Network Cross-Validation for Determining the Number of Communities in Network Data

Kehui Chen and Jing Lei

SLDM research group

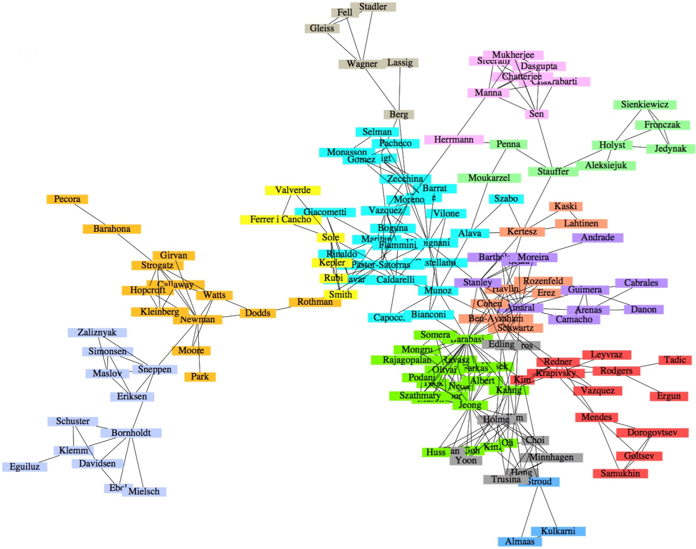
September 17, 2015

Modified from Jing Lei's slides

Network data and community structure

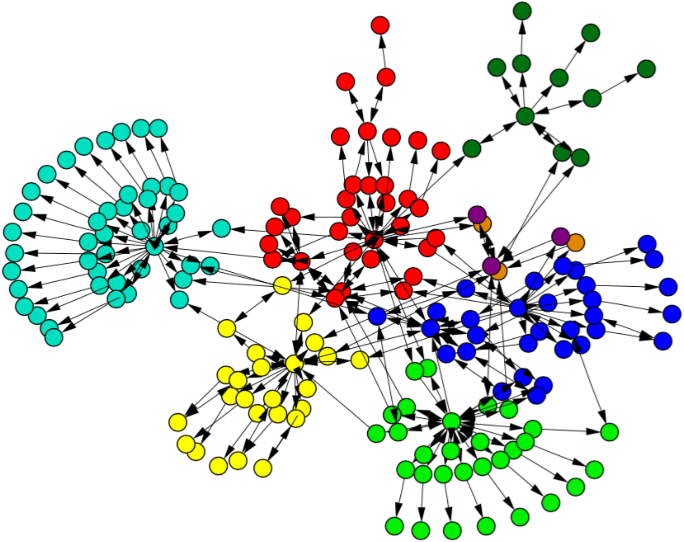
- Network data records interactions (edges) between individuals (nodes).
- Community: a group of nodes with many links (more weights) between themselves and fewer links (less weights) to the rest of the network.

Example: Coauthorships between physicists



[Newman & Girvan '03] A network of coauthorships between physicists who have published on topics related to networks

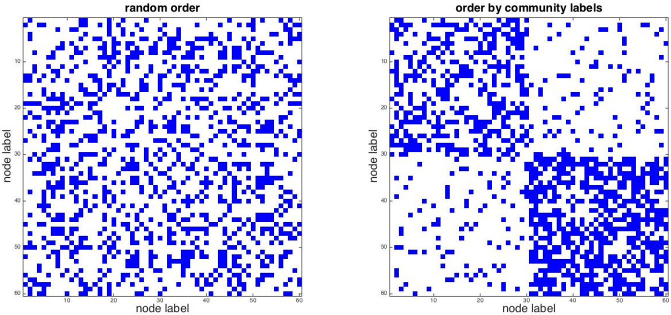
Example: Hyperlinks between pages



[Newman & Girvan '03] Pages on a website and the hyperlinks between them

The stochastic block model (SBM) [Holland et al. ’83]

$B_{11} = 0.4, B_{22} = 0.6, B_{12} = 0.1$, equal-sized communities



The stochastic block model (SBM) [Holland et al. '83]

