1. Scope

The purpose of this manuscript is to extend the "Graph Classication using Signal Subgraphs" by Vogelstein et al. (Need bibtex reference) to n groups, weighted or categorical edges with more than 2 categories, and a covariate-adjusted classifier.

2. Multiple groups

Although "Graph Classication using Signal Subgraphs" dealt specifically with 2 groups, which is a common application, it is trivial to extend this method to n groups, as Fisher's exact test and χ^2 tests are implemented for n groups and a binary classifier. The only change is now we calculate this separately for each group $P(\mathbb{G} = G|Y = y_i) = \prod_{u,v \in g} p_{uv|y_i}^{a_{u,v}} (1 - p_{uv|y_i})^{(1-a_{u,v})}$.

3. Outline

Talk about moving into weighted signal subgraphs.

4. Weighted/Categorical Classifier

4.1. **Estimation of Statistics for Edge Discrimination.** In order to estimate the signal subgraph or weight edges in classification of a group, we need test statistics on the edges that discriminate the groups to be classified. For categorical or ordered edge values, Fisher's exact test (**fisher.test**) and χ^2 (**chisq.test**) tests will suffice. The problem exists when the computation of Fisher's exact test is too intense, and each cell of the category by group table does not have a count ≥ 5. This will usually happen if there are a large number of groups or large number of edge categories, or both. The extreme example of this is when there are continuous edge weights. In both cases, the following describe some intuitive nonparametric and parametric tests to determine the signal subgraph.

4.1.1. *Non-Parametric Tests.* In order to determine test statistics for edges, one should do some exploratory data analysis (EDA) to determine what parameters of the distribution should be tested upon. For example, if looking at the distribution of values at each edge across groups, and the mean and median appear similar, but the variances appear different across groups, one would not want to do a test of means. Also, one can compare the rank of discrimination across edges across tests.

For ordered data, the extension of the Mann-Whitney U/Rank-sum test, the Kruskal-Wallis (K-W) test can provide χ^2 statistics and p-values for the ability of an edge to discriminate groups due to difference in medians. This defaults to the rank-sum test when the number of groups are two. The advantages of this approach

is that no parametric model is assumed, it is robust and does not lose much efficiency over t-tests even when the data is normal, and is easily implemented. **kruskal.test**

In order to test difference of variances, variance ratio tests such as Levene's test and the BrownForsythe test will allow comparison of group variances, (Brown and Forsythe, 1974; Levene, 1960). **levene.test**

4.1.2. *Parametric/Model Approaches.* Extending to parametric models, one can use a basic ANOVA to get an F statistic, and subsequently a p-value. Although this is useful, the K-W test does not have these assumptions, and results can be greatly skewed if the data is not normally distributed.

In general, this method can be expanded to any generalized linear model (GLM) depending on the type of edge weights are given. For example, if the data is a correlation matrix, one can use the transformation $\tilde{x} = \frac{x+1}{2}$ so that the data $\in [0, 1]$ and a GLM with a beta distribution can be used. If the data were ranks, one may use do Poisson or negative binomial regression. This can be extrapolated to all data that GLMs can model, which is a large set of data. Using likelihood-ratio tests (LR), the group indicator coefficients can be tested and a test statistic and p-value can be used to rank edge discrimination.

With the growth of many machine learning algorithms for discrimination, using these methods may prove more useful than GLMs to rank edges. Recursive partitioning (Breiman, 1984), random forests (Breiman, 2001), and other machine learning algorithms have measures of importance other than β coefficients and p-values, and can be used to determine the signal subgraph. If one decides to use inverse p-value weighting (discussed later), one may be able to use a transformation such as $p_r = \frac{r_i}{\sum_{i=1}^{P} r_i}$, where r_i is the importance measure of edge i of the P edges.

Using some parametric models may be a bit more desirable/wanted, especially if there are a large number of ties within the data. For example, let's say we have edges that have values, 0, 1, 2, where 0 is not connected, 1 is connected (weak) and 2 is connected (strong). One could use the Poisson distribution in a GLM and do the model $G \sim Gr$ where Gr is the group indicator and G is the graph edge. A likelihood ratio test could then get one p-value for how well these groups predict the edge.

5. Generalized Linear Model (GLM) graph classifier

Let's give a couple examples of how we can use GLMs to estimate the probability for an edge. Here's how the procedure would go. Specify a family/distribution, use either Kruskal-Wallis or family to determine signal subgraph, then use the family/distribution to get a probability under that distribution.

5.1. **randomForest.** Using a machine learning algorithm, you can do regression type analyses or classification for discrete outcomes. This is similar to GLM but you do not need to specify a family.

