Statistical Analysis of Network Data

Lecture 3: Network Modeling

Eric D. Kolaczyk

Dept of Mathematics and Statistics, Boston University

kolaczyk@bu.edu

BOSTON UNIVERSITY

Outline

BOSTON UNIVERSITY

Topics this Lecture

Recall our lecture schedule for statistics:

- L1 Introduction, Background, and Descriptive Statistics (1.5hrs)
- L2 Network Sampling (1hr)
- L3 Network Modeling (1.5hrs)
- L4 Additional Topics in Modeling/Analysis (1.5hr)

In this lecture, we turn to modeling as it relates to networks.

Topics (cont)

We will look at two complementary scenarios[1]:

1. we observe a network G (and possibly attributes $\mathbf{X}$) and we wish to model G (and $\mathbf{X}$);
2. we observe the network G, but lack some or all of the attributes $\mathbf{X}$, and we wish to infer $\mathbf{X}$.

These are, of course, caricatures. Reality can be more complex!

[1] A third option, that we observe attributes $\mathbf{X}$, but lack some or all of the network G, and we wish to infer G, will be the topic of our fourth statistics lecture. This is the topic of network topology inference.

Outline

BOSTON UNIVERSITY

Probabilistic/Mathematical vs. Statistical Network Models

The models encountered in probability/mathematics serve various useful purposes, but arguably come up short as statistical models.

> *"A good [statistical network graph] model needs to be both estimable from data and a reasonable representation of that data, to be theoretically plausible about the type of effects that might have produced the network, and to be amenable to examining which competing effects might be the best explanation of the data."*
>
> *Robins & Morris (2007)*

High Standards

Statisticians demand a great deal of their modeling:

1. theoretically plausible
2. estimable from data
3. computationally feasible estimation strategies
4. quantification of uncertainty in estimates (e.g., confidence intervals)
5. assessment of goodness-of-fit
6. understanding of the statistical properties of the overall procedure

Classes of Statistical Network Models

Roughly speaking, there are network-based versions of three canonical classes of statistical models:

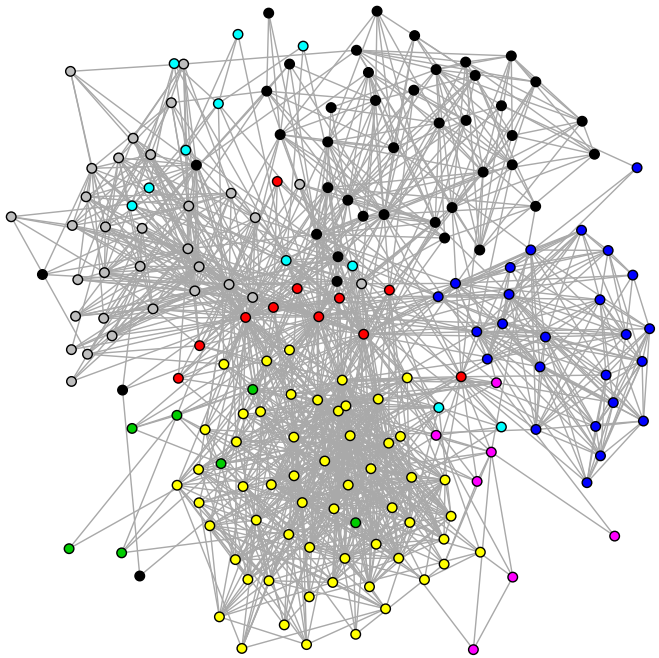
1. regression models (i.e., ERGMs)
2. latent variable models (i.e., latent network models)
3. mixture models (i.e., stochastic blocks models)

