36-781, Advanced Statistical Network Models **Fall 2016**

Lecture 8: November 22, 2016

Lecturer: Cosma Shalizi *Scribes: Neil Spencer*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

8.1 Agenda

1. Introduction to network comparisons
2. Discussion of the general case of network comparisons, its properties, and the "pooled estimate "test
3. Discussion of network comparisons for continuous latent space models, including embedding, density estimation, and the pooled estimate test.

8.2 Network Comparisons: The Idea

Given two observed graphs g_1 and g_2, could they have come from the same source? That is, did they come from the same probability distribution over graphs?

Use cases

- networks from different populations. Examples: brain networks from normal versus diseased individuals, social networks from different schools, organizations, gangs, academic disciplines
- treatment versus control. Examples: brain networks before and after learning, organizational networks before and after treatment.
- variation across time or space

Uninteresting questions:

- Are g_1 and g_2 the same graph? The answer is almost always no.
- Are g_1 and g_2 isomorphic? Again, the answer is almost always no.

A better question is: How probable is it that graphs as different as g_1 and g_2 came from the same source?

Answering this question requires two things:

1. A way to say how different graphs g_1 and g_2 are.
2. A way to estimate probability distributions over graphs

Requirement 1 can be addressed using graph theory ideas. Requirement 2 is what this class has been building towards.

8.2.1 The Pooled Estimate Test

The pooled estimate test requires three ingredients

- A way of estimating a graph distribution from one (or more) graphs. Examples: stochastic block models, continuous latent space models, graphons, ...
- A measure of distance between the estimates (not necessarily between the graph). Examples: distance between the node densities in continuous latent space models, or L_2 distance between graphon functions.
- The ability to simulate from estimated models.

Figure 8.1 is a diagram of the pooled estimate test. Here, $\hat{\theta}_i = T(g_i)$ and $\hat{\theta}_{\text{p}} = T(g_1, g_2)$. Repeat the simulation step many times to collect distribution of $d(\hat{\theta}_1^*, \hat{\theta}_2^*)$.

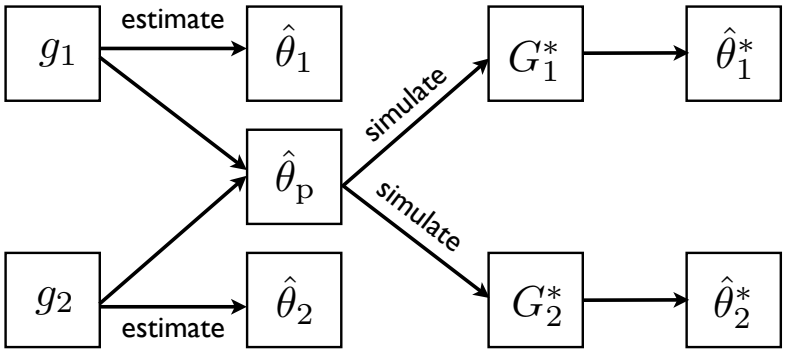

Figure 8.1: A diagram for the pooled estimate test with $\hat{\theta}_i = T(g_i)$ and $\hat{\theta}_{\text{p}} = T(g_1, g_2)$.

To get a p-value, see where the observed distance $d(\hat{\theta}_1, \hat{\theta}_2)$ falls in the distribution from simulations, as seen in Figure 8.2.

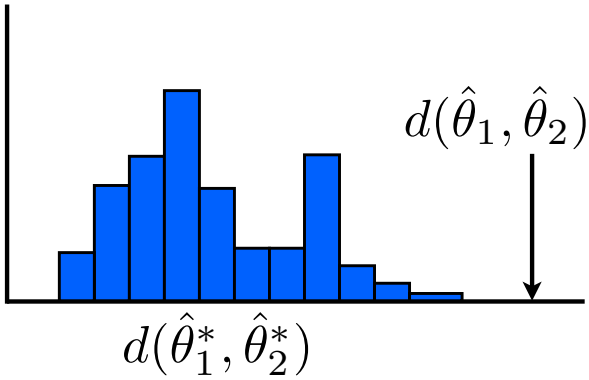

Figure 8.2: An illustration of obtained a p-value in the pooled estimate test

For an analogy, consider the t-test for difference in means. There, the test statistic is the difference in means divided by an estimate from pooling the samples. Instead of simulations, t-tables are used. This is because the central limit theorem is used to shortcut simulation.