6. Estimation of $P(\mathbb{G} = G|Y = y_i, X)$, probability of an edge

6.1. **Nonparametric.**

6.1.1. *Signal Subgraph (model/estimation at each node).*
Bernoulli/Multinomial.

$$P(Y = y_i|G = g) = \frac{P(\mathbb{G} = G|Y = y_i)P(Y = y_i)}{\sum_{j=1}^{k} P(\mathbb{G} = G|Y = y_j)P(Y = y_j)} \tag{1}$$

$$P(\mathbb{G} = G|Y = y_i)P(Y = y_i) = \prod_{u,v \in g} p_{uv|y}^{a_{u,v}} (1 - p_{uv|y})^{(1-a_{u,v})} \pi_{y_i} \tag{2}$$

where $\pi_{y_i} = P(Y = y_i)$, and k is total possible number of groups and u, v are indices in the graph. Signal subgraph classifier uses

$$P(\mathbb{G} = G|Y = y_i)P(Y = y_i) = \prod_{u,v \in \mathcal{S}} p_{uv|y}^{a_{u,v}} (1 - p_{uv|y})^{(1-a_{u,v})} \pi_{y_i} \tag{3}$$

So for n groups, we can use the binomial distribution still to estimate $P(\mathbb{G} = G|Y = y_i)$. For non-binary outcomes, let's say the groups are "categorical", ie (-1, 0, 1) for negatively connected, not connected, positively connected, then we need to use the multinomial distribution. Thus, we may want to use the generalization of the binomial classifier to the "categorical" classifier.

$$P(\mathbb{G} = G|Y = y_i)P(Y = y_i) = \prod_{u,v \in \mathcal{S}} \left(p_{u_1 v_1|y}^{a_{u,v,1}} p_{u_2 v_2|y}^{a_{u,v,2}} \cdots p_{u_c v_c|y}^{a_{u,v,c}} \right) \pi_{y_i} \tag{4}$$

where c is the number of categories and

$$a_{u,v,i} = \begin{cases} 1 & \text{if } a_{u,v} = c_i \\ 0 & \text{otherwise} \end{cases}$$

and c_i is case. **Problem** - need some sort of numerical perturbation (ie if 0% of people in group A have an edge here, then maybe use 0.001), so not to make the classifier a probability 0 for this group. Josh had this in his paper, which I think works well

$$\hat{p}_{u,v|y} = \begin{cases} \eta_n & \text{if } a_{u,v} = c_i \\ 1-\eta_n & \text{if } \max_i a_{uv}^{(i)} = 0 \\ \hat{p}_{u,v|y}^{MLE} & \text{otherwise} \end{cases}$$

where $\eta = \frac{1}{10n}$.

6.2. **randomForest.** Using a machine learning algorithm, you can do regression type analyses or classification for discrete outcomes. This is similar to GLM but you do not need to specify a family.

6.2.1. *Whole Graph.*
Weight using p-values. Instead of using a derived cutoff for the number/percent of edges for the signal subgraph, the entire graph can be used, inverse weighting using p-values. By "inverse", denoted by $f(x)$ can take different forms, $1-x$, $\frac{1}{x,}$, $\frac{1}{\sqrt{x}}$, etc.

$$P(\mathbb{G} = G|Y = y_i) = \prod_{u,v \in g} f(x) p_{uv|y}^{a_{u,v}} (1 - p_{uv|y})^{(1-a_{u,v})} \pi_{y_i} \tag{5}$$

If not weighting using p-values, then just a dimension reduction approach using Fisher's exact test.

6.3. **GLM/Model Based: Covariate Adjusted.**
Categorical. The same argument can be used for signal subgraph and weighted using p-values.

$$P(Y = y_i|G = g, X) = \frac{P(\mathbb{G} = G|Y = y_i, X)P(Y = y_i|X)}{\sum_{j=1}^{k} P(\mathbb{G} = G|Y = y_j, X)P(Y = y_j|X)} \tag{6}$$

$$P(\mathbb{G} = G|Y = y_i, X)P(Y = y_i|X) = P(\mathbb{G} = G|Y = y_i, X)\hat{\pi}_{y|X} \tag{7}$$

$P(\mathbb{G} = G|Y = y_i, X)$ can be estimated using a GLM or a machine learning algorithm. $P(Y = y_i|X)$ can also be, in the same way (but different model most likely), using multinomial/logistic/any classification. The only caveat is that X should be the same (I think), for it to be a true Bayesian estimator.
Regression. $P(\mathbb{G} = G|Y = y_i, X)$ probably need to be given by fitting separate models for each group and using the predictions from the separate models.

7. Future work

Use GEE/mixed effects models to estimate the covariance matrix of the graph. For correlation matrices (say fMRI), this would be the spatial covariance of the temporal correlation between voxels.

References

Breiman, L. (1984). *Classification and regression trees*. Chapman & Hall/CRC.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brown, M. and Forsythe, A. (1974). Robust tests for the equality of variances. *Journal of the American Statistical Association*, pages 364–367.

Levene, H. (1960). Robust tests for equality of variances. *Contributions to probability and statistics*, 1:278–292.