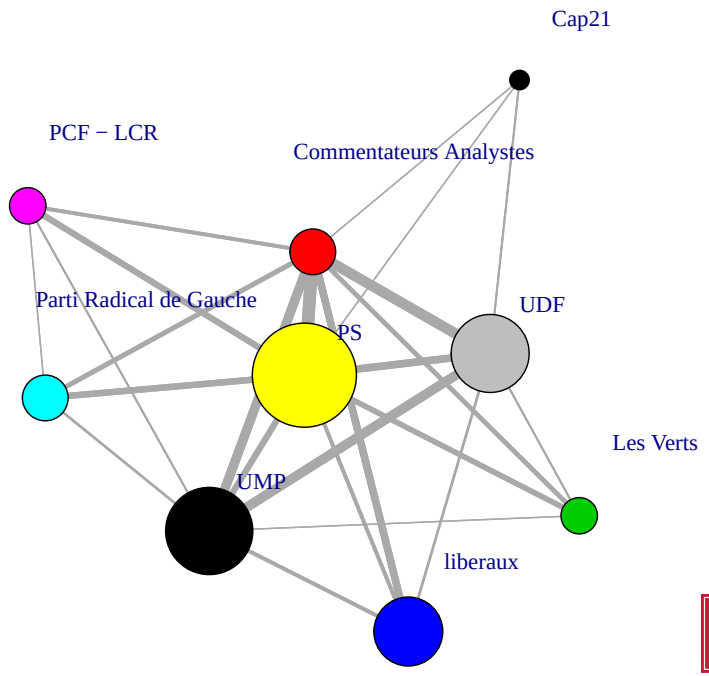
In the interests of time, we will look at just one of these – stochastic block models.

Stochastic Block Models: Motivation

Canonical random graph models (e.g., Erdos-Renyi, Barbasi-Albert, etc.) typically are not flexible enough to capture sufficiently the structure of observed network graphs.

Consider, for example, the network of French political blogs we saw earlier.





Stochastic Block Models: Details

Stochastic block models explicitly parameterize the notion of

- groups/modules, say $\mathcal{C}_1, \ldots, \mathcal{C}_Q$, with
- different rates of connections between/within.

More specifically, this is a generative model, where

1. Each vertex independently belongs to a group $\mathcal{C}_q$ with probability α_q, where $\sum_{q=1}^{Q} \alpha_q = 1$.
2. For vertices $i, j \in V$, with $i \in \mathcal{C}_q$ and $j \in \mathcal{C}_r$, the probability that $\{i, j\} \in E$ is π_{qr}.

This is, effectively, a mixture of classical random graphs,
e.g., note that the probability that there is no edge between i and j is

$$1 - \sum_{1 \leq q,r \leq Q} \alpha_q \alpha_r \pi_{qr} \quad .$$

Stochastic Block Models: Inference

Stochastic block models are defined up to parameters

- $\{\alpha_q\}_{q=1}^{Q}$ and
- $\{\pi_{qr}\}_{1 \leq q,r \leq Q}$.

Suggests, conceptually, thinking of parallel sets of observations

- $\mathbf{Z} = \{\{Z_{iq}\}_{q=1}^{Q}\}_{i \in V}$, for $Z_{iq} = I_{i \in \mathcal{C}_q}$; and
- $\mathbf{Y} = (Y_{ij})$, where $Y_{ij} = I_{\{i,j\} \in E}$.

NOTE: We observe $\mathbf{Y}$ but not $\mathbf{Z}$.

BOSTON UNIVERSITY

Stochastic Block Models: Inference (cont.)

If we were to observe both $\mathbf{Z}$ and $\mathbf{Y}$, then the log-likelihood would take the form

$$\ell(\mathbf{y}; \{\mathbf{z}_i\}) = \sum_i \sum_q z_{iq} \log \alpha_q + \frac{1}{2} \sum_{i \neq j} \sum_{q \neq r} z_{iq} z_{jr} \log b(y_{ij}; \pi_{qr}) \ , \qquad (1)$$

where $b(y; \pi) = \pi^y (1 - \pi)^{1-y}$.

Since we do not, in principle we obtain the likelihood through

$$\mathbb{P}(\mathbf{Y} = \mathbf{y}) = \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{Y} = \mathbf{y}, \mathbf{Z}) \ .$$

Unfortunately, however, this is intractable in problems of any real interest.
$\Longrightarrow$ Computationally intensive methods are needed.

