EN.580.694: Statistical Connectomics
Final Project Proposal

Max Collard · April 2, 2015

Does SBM-style vertex clustering "work" for preferential attachment networks?

Opportunity Recent work (like "Perfect Clustering . . . ") has demonstrated that there is hope for accurately grouping vertices generated by stochastic blockmodels (SBMs), as well as more general random dot product graphs (RDPGs), using mean-square error clustering on the adjacency spectral embedding of the graph.

Challenge Although asymptotically optimal for this particular class of generative models, real-world networks, including connectomes, have been proposed to be generated by alternate mechanisms, which would have different statistical properties than RDPGs. The decay in the optimality of the proposed clustering methods in this setting remains largely unexplored.

Action I will try applying the "Perfect Clustering . . . " methods to three settings, and examine the accuracy as the graph size / applied perturbation grows, or as the number of clusters varies. The classes I want to look at are:

1. An SBM, the "Perfect Clustering . . . " method's home. Here, the clustering accuracy (in detecting which block of the SBM the vertex was generated in) should increase with the size of the graph.

2. A Barabási–Albert (BA) preferential attachment model. Here, the "ground truth" clusters will be subtly different: the initial conditions for the BA evolution will consist of k connected components of equal size, and the clustering will be trying to detect the "majority parent" component, as inherited through connections.

3. The *C. Elegans* connectome, as the starting condition for a BA process of varying length. Here, the goal will be to see how the "Perfect Clustering . . . " method's predictions on the original vertices changes as the network evolves.

Resolution The end goal of this project is to create plots of how the accuracy falls off (in the case of scenarios 1. and 2.) or how the labeling diverges from the unperturbed state (for scenario 3.) as we alter things like number of clusters k or number of final vertices N.

Future Work The ideal future work would be determining the *actual* optimal clustering algorithm for this decision theory problem—though, this would be quite a daunting theoretical problem.

Statistical Decision Theoretic Framework

...

Sample Space Unweighted, undirected graphs, baby:

$$\mathcal{S}_N = \mathcal{G}_N$$

for varying N. Note that these are also jointly distributed with the vertex labels, which live in $\mathcal{A}_N$, as defined below.

Model For the first case, we'll be generating graphs using the

$$\mathcal{M}_1 = \mathrm{SBM}_N^k(\vec{\rho}, B)$$

model, where $\vec{\rho} \in \Delta_K$ and $B \in [0,1]^{k \times k}$.

In the second case, we'll be generating graphs using the BA algorithm, in which new vertices are added to the network sequentially, and connected to existing vertices with probability proportional to the input degree of the existing vertex. In particular, we will be starting this procedure off with k groups of N_k vertices, and stopping when the graph has a total of N; we could denote the class of such models

$$\mathcal{M}_2 = \mathrm{BA}_N^{(k, N_k)}$$

It's extremely important to note, however, that the "Perfect Clustering ..." method is still "treating" this network as an SBM / RDPG; in essence, this project could be thought of as an examination of the projection of BA-generated data onto RDPG-fit models.

Action Space The action space is a labeling of the N vertices into k clusters; that is,

$$\mathcal{A}_N = [k]^N$$

To make the problem easier, we will assume that the true number of clusters k is known. In the *C. Elegans* paradigm, k will be left as a free parameter and varied.

Decision Rule Class This project will be evaluating the performance of the optimal vertex clustering method described in "Perfect Clustering ...". This procedure consists of adjacency spectral embedding of the observed graph, followed by labeling of the vertices by mean-square error clustering. Hence, the decision rule $F : \mathcal{S}_N \to \mathcal{A}_N$ is not random, since it does not need to be trained; in fact, it is

$$F = F_{\mathrm{MSEC}} \circ F_{\mathrm{ASE}}$$

where F_{ASE} is the (deterministic) adjacency spectral embedding function, and F_{MSEC} is the (deterministic) mean square error clustering function.

Loss Functional We'll be evaluating performance using ARI. Specifically, for a decision rule F and a dataset $(G, A^*) \in \mathcal{S}_N \times \mathcal{A}_N$, we will take

$$L[F; (G, A^*)] = \mathrm{ARI}(F(G), A^*)$$

Risk Functional We will use the standard:

$$R[F] = \mathbb{E}[L] = \mathbb{E}_{(G, A^*) \in \mathcal{S}_N \times \mathcal{A}_N}[L[F; (G, A^*)]$$