To implement the pooled estimate test idea, we need to choose:

- a model class and estimator (the function T)
- distance measure for probability models (d)
- the number of simulations to run (should be as large as practical)

For model choice, the closer the models fit to the actual graph distributions, the more power the test will have. There is trade-off between the expressive power (or range of the model) and its noise-sensitivity (estimation error). This phenomenon is akin to the bias-variance tradeoff, or accuracy versus precision. As often is the case in statistics, it is good to come up with good models ahead of time (before looking at the data) to avoid overfitting.

The choice of distance measure can be straightforward for parametric models. One option is the distance between the parameter vectors. A straightforward variant is to weight the parameters by some measure of influence and/or precision of the estimator. This may be linked via the Fisher information and Cramer-Rao lower bound.

Without parametric models, we need some measure of distance that reflects distances between graph distributions. For continuous latent space models, the estimate is given by an estimated density $\hat{f}$ on a latent space M. Thus, we can use any method we like to compare probability densities. For example, we could use

$$L_1 = ||\hat{f}_1 - \hat{f}_2|| = \int_M |\hat{f}_1(x) - \hat{f}_2(x)|dx \text{ or } L_2 = \left(\int_M (\hat{f}_1(x) - \hat{f}_2(x))^2 dx\right)^{1/2}.$$

Another option would be the symmetric Kullback-Liebler divergence $1/2(\mathrm{KL}(f_2, f_1) - \mathrm{KL}(f_1, f_2))$.

For graphon functions, $w_1 : [0,1]^2 \to [0,1]$ and $w_2 : [0,1]^2 \to [0,1]$, it would be natural to try

$$\left(\int |w_1(u,v) - w_2(u,v)|^p dudv\right) 1/p \text{ for } p = 1, 2, \ldots$$

However, the fact that graphon functions are not unique (the graphs are invariant under measure-preserving maps) means more care must be taken.

One option is to put w_1 and w_2 in "canonical form". For example, order by increasing expected degree, given by $\mathbb{E}(D|U_i = u) = (n-1)\int_0^1 w(u,v)dv$.

An alternative is to find

$$\inf_\phi \left(\int |w_1(u,v) - w_2(\phi(u), \phi(v))|^p dudv\right)^{1/p}$$

where ϕ is a measure preserving map. Finally, if g_1 and g_2 share the same node set, try

$$\sum_{(i,j)} |\hat{p}_{ij}^{(1)} - \hat{p}_{ij}^{(2)}|.$$

8.3 Continuous Latent Space Models

Recap: each node i has a latent position $x_i \in M$.

$$\mathbb{P}(A_{ij} = 1 | X_i, X_j) = w(\rho(X_i, X_j))$$

with ρ a metric on M and all A_{ij} independent given X's. From last time, we know that if M, ρ, w are well behaved, the MLE $\hat{x}_{1:n} \to^p [x^*_{1:n}]$ where x^* is the true coordinates and $[x]$ is the isometry class of x.

If the ML embedding is consistent with $x_i \sim f$ independently, and $\hat{f}$ is consistent for iid data and continuous in the data points, then $\hat{f}(\hat{x}_{1:n}) \to^p [f]$ by the continuous mapping principle (the Mann-Whitney Theorem).

To implement a pooled estimate test:

1. Embed g_1 in M to get $\hat{X}$.
2. Embed g_2 in M to get $\hat{Y}$.
3. Estimate densities $\hat{f}_1$ and $\hat{f}_2$ from $\hat{X}$ and $\hat{Y}$, respectively.
4. Estimate pooled density $\hat{f}$ from $\hat{X}U\hat{Y}$.
5. Draw two samples of size $v(g_1)$ and $v(g_2)$ from $\hat{f}$.
6. Generate G_1^* and G_2^* from these samples.
7. Re-embed to get $\hat{X}^*$, $\hat{Y}^*$ and $\hat{f}_1^*, \hat{f}_2^*$.
8. Calculate $d(\hat{f}_1^*, \hat{f}_2^*)$.
9. Repeat steps 5 through 8 many times and compare $d(\hat{f}_1, \hat{f}_2)$ to this distance.

If (M, ρ) is an ordinary Euclidean space, a kernel density estimate (KDE) is consistent, nonparametric, and minimax.

$$\begin{aligned}\hat{f}(x) &= \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h}) \text{ where } K(u) \text{ is a probability density.} \\ &= K_h * \frac{1}{n}\sum_{i=1}^{n} \delta(x - x_i) \text{ where } * \text{ is convolution}\end{aligned}$$

This generalizes to other kinds of metric space: convolution of basic kernel density with empirical distribution of observations. (Asta 2015)

In Euclid: convolution = multiply Fourier transforms.
Non-Euclid: convolution = multiply Helgason transforms.