BOSTON UNIVERSITY

Computationally-Intensive Methods for SBMs

There are three main computational approaches for inference with SBMs.

- EM Algorithm Snijders and Nowicki (1997)
  Only for $K = 2$ classes.
- Bayesian algorithm Nowicki and Snijders (2001)
  Extends to arbitrary K, but only feasible on relatively small (100 node?) graphs.
- Variational EM Daudin, Picard, and Robin (2008)
  Using variational methods for optimization to approximate the EM algorithm, using principles related to the mean-field approximation in physics, and allowing for SBMs to be fit to thousand-node graphs.

Illustration: MixER

Applying the R package **mixer** to the French blog data

```
> fblog.sbm <- mixer(as.matrix(get.adjacency(fblog)),
+                    qmin=2,qmax=15)
```

we obtain a model

```
> fblog.sbm.output <- getModel(fblog.sbm)
> names(fblog.sbm.output)
[1] "q"  "criterion"  "alphas"  "Pis"  "Taus"
```

for which the data were fit with

```
> fblog.sbm.output$q
[1] 12
```

classes, in estimated proportions

```
> fblog.sbm.output$alphas
 [1] 0.15119107 0.12735912 0.10799621 0.05729167
 [5] 0.13583884 0.03123704 0.12380263 0.09333527
 [9] 0.01041667 0.02089739 0.12500909 0.01562500
```

BOSTON UNIVERSITY

Illustration: MixER (cont)

Posterior estimates of class-membership may be used to assign vertices to groups.

E.g., For the first three vertices in the French blog network

```
> fblog.sbm.output$Taus[,1:3]
               [,1]         [,2]         [,3]
 [1,] 9.999748e-01 0.0049322556 9.999816e-01
 [2,] 1.402040e-05 0.0000000001 7.588859e-07
 [3,] 9.509516e-06 0.9941965686 1.756997e-05
 [4,] 1.000000e-10 0.0000000001 1.000000e-10
 [5,] 1.338517e-06 0.0000000001 8.213317e-09
 [6,] 1.000000e-10 0.0000000001 1.000000e-10
 [7,] 3.529513e-07 0.0008711750 6.109566e-08
 [8,] 1.000000e-10 0.0000000001 1.000000e-10
 [9,] 1.000000e-10 0.0000000001 1.000000e-10
[10,] 1.000000e-10 0.0000000001 1.000000e-10
[11,] 4.446509e-09 0.0000000001 1.000000e-10
[12,] 1.000000e-10 0.0000000001 1.000000e-10
```

a *maximum a posteriori* criterion would assign vertices 1 and 3 to $\mathcal{C}_1$, and vertex 2 to $\mathcal{C}_2$.

BOSTON UNIVERSITY

Properties of SBM Inference

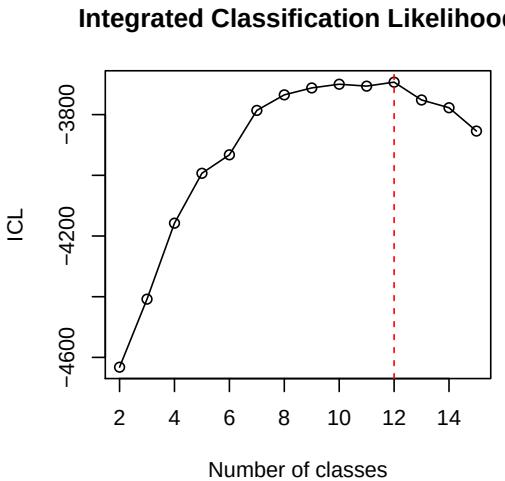
A great deal of recent work has focused on properties of inferences made in the context of SBMs.

