The Stochastic Block Model and Module Detection

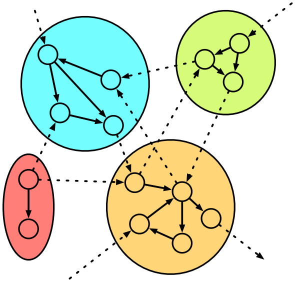
Dave Darmon

January 16, 2014

Overview

Modules and Module Detection

Modules and Module Detection

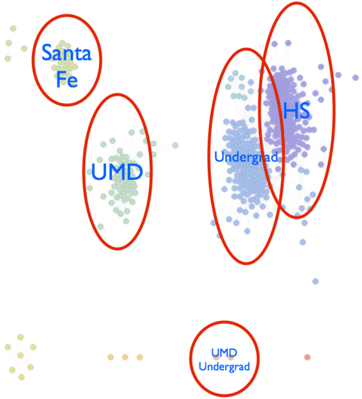


Modules and Module Detection

Basic Idea:

- A *module* or *community* is a collection of nodes defined by how its *edges* behave:
  - **Edge Density:** For social networks, we expect edge density to be greater within a community than without. (Assortative Community)
  - **Edge Weight:** For coexpression networks, we expect the correlations to be higher within a functional module than without.
  - Etc.

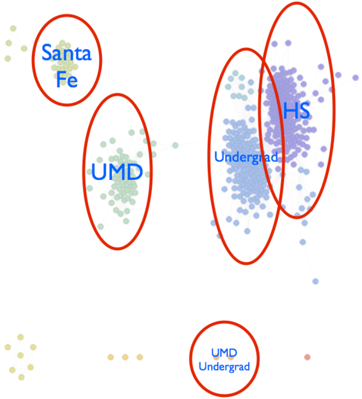
Modules and Module Detection



Modules and Module Detection

- The goal of module detection is to *partition* the network such that each module has a similar distribution over the nodes of within-community edge weights and without-community edge weights.
- The resulting partition will consist of K sets $C_k, k = 1, \ldots, K$, of nodes.
  - $\bigcup_{k=1}^{K} C_k = V = \{1, 2, 3, \ldots, n\}$
  - $C_i \cap C_j = \emptyset, \quad i \neq j$
- Generalizations allow for coverings (partitions where we allow overlap), mixed membership, etc.

Modules and Module Detection



Modules and Module Detection

Questions:

- For a fixed K, how do we decide on a partition?
  - We need some sort of *goodness-of-fit* / *loss* function.
  - This tells us if our partition 'makes sense' / explains the data well.
- How do we choose K?
  - K is a tuning parameter, controls the flexibility of our partition.
  - Usual model checking (leave-one-out cross-validation, etc.) requires a bit of finessing due to the dependencies in the model.

Modules and Module Detection

Approaches:

- Modularity maximization
  - Choose a particular loss function based on a null model for the network (called the configuration model).
- **Stochastic Block Models**
  - Specify a probabilistic model for the network, and use standard techniques from statistical inference.
- *Ad hoc* / heuristic approaches
  - Try things out empirically and hope for the best.

Vanilla Stochastic Block Model

Some notation:

- Let $G = (V, \mathbf{A})$ be a graph / network where:
  - $V = \{1, 2, \ldots, n\}$ indexes the nodes (vertices) in the network.
  - $\mathbf{A}$ is the (binary) adjacency matrix associated with G, such that

$$(\mathbf{A})_{uv} = a_{uv} = \begin{cases} 1 & : \text{an edge exists between } u \text{ and } v \\ 0 & : \text{otherwise} \end{cases}.$$

Vanilla Stochastic Block Model

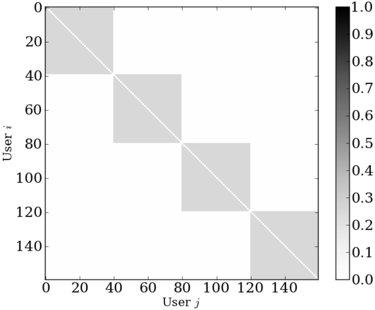
Erdös-Rényi Random Graph:

- We imagine that our network is a realization of a random network where each edge a_{uv} is a realization of a Bernoulli random variable A_{uv} with bias p.
  - $A_{uv} \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$.
- Thus, the number of edges connected to u is Binomial(n, p).
  - For each possible edge incident to u, of which there are n (including self-loops), we flip a coin.
  - Coin flips are 'governed' by the Binomial distribution.
- For small p and large n, we can approximate a Binomial(n, p) distribution with a Poisson(np) distribution.
  - Hence all the claims in Clauset's notes that the degree distribution of an Erdös-Rényi random graph is Poisson, since for real world networks n will be large and p will be small (assuming the network is sparse).

Vanilla Stochastic Block Model

Stochastic Block Model:

- Basically the same model as an Erdös-Rényi random graph, but we allow p to vary between *blocks* of nodes. (Hence the name.)



Vanilla Stochastic Block Model

Parameters of the Binary Stochastic Block Model:

- K, the number of communities / modules.
- $\mathbf{z} = (z_1, z_2, \ldots, z_n)$, a vector giving the community membership for each node.
  - i.e. $z_u \in \{1, \ldots, K\}, u = 1, \ldots, n$.
- $\mathbf{M}$, a $K \times K$ matrix where $m_{ij} = (\mathbf{M})_{ij}$ gives the probability of an edge between a node in community i and a node in community j.
  - i.e. For an edge a_{uv}, we use $\mathbf{z}$ to index into $\mathbf{M}$, and read off $P(A_{uv} = 1) = m_{z_u z_v}$.

