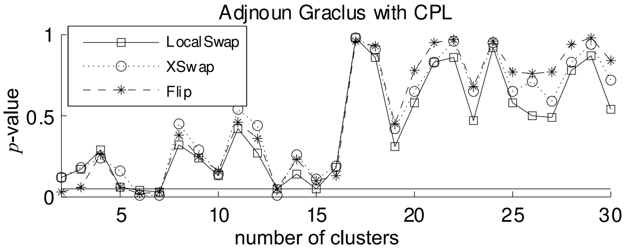


Figure 3. Results for Adjnoun dataset, all randomization algorithms and CPL as graph statistic. The continuous horizontal line signifies the $\alpha = 0.05$ confidence threshold. The p-values vary greatly when the number of clusters is changed, which is consistent with Adjnoun being sampled from $\rho(G_s)$: a p-value of a graph sampled from $\rho(G_s)$ is uniformly distributed in $[0, 1]$.

For the pattern mining experiments, we first generated an artificial dataset and ran the experiments with it. We used the FSGalgorithm, which is a part of Pafi[2], to find the frequent subgraphs in this database. We run the tests with different minimum support values to see if there is any difference. From the results with the original graphs, frequent patterns were stored as well as their support. The p-value of a pattern was taken to be the fraction of randomized graphs with a support higher than the original graph. The results showed that, for both artificial and Compound datasets, all the patterns found by the FSG were not significant, but still more than half of them always were. The additional constraint of restricting the randomizations to maintain AvgCC did not have much effect.

References

Besag, J., & Clifford, P. (1989). Generalized Monte Carlo significance tests. *Biometrica*, *76*, 633–642.

Gionis, A., Mannila, H., Mielikäinen, T., & Tsaparas, P. (2006). Assessing data mining results via swap randomization. *KDD'06*.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrica*, *57*, 97–109.

Koyutürk, M., Szpankowski, W., Grama, A. (2007). Assessing Significance of Connectivity and Conservation in Protein Interaction Networks. JCB, *14*, 747–764.

Sharan, R., Ideker, T., Kelley, B., Shamir, R., Karp, R. (2005). Identification of Proteing Complexes by Comparative Analysis of Yeast and Bacterial Protein Interaction Data. JCB, *12*, 835–846.

Yu, S. X., & Shi, J. (2003). Multiclass spectral clustering. *ICCV'03*.

[1] http://www.cs.utexas.edu/users/dml/Software/graclus.html

[2] http://glaros.dtc.umn.edu/gkhome/pafi/overview