- Data: adjacency matrix $A \in \{0,1\}^{n \times n}$, where A_{ij} indicates the presence/absence of an edge between nodes pair (i,j).
- $A_{ii} = 0, \forall\ i. A_{ij} = A_{ji}, \forall\ i \neq j$.
- Each node i belongs to a community with label $g_i \in \{1, \cdots, K\}$.
- $B \in [0,1]^{K \times K}$, symmetric, entries are the community-wise edge probabilities.
- Given $g = (g_1, \cdots, g_n), A_{ij} \sim \mathrm{Bernoulli}(B_{g_i g_j})$, independently.
- Nodes in the same community have similar connectivity patterns.

Model selection for SBM: determine the number of communities, K

Cross-validation (CV) for network data

- Cross-validation splits the data so that the fitted model can be validated on an independent sample.
- A naive node splitting method:
  1. Split the nodes into two subsets.
  2. Estimate model parameters using sub-network confined on the fitting set of nodes.
  3. Validate the estimate using the sub-network confined on the testing subset of nodes.
- Does not use the edges between the training and testing nodes.

Network cross-validation (NCV)

- For a given realization of an SBM,
  1. Useful information for inference is mostly contained in edge formulation.
  2. Given the membership variables, edges are independent.
- The sample splitting should be on the edges, not the nodes.

Step 1: block-wise edge splitting

- Given $n_1 < n$, consider a block-split of A:

$$\begin{pmatrix} A^{(11)} & A^{(12)} \\ A^{(21)} & A^{(22)} \end{pmatrix},$$

  where $A^{(11)}$ is the adjacency matrix on n_1 nodes chosen at random.

- Training set: $A^{(1)} = (A^{(11)}, A^{(12)})$
- Testing set: $A^{(22)}$

Step 2: model fitting for a given K

- The rectangular submatrix $A^{(1)}$ carries relationship information for all the nodes in the network.
- Can estimate membership variables from $A^{(1)}$ using spectral clustering.
- Given membership variables, $\hat{B}$ is obtained by taking sample mean of Bernoulli variables in $A^{(1)}$.

Step 3: validation on the testing sample

The validated predictive loss is

$$\hat{L}(A, K) = \sum_{A^{(22)}} l(A_{ij}, \hat{P}_{ij}) \,,$$

where

- the sum is over all pairs (i, j) in $A^{(22)}$.
- $\hat{P}_{ij} = \hat{B}_{\hat{g}_i \hat{g}_j}$.
- $l(\cdot, \cdot)$ is a loss function, e.g., negative log-likelihood:

$$l(a, p) = -a \log p - (1 - a) \log(1 - p) \,.$$

V-fold network cross validation

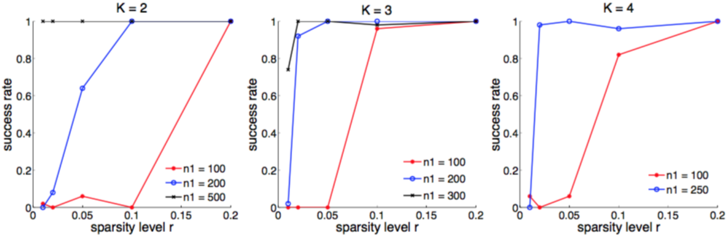
- Randomly split A into $V \times V$ equal-sized blocks:

$$A = \left(A^{(rs)} : 1 \le r, s \le V\right).$$

- For each candidate K, for each $1 \le v \le V$,
  training: $A^{(-v)} = (A^{(rs)} : r \neq v, 1 \le r, s \le V)$
  testing: $A^{(vv)}$
  parameter estimates: $(\hat{g}^{(v)}, \hat{B}^{(v)})$ using $A^{(-v)}$
  predictive loss:
  $\hat{L}^{(v)}(A, K) = \sum_{A^{(vv)}} l(A_{ij}, \hat{P}_{ij}^{(v)}), \hat{P}_{ij}^{(v)} = \hat{B}^{(v)}_{\hat{g}_i^{(v)} \hat{g}_j^{(v)}}$.
- Model selection: $\hat{K} = \arg\min_K \sum_{v=1}^{V} \hat{L}^{(v)}(A, K)$.

Simulation: sparsity and community imbalance

$B = rB_0, B_0(k,k) = 3, B_0(k,k') = 1(k \neq k'), n = 1000,$
community size: n_1 for the smallest, $(n - n_1)/(K-1)$ for others.
Plotted are success rates in 50 repetitions using three-fold NCV.



Extension to the degree corrected block model

The degree corrected block model (DCBM)

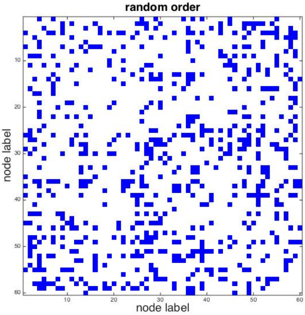
- Limitation of SBM: node degrees are clustered.
- Extension: degree corrected block model [Karrer & Newman '11]

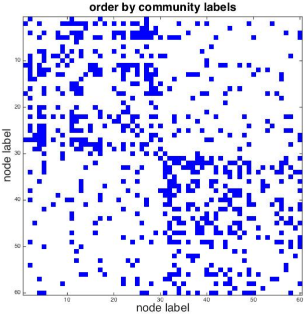
  $$A_{ij} \sim \text{Bernoulli}(\phi_i \phi_j B_{g_i g_j})$$

  $\phi_i \in (0, 1]$: activeness of node i.
- DCBM allows for arbitrary degree distribution.

The degree corrected block model (DCBM)

$$B_{11} = B_{22} = 0.8, B_{12} = 0.2, \phi_i \overset{\text{iid}}{\sim} \text{Unif}(0.2, 1).$$





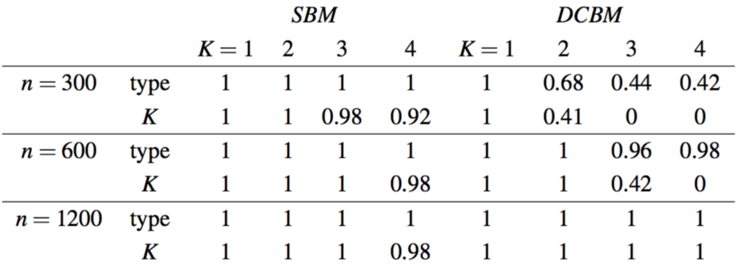
More variations in the node degrees compared to SBM.

Extension to DCBM

- NCV can be extended to the degree corrected block model.
- Only need to modify the parameter estimation step. Perform a spherical spectral clustering method for DCBM.
- NCV can simultaneously select between the regular SBM and the DCBM, and choose K.
- Just compute $\hat{L}_{\text{sbm}}(A, K)$ and $\hat{L}_{\text{dcbm}}(A, K)$ for all candidate K, and pick the best model that attains the overall minimum.

Simulation: simultaneously pick model type and K

$B(k,k) = 0.25, B(k,k') = 0.1 (k \neq k')$, equal-sized communities.
$\phi_i \sim \mathrm{Unif}(0.2, 1)$ for DCBM.
Reporting success rates of three-fold NCV in choosing model type and K in 50 repetitions.



		SBM				*DCBM*			
		$K=1$	2	3	4	$K=1$	2	3	4
$n=300$	type	1	1	1	1	1	0.68	0.44	0.42
	K	1	1	0.98	0.92	1	0.41	0	0
$n=600$	type	1	1	1	1	1	1	0.96	0.98
	K	1	1	1	0.98	1	1	0.42	0
$n=1200$	type	1	1	1	1	1	1	1	1
	K	1	1	1	0.98	1	1	1	1

Discussion

In general, NCV is applicable to network models where

1. edges form independently given an appropriate set of model parameters; and
2. the model parameters can be estimated accurately using a subset of rows of the adjacency matrix.

Thank you!