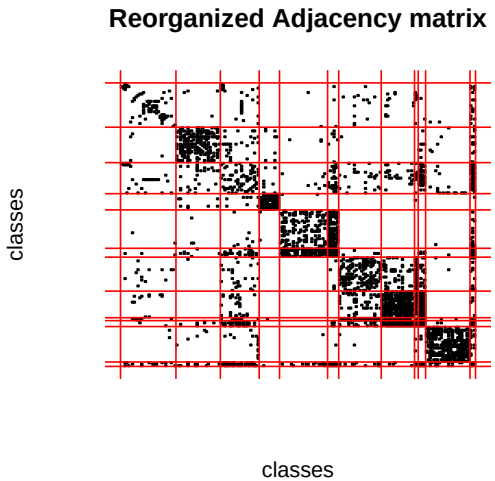
Examples include Bickel and Chen (2009); Choi, Wolfe, and Airoldi (2010); Célisse, Daudin, and Pierre (2011); and Rohe, Chatterjee, and Yu (2010).

Results relate to asymptotic correctness of classification (i.e., group membership), plus properties of parameter estimates in some cases.

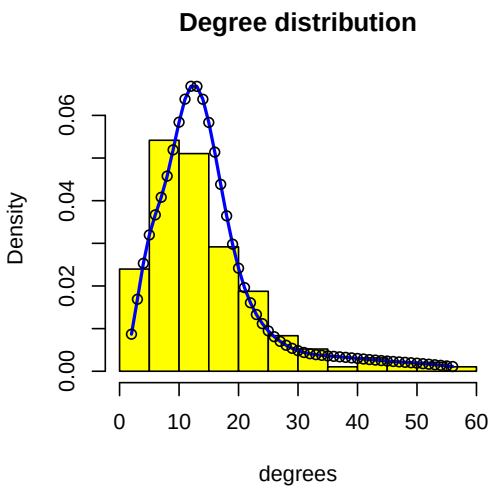
Goodness of Fit?

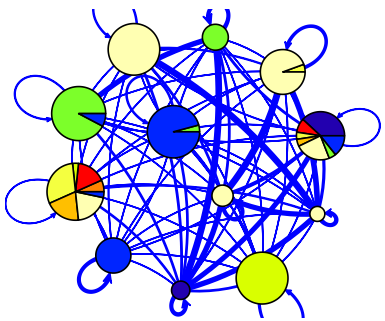
The assessment of model goodness-of-fit is still a relatively new and undeveloped topic in network modeling.






```
1  > plot(fblog.sbm,
2  +   classes=as.factor(V(fblog)$PolParty))
```





Extensions of Stochastic Block Models

There are a variety of extensions of SBMs.

Two important ones are

- Mixed-membership Stochastic Block Models Airoldi *et al.* (2008)

  Nodes may belong only partially to more than one class.

- Hierarchical Block Models Clauset, Moore, and Newman (2008)

  A cross of hierarchical structures (i.e., think of hierarchical clustering) and SBMs.

Also, see Doreian *et al*'s book on generalized stochastic block modeling.

BOSTON UNIVERSITY

Outline

Processes on Network Graphs

So far we have focused on network graphs, as representations of *network systems of elements and their interactions*.

But often it is *some quantity associated with the elements* that is of most interest, rather than the network *per se*.

Nevertheless, such quantities may be *influenced by* the interactions among elements.

Examples:

- Behaviors and beliefs influenced by social interactions.
- Functional role of proteins influenced by their sequence similarity.
- Computer infections by viruses may be affected by 'proximity' to infected computers.

BOSTON UNIVERSITY

Problem Context

Let $G = (V, E)$ be a network graph.

Our interest in this second part of the lecture is in the case of static processes

$$\{X_i\}_{i \in V}$$

and problems where the structure of G may be useful for predicting elements of $\mathbf{X}$.

Illustration: Predicting Protein Function

Proteins are fundamental to the complex molecular and biochemical processes within organisms.

Understanding their role – or 'function' – is critical in biology and medicine.

It has been conjectured that, on average, as many as 70% of genes in a genome code for proteins with poorly known or unknown functions.