Vanilla Stochastic Block Model

See Clauset's notes for examples.

Vanilla Stochastic Block Model

Using a Stochastic Block Model for Simulation:

- Given the parameters $\boldsymbol{\theta} = (K, \mathbf{z}, \mathbf{M})$, we can easily simulate a graph from the model:
  - For each possible entry of $\mathbf{A}$, e.g. between u and v, flip a coin with bias $m_{z_u z_v}$, and record an edge if it comes up 1.
- Similarly, we can write down the probability of generating any graph $G = (V, \mathbf{A})$, since each edge is independent:
  - $$\begin{aligned} P(G = g; \boldsymbol{\theta}) &= \prod_{u,v} P(A_{uv} = a_{uv}) \\ &= \prod_{u,v} m_{z_u z_v}^{a_{uv}} (1 - m_{z_u z_v})^{1 - a_{uv}} \end{aligned}$$
  - We'll need this for inference.

Inference for the Stochastic Block Model

Forward vs. Inverse Probability:

- We know how to generate a network $G = (V, \mathbf{A})$ given $\boldsymbol{\theta} = (K, \mathbf{z}, \mathbf{M})$.
- How do we do the opposite?
- Given $G = (V, \mathbf{A})$, how do we infer $\boldsymbol{\theta} = (K, \mathbf{z}, \mathbf{M})$?
- As always: statistics.

Bayes v. Fisher:

- As with most problems in statistics, there are at least two ways:
    - **The Frequentist Way:** treat $\boldsymbol{\theta} = (K, \mathbf{z}, \mathbf{M})$ as fixed and G as random, and use maximum likelihood to get $\hat{\boldsymbol{\theta}}$.
    - **The Bayesian Way:** treat $\boldsymbol{\theta}$ as random and G as fixed, compute the posterior distribution of $\boldsymbol{\theta}$ given G, and compute the posterior mean / median / mode for $\hat{\boldsymbol{\theta}}$.
- It's interesting to think about what each of these interpretations *mean* in terms of a community / module.

Inference for the Stochastic Block Model

Maximum Likelihood Estimation:

- There are two parts to Maximum Likelihood Estimation for the Stochastic Block Model:
    - Choose $\hat{\mathbf{z}}$, a particular assignment of each of the nodes into the K communities.
        - This is the hard part. NP-complete in general cases.
    - Once we have chosen a $\hat{\mathbf{z}}$, estimate $\hat{\mathbf{M}}$.
        - This is easy, and we can write down the answer.
- Clauset sidesteps the first part. Approximation methods exist.
    - The Expectation-Maximization (EM) algorithm works here.

Inference for the Stochastic Block Model

The Maximum Likelihood Estimator for M:

- See Clauset's notes for details.
- Let N_{ij} be the number of edges between community i and community j. Then

$$\hat{m}_{ij} = \frac{N_{ij}}{n_{ij}}$$

  where n_{ij} is the number of possible edges between communities i and j.
- i.e. Compute the proportion of **potential** edges between i and j that show up in the network.

Inference for the Stochastic Block Model

The Log Likelihood for the Stochastic Block Model:

- $$\begin{aligned} P(G = g; \boldsymbol{\theta}) &= \prod_{u,v} P(A_{uv} = a_{uv}) \\ &= \prod_{u,v} m_{z_u z_v}^{a_{uv}} (1 - m_{z_u z_v})^{1-a_{uv}} \\ &= \prod_{i,j \in \{1,\ldots,K\}} m_{ij}^{N_{ij}} (1 - m_{ij})^{n_{ij} - N_{ij}} \end{aligned}$$

- $$\log P(G; \hat{\boldsymbol{\theta}}) = \sum_{i,j} N_{ij} \log \frac{N_{ij}}{n_{ij}} + (n_{ij} - N_{ij}) \log \left(\frac{n_{ij} - N_{ij}}{n_{ij}} \right).$$

- Allows us to measure the goodness-of-fit of the stochastic block model with respect to the observed network.

Extensions of the Stochastic Block Model

Problems with the Vanilla Stochastic Block Model:

- Degree distribution of any node will always be a mixture of Binomials (approximately a mixture of Poissons).
- To get long-tailed ('power law') degree distributions, the community memberships have to take a very particular form.
- Instead, allow each edge to have a 'propensity' of connecting to other nodes, γ_u, $u = 1, \ldots, n$.
- This allows each node to have its own expected degree, which also depends on its community membership.
- This model is called the degree-corrected stochastic block model, and adds an additional parameter $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)$ to the model.

Extensions of the Stochastic Block Model

Networks with Weighted Edges:

- Instead of having the edges take on binary or non-negative integer values, we assume that edges are drawn from some continuous distribution, with the distribution differing between and within different communities.
- For example, we could assume we have a weighted adjacency matrix $\mathbf{W}$ where

$$W_{uv} \sim N(\mu_{z_u z_v}, \sigma^2_{z_u z_v}).$$

- We have to work a bit harder to infer $\hat{\mathbf{W}}$, but (very recent) methods do exist.