Network Model Comparison using Network Cross-Validation

Jing Lei
Department of Statistics, Carnegie Mellon University

Nonparametric Statistics Workshop, Ann Arbor, MI

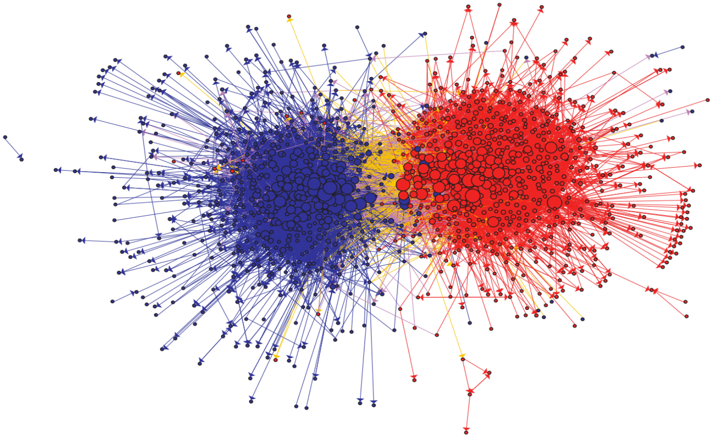
Oct 6, 2016

Based on joint work with Kehui Chen (U. Pitt.)

Network data

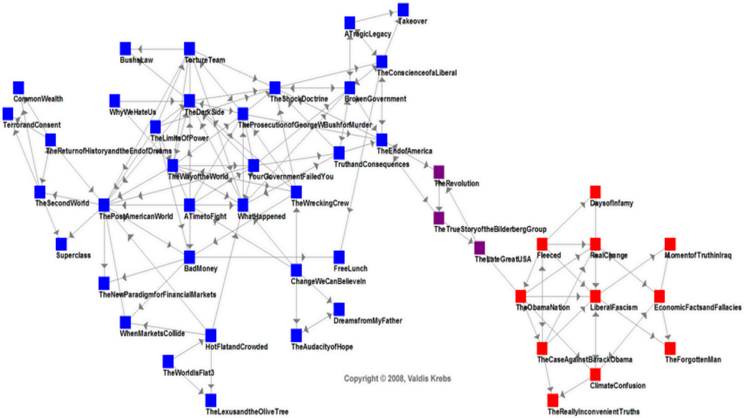
- Network data record interactions (edges) between individuals (nodes).
- From WIKIPEDIA: "... a complex network is a graph (network) with non-trivial topological features ..."
- Examples of "non-trivial topological features"
  - heavy-tail degree distribution (a.k.a "scale-free", "power law")
  - large clustering coefficient (transitivity)
  - community structure: the nodes can be grouped into subsets with dense internal connection.
  - ...

Example: U.S. Political Blogs



[Adamic & Glance '05] The political blogosphere and the 2004 US election: divided they blog

Example: Amazon Books



[V. Krebs '04] Co-purchased political books on Amazon.

The Exchangeable Random Graph Model

- Basic idea: the node ordering carries no information.
- In other words, the random graph is jointly row-column exchangeable.
- de Finetti for two-way array (Hoover '79, Aldous '81, Bickel & Chen '09): All such random graphs must be generated as

$$\xi_i \overset{iid}{\sim} \text{Unif}(0,1), \quad i = 1, \ldots, n\,.$$

$$(A_{ij}|\xi) \overset{indep.}{\sim} \text{Bernoulli}(f(\xi_i, \xi_j))\,.$$

where $f : [0,1]^2 \mapsto [0,1]$, measurable and symmetric, is called a graphon.

Popular Special Cases

- The stochastic block model (SBM, Holland *et al* '83): f is block-wise constant.
- The degree corrected block model (DCBM, Karrer & Newman '11): f is block-wise rank-one.
- Smooth graphon (Wolfe & Olhede '13, Airoldi *et al* '13, Gao *et al* '15): f is smooth.

Inference Problems

- Estimation
  - Community recovery: find block structure of f in SBM and DCBM.
  - Graphon estimation: estimate f, assuming smoothness.
- Model selection
  - How many communities are there?
  - Shall I use SBM, or DCBM, or a smooth graphon to fit my data?
  - How smooth is f? What tuning parameter(s) shall I use in estimation?

Choosing number of communities in SBM/DCBM

- Information criteria: [Handcock *et al* '07], [Daudin *et al* '07], [Airoldi *et al* '08].
- Penalized likelihood: [Wang & Bickel '15], [Saldana *et al* '16].
- Hypothesis testing: [Bickel & Sarkar '15], [Lei '16].
- Spectral methods: [Le & Levina '15]

Network Cross-validation (Chen & Lei '16)

- Why cross-validation?
  1. CV is conceptually simple, statistically principled, and easy to implement (the only tuning parameter is the number of folds).
  2. CV can be used to compare non-nested models, such as SBM vs DCBM vs smooth graphon.

Network Cross-validation (Chen & Lei '16)

- Why cross-validation?
  1. CV is conceptually simple, statistically principled, and easy to implement (the only tuning parameter is the number of folds).
  2. CV can be used to compare non-nested models, such as SBM vs DCBM vs smooth graphon.
- Challenges
  1. Cross-validation splits the data so that the fitted model can be validated on an independent sample.
  2. Challenges: How to split the network? Where to find independence?

Starting Example: Choosing K in SBM

- Data: $(A_{ij} : 1 \leq i < j \leq n)$ satisfying

$$A_{ij} \overset{indep.}{\sim} \text{Bernoulli}(B_{g_i g_j})$$

with unknown parameters

1. $g \in \{1, ..., K\}^n$, the membership vector;
2. $B = B^T \in [0, 1]^{K \times K}$, the community-wise edge probability matrix.

Starting Example: Choosing K in SBM

- Data: $(A_{ij} : 1 \leq i < j \leq n)$ satisfying

$$A_{ij} \overset{indep.}{\sim} \text{Bernoulli}(B_{g_i g_j})$$

with unknown parameters

1. $g \in \{1, ..., K\}^n$, the membership vector;
2. $B = B^T \in [0,1]^{K \times K}$, the community-wise edge probability matrix.
3. K, the number of communities.

Starting Example: Choosing K in SBM

- Data: $(A_{ij} : 1 \leq i < j \leq n)$ satisfying

$$A_{ij} \overset{indep.}{\sim} \text{Bernoulli}(B_{g_i g_j})$$

  with unknown parameters

  1. $g \in \{1, ..., K\}^n$, the membership vector;
  2. $B = B^T \in [0,1]^{K \times K}$, the community-wise edge probability matrix.
  3. K, the number of communities.

- Goal: estimate K for a given A.

A naive node splitting method

- For a given K, treat the node memberships as random and independent: $P(g_i = k) = \pi_k$, where $\sum_{k=1}^{K} \pi_k = 1$, $\pi_k \geq 0$.
- Split the nodes into two subsets.
- Estimate $(\hat{\pi}, \hat{B})$ using sub-network on the training nodes.
- Validate the estimate using the sub-network on testing nodes, treating node memberships as missing variables.

A naive node splitting method

- For a given K, treat the node memberships as random and independent: $P(g_i = k) = \pi_k$, where $\sum_{k=1}^{K} \pi_k = 1$, $\pi_k \geq 0$.
- Split the nodes into two subsets.
- Estimate $(\hat{\pi}, \hat{B})$ using sub-network on the training nodes.
- Validate the estimate using the sub-network on testing nodes, treating node memberships as missing variables.
- Drawbacks:
  1. Missing memberships make it computationally hard.
  2. 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 vector, edges are independent.

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 vector, edges are independent.
- The sample splitting shall be made on the node-pairs, not the nodes.

Step 1: block-wise node-pair splitting

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

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

  where $A^{(11)}$ is the adjacency matrix on n_1 nodes chosen at random.
- Training set of node pairs: $A^{(1)} = (A^{(11)}, A^{(12)})$
- Testing set of node pairs: $A^{(22)}$

Step 2: model fitting for a given K

- Observation: the rectangular submatrix $A^{(1)}$ contains full information of the model, provided that n_1 is not too small.
- Most community recovery methods can be extended to the rectangular submatrix.
- We have implemented three estimators of g: profile likelihood, least squares, and spectral clustering.
- Given $\hat{g}$, $\hat{B}$ is obtained by taking sample means of the Bernoulli random variables in corresponding blocks of $A^{(1)}$.

Step 3: validation on the testing sample

The validated predictive loss is

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

where

- the summation is over all pairs (i,j) in $A^{(22)}$ and $i \neq j$;
- $\hat{P}_{ij} = \hat{B}_{\hat{g}_i \hat{g}_j}$;
- $\ell(\cdot,\cdot)$ is a loss function, for example
  1. negative log-likelihood: $\ell(a,p) = -a\log p - (1-a)\log(1-p)$;
  2. squared loss: $\ell(a,p) = (a-p)^2$.
- Can treat other observation models, such as Poisson, Gaussian, etc.

V-fold network cross validation

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

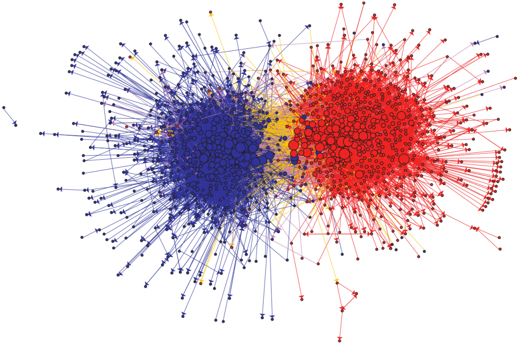
$$A = (A^{(rs)} : 1 \le r, s \le V).$$

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

Extension to DCBM

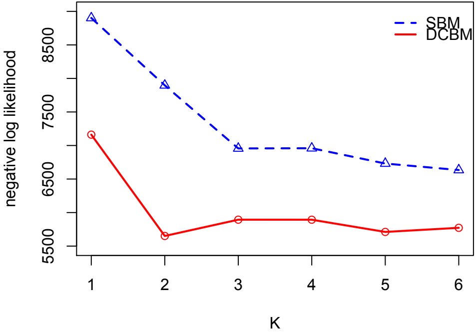
- NCV can be extended to the degree corrected block model: $A_{ij} \sim \text{Bernoulli}(B_{g_i g_j} \psi_i \psi_j)$.
- ψ can be estimated — up to scaling — when $\hat{g}$ is available.
- NCV can simultaneously select between the regular SBM and the DCBM, and choose K.
- Just calculate $\hat{L}_{\text{sbm}}(A, K)$ and $\hat{L}_{\text{dcbm}}(A, K)$ for all K, and pick the overall minimum.

Data example: U.S. political blogs



- [Adamic & Glance '05] Snapshots of weblogs shortly before 2004 U.S. Presidential Election. Nodes: weblogs; edges: hyperlinks.
- Fitted well by a DCBM with two clusters.

NCV on political blog data



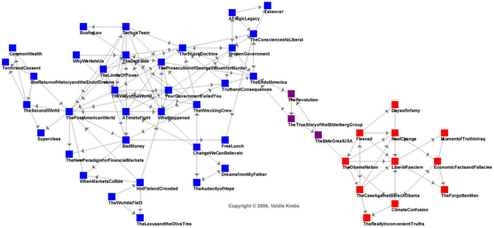
- Plotted: cross-validated predictive loss.
- Spherical spectral clustering with $K = 2$ recovers with 95% accuracy.

Code is available at `www.stat.cmu.edu/~jinglei/code.shtml`.

Reducing Variability via Split-Aggregate

- The random data splitting introduces additional variability in the selected model.
- Split-aggregate: repeat NCV many times using independent splits, and output the most frequent estimate.

Political Books Data



Co-purchase of political books on Amazon (V. Krebs '04)

3-fold NCV results from 100 random splits

$\hat{K}$	1	2	**3**	4	5	≥ 6
Frequency	0	11	**52**	15	13	9

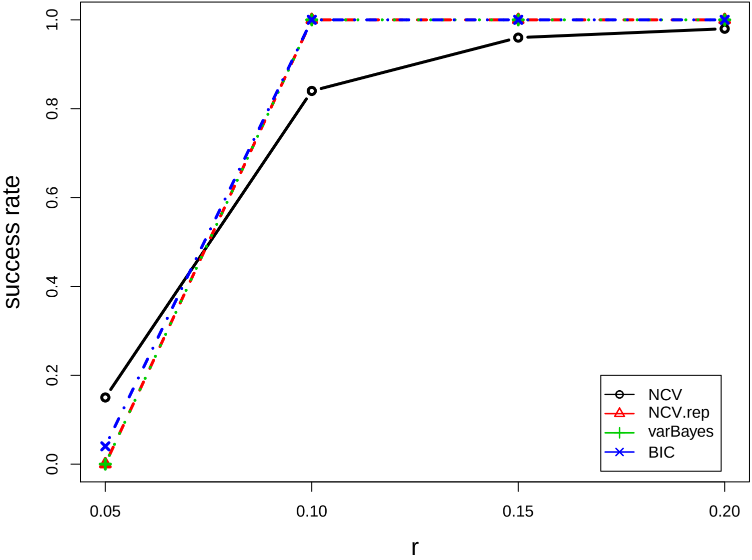
Competing methods: $\hat{K}_{\text{BIC}} = 4$, $\hat{K}_{\text{BHm}} = 3$, $\hat{K}_{\text{BHa}} = 4$.

Code is available at `www.stat.cmu.edu/~jinglei/code.shtml`.

Simulation: choosing K in SBM

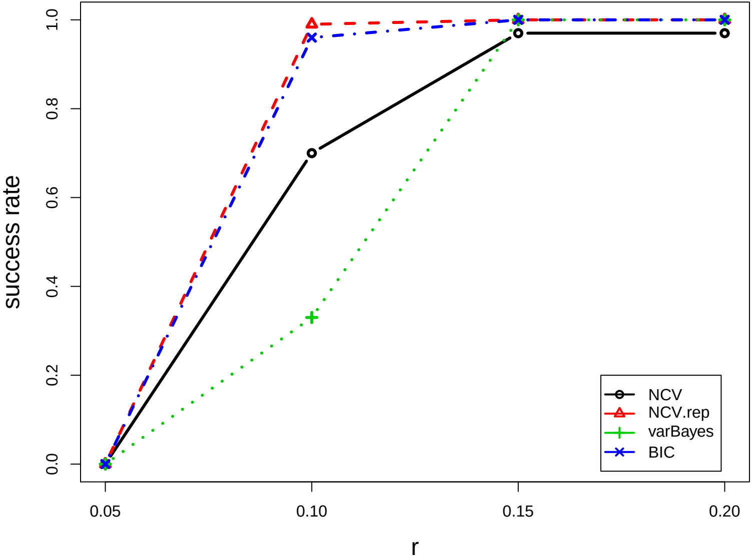
- $n = 600$
- $K = 3, 4, 5$ balanced communities
- B: $B_{k,k'} = 2r$ for $k \neq k'$, and
    - $\text{diag}(B) = (3r, 2r, r)$ for $K = 3$
    - $\text{diag}(B) = (3r, 3r, r, r)$ for $K = 4$
    - $\text{diag}(B) = (3r, 3r, 2r, r, r)$ for $K = 5$
- Compare NCV with aggregated NCV over 20 repetitions, and other methods.

$K = 3$



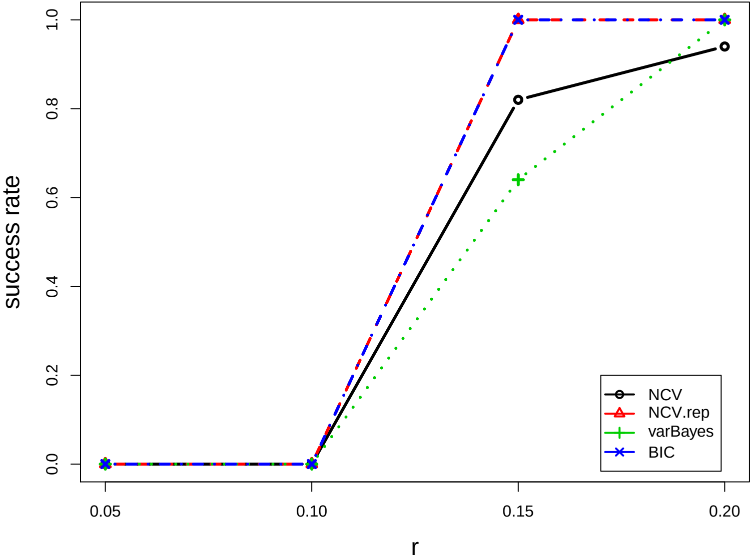
success rate
0.0
0.2
0.4
0.6
0.8
1.0
0.05
0.10
0.15
0.20
r
NCV
NCV.rep
varBayes
BIC

$K = 4$



1.0
0.8
0.6
0.4
0.2
0.0
success rate
0.05
0.10
0.15
0.20
r
NCV
NCV.rep
varBayes
BIC

$K = 5$



1.0
0.8
0.6
0.4
0.2
0.0
success rate
0.05
0.10
0.15
0.20
r
NCV
NCV.rep
varBayes
BIC

Beyond SBM and DCBM

- Can we bring smooth graphons in the comparison?
- Yes
  1. When f is smooth, it is approximately low rank.
  2. Under block-wise node-pair splitting, the spectral decomposition of rectangular adjacency matrix provides estimation of $\mathbb{E}(A_{ij})$ for all $1 \leq i < j \leq n$.
  3. NCV can be applied to select the number of components, as well as to compare the low rank graphon fit with other models such as SBM and/or DCBM.

References

1. Chen, K. & Lei, J. (2016+) "Network Cross-Validation for Determining the Number of Communities in Network Data", *JASA T&M*, to appear. *arXiv:1411.1715*.
2. Code: `www.stat.cmu.edu/~jinglei/code.shtml`
3. Slides: `www.stat.cmu.edu/~jinglei/201610_nonparametric.pdf`

Thank You!

Questions?