$\Longrightarrow$ Prediction of protein function a task of fundamental interest.

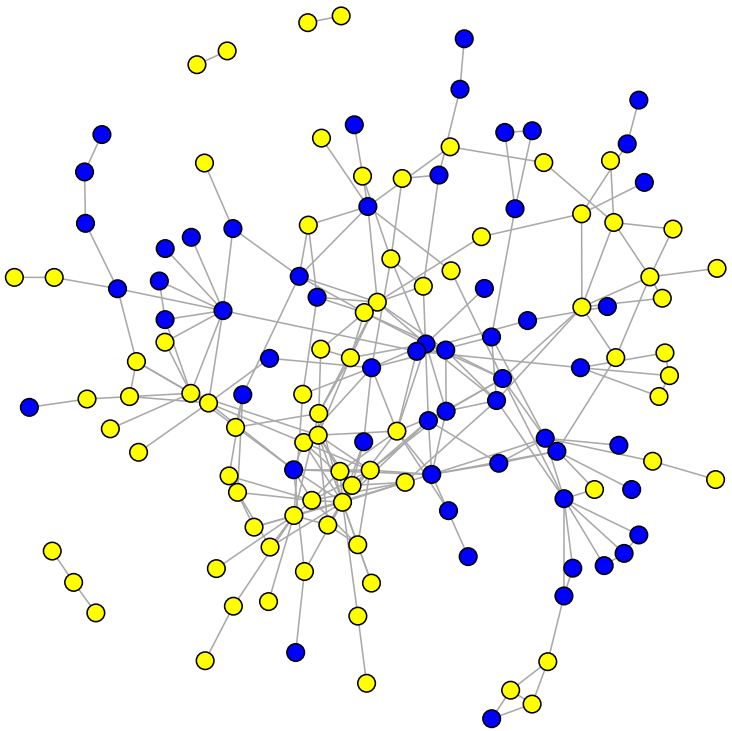
Predicting Protein Function (cont.)

Options that have been explored include

- traditional experiment-intensive approaches
- methods based on sequence-similarity, protein structure, etc.
- network-based methods.

Networks of protein-protein interactions are natural here.

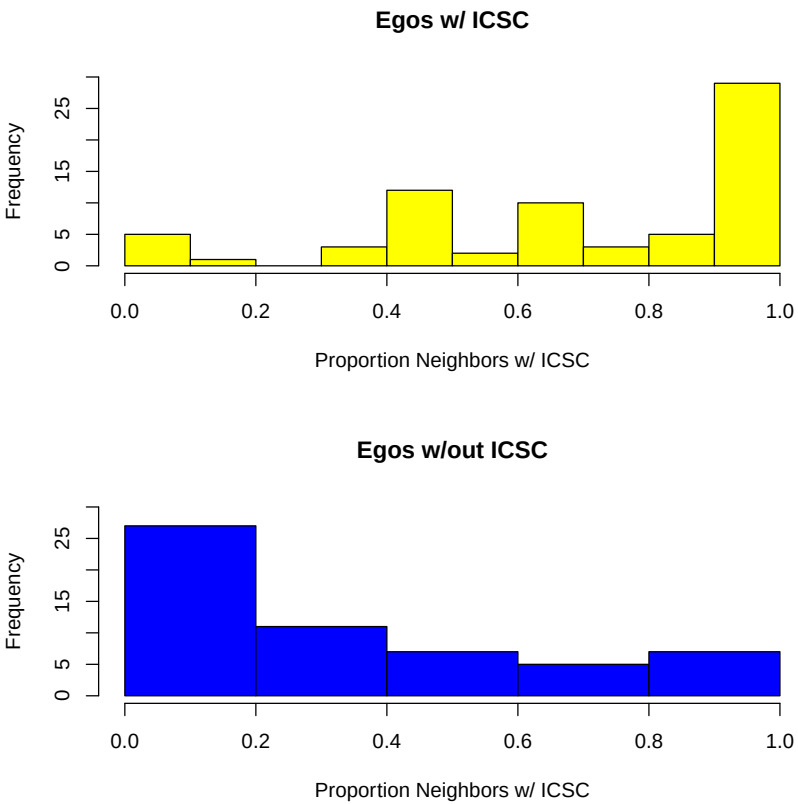
Predicting Signaling in Yeast



- Baker's yeast (i.e., *S. cerevisiae*)
- All proteins known to participate in *cell communication* and their interactions
- *Question: Is knowledge of the function of a protein's neighbors predictive of that protein's function?*

In fact . . . yes!

A Simple Approach: Nearest-Neighbor Prediction



A simple predictive algorithm uses nearest neighbor principles.

Let

$$X_i = \begin{cases} 1, \text{if corporate} \\ 0, \text{if litigation} \end{cases}$$

Compare

$$\frac{\sum_{j \in \mathcal{N}_i} x_j}{|\mathcal{N}_i|}$$

to a threshold.

Modeling Static Network-Indexed Processes

The nearest-neighbor algorithm (also sometimes called 'guilt-by-association'), although seemingly informal, can be quite competitive with more formal, model-based methods.

Nevertheless, model-based methods have certain potential advantages:

- probabilistically rigorous predictive statements;
- formal inference for model parameters; and
- natural mechanisms for handling missing data.

Modeling Static Network-Indexed Processes (cont.)

Various models have been proposed for static network-indexed processes.

Two commonly used classes/paradigms:

- Markov random field (MRF) models

  $\Longrightarrow$ Extends ideas from spatial/lattice modeling.

- Kernel-learning regression models

  $\Longrightarrow$ Key innovation is construction of graph kernels

We'll take a look at only the first – Markov random fields.

Markov Random Fields

Let $G = (V, E)$ be a graph and $\mathbf{X} = (X_1, \ldots, X_{N_v})^T$ be a collection of discrete random variables defined on V.

We say that $\mathbf{X}$ is a *Markov random field* (MRF) on G if

$$\mathbb{P}(\mathbf{X} = \mathbf{x}) > 0, \quad \text{for all possible outcomes} \quad \mathbf{x} \ ,$$

and

$$\mathbb{P}\left(X_i = x_i \,|\, \mathbf{X}_{(-i)} = \mathbf{x}_{(-i)}\right) = \mathbb{P}\left(X_i = x_i \,|\, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i}\right) \ ,$$

where

$$\mathcal{N}_i = \text{ neighborhood of } i \text{ in } G.$$

BOSTON UNIVERSITY

MRFs and Gibbs Random Fields

NOTE: Specification of conditional distributions for each X_i does *not* guarantee a well-defined joint distribution for $\mathbf{X}$.

The celebrated *Hammersley-Clifford Theorem* establishes that, under appropriate conditions, a MRF is equivalent to a *Gibbs random field,* i.e., a probability model of the form

$$\mathbb{P}(\mathbf{X} = \mathbf{x}) = \left(\frac{1}{\kappa}\right) \exp\{U(\mathbf{x})\} \quad ,$$

for *energy function* $U(\cdot)$ and *partition function*

$$\kappa = \sum_{\mathbf{x}} \exp\{U(\mathbf{x})\} \quad .$$

BOSTON UNIVERSITY

MRFs and Gibbs Random Fields (cont.)

The underlying graph G enters by facilitating a *factorization*. That is,

$$U(\mathbf{x}) = \sum_{c \in \mathcal{C}} U_c(\mathbf{x}) \ ,$$

where

- $\mathcal{C} = \{$ all cliques of all sizes in $G\}$, and
- a clique of size 1 consists of just a single vertex $v \in V$.

As a result, conditional probabilities take the form

$$\begin{aligned} \mathbb{P}\left(X_i = x_i \,|\, \mathbf{X}_{(-i)} = \mathbf{x}_{(-i)}\right) &= \frac{\mathbb{P}(\mathbf{x})}{\sum_{\mathbf{x}' : \mathbf{x}'_{(-i)} = \mathbf{x}_{(-i)}} \mathbb{P}(\mathbf{x}')} \\ &= \frac{\exp\left\{\sum_{c \in \mathcal{C}_i} U_c(\mathbf{x})\right\}}{\sum_{\mathbf{x}' : \mathbf{x}'_{(-i)} = \mathbf{x}_{(-i)}} \exp\left\{\sum_{c \in \mathcal{C}_i} U_c(\mathbf{x}')\right\}} \ . \end{aligned}$$

Illustration: Auto-logistic MRFs

So we may specify MRFs by specifying their *clique potentials* $U_c(\cdot)$.

Besag (1974) suggested a class of *auto-models*, assuming

1. only cliques $c \in \mathcal{C}$ of size one or two have non-zero potential functions U_c,
2. the conditional probabilities have an exponential family form.

In the case that the X_i are binary (i.e., 0/1 variables), it follows that

$$U(\mathbf{x}) = \sum_{i \in V} \alpha_i x_i + \sum_{\{i,j\} \in E} \beta_{ij} x_i x_j$$

and

$$\mathbb{P}\left(X_i = 1 \,|\, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i}\right) = \frac{\exp\left(\alpha_i + \sum_{j \in \mathcal{N}_i} \beta_{ij} x_j\right)}{1 + \exp\left(\alpha_i + \sum_{j \in \mathcal{N}_i} \beta_{ij} x_j\right)} \,.$$

Auto-logistic MRFs (cont.)

It is often convenient to assume that the parameters α_i and β_{ij} are *homogeneous* across the network G

Examples include

- $\alpha_i \equiv \alpha$ and $\beta_{ij} \equiv \beta$, and therefore

$$\log \frac{\mathbb{P}\left(X_i = 1 \,|\, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i}\right)}{\mathbb{P}\left(X_i = 0 \,|\, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i}\right)} = \alpha + \beta \sum_{j \in \mathcal{N}_i} x_j \ ;$$

- $\alpha_i = \alpha + |\mathcal{N}_i|\beta_2$ and $\beta_{ij} = \beta_1 - \beta_2$, and therefore

$$\log \frac{\mathbb{P}\left(X_i = 1 \,|\, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i}\right)}{\mathbb{P}\left(X_i = 0 \,|\, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i}\right)} = \alpha + \beta_1 \sum_{j \in \mathcal{N}_i} x_j + \beta_2 \sum_{j \in \mathcal{N}_i} (1 - x_j) \ .$$

BOSTON UNIVERSITY

Inference and Prediction for MRFs

Unlike nearest neighbor methods, with the MRF approach you need to

- infer model parameters

and also potentially

- predict missing values.

Inference for MRFs

In principle, to infer the parameters θ, we would use *maximum likelihood*, i.e., we would maximize

$$\log \mathbb{P}_\theta(\mathbf{X} = \mathbf{x}) = U(\mathbf{x}; \theta) - \log \kappa(\theta) \quad .$$

In practice, however, the need to evaluate $\kappa(\theta)$ makes this approach intractable in practical problems.

A common alterantive is *maximum pseudo-likelihood*, where we instead seek to optimize

$$\sum_{i \in V} \log \mathbb{P}_\theta \left(X_i = x_i \,|\, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i} \right) \quad .$$

Ignores dependencies beyond the neighborhood of each X_i and, as a result, each component is a function of only the (user-specified!) clique potentials $U_c(\mathbf{x}; \theta)$.

Inference for MRFs (cont.)

In the case of the two-parameter homogeneous auto-logistic model,

$$\left(\hat{\alpha},\hat{\beta}\right)_{MLE} = \arg\max_{\alpha,\beta}\left[\,\alpha M_1(\mathbf{x}) + \beta M_{11}(\mathbf{x}) - \kappa\left(\alpha,\beta\right)\right] \quad ,$$

but

$$\left(\hat{\alpha},\hat{\beta}\right)_{MPLE} = \arg\max_{\alpha,\beta}\left\{\alpha M_1(\mathbf{x}) + \beta M_{11}(\mathbf{x}) - \sum_{i=1}^{N_v}\log\left[1+\exp\left(\alpha+\beta\sum_{j\in\mathcal{N}_i}x_j\right)\right]\right\}$$

The second can be done using standard software for logistic regression.

Prediction with MRFs

It may be (such as in the protein function prediction problem!) that we do not observe all of $\mathbf{X}$, but rather we view as $\mathbf{X} = (\mathbf{X}^{obs}, \mathbf{X}^{miss})^T$ and we wish to predict $\mathbf{X}^{miss}$ from $\mathbf{X}^{obs} = \mathbf{x}^{obs}$, using

$$\mathbb{P}_\theta(\mathbf{X}^{miss} | \mathbf{X}^{obs} = \mathbf{x}^{obs}) \ .$$

Given estimates of θ, can do this vertex-by-vertex within a Gibbs sampler, via

$$\mathbb{P}\left(X_i \,|\, \mathbf{X}^{obs} = \mathbf{x}^{obs}, \mathbf{X}^{miss}_{(-i)} = \mathbf{x}^{(m-1),miss}_{(-i)}\right) \ ,$$

where

- $\mathbf{X}^{miss}_{(-i)}$ denotes all of $\mathbf{X}^{miss}$ except X_i, for $i \in V^{miss}$, and
- $\mathbf{x}^{(m-1),miss}_{(-i)}$ is the vector of values sampled for $\mathbf{X}^{miss}_{(-i)}$ at the $(m-1)$-th iteration.

Comparison: Autologistic MRFs versus NN

The **R** package **ngspatial** allows for fitting of MRFs like those just described, with a slightly more general parameterization, i.e.,

$$\log \frac{\mathbb{P}_{\alpha,\beta}\left(X_i = 1 \,|\, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i}, \mathbf{Z}_i = \mathbf{z}_i\right)}{\mathbb{P}_{\alpha,\beta}\left(X_i = 0 \,|\, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i}, \mathbf{Z}_i = \mathbf{z}_i\right)} = \mathbf{z}_i^T \alpha + \beta \sum_{j \in \mathcal{N}_i} (x_j - \mu_j)$$

Allows for incorporation of endogenous and exogenous effects.

On the yeast data, improves the leave-one-out prediction error to 20%, over 26% for NN.

Comparison: Autologistic MRFs versus NN (cont.)

```
> nn.ave <- sapply(V(ppi.CC.gc),
+  function(x) mean(V(ppi.CC.gc)[nei(x)]$ICSC))
> nn.pred <- as.numeric(nn.ave > 0.5)
> mean(as.numeric(nn.pred != V(ppi.CC.gc)$ICSC))
[1] 0.2598425
```

```
> library(ngspatial)
> X <- V(ppi.CC.gc)$ICSC
> A <- get.adjacency(ppi.CC.gc, sparse=FALSE)
> formula1 <- X~1
> m1.mrf <- autologistic(formula1, A=A,
+  control=list(confint="none"))
> m1.mrf$coefficients
(Intercept)         eta
  0.2004949   1.1351942
> mean(as.numeric(mrf1.pred != V(ppi.CC.gc)$ICSC))
[1] 0.2047244
```

Wrapping Up

Statisticians demand a great deal of their modeling:

1. theoretically plausible
2. estimable from data
3. computationally feasible estimation strategies
4. an ability to reflect uncertainty in estimates (e.g., confidence intervals)
5. assessment of goodness-of-fit
6. understanding of the statistical properties of the overall procedure

In network modeling, we still have a long way to go.

Wrapping Up (cont.)

One more to go . . .

- L1 Introduction, Background, and Descriptive Statistics (1.5hrs)
- L2 Network Sampling (1hr)
- L3 Network Modeling (1.5hrs)
- L4 Additional Topics in Modeling/Analysis (1.5hr)

BOSTON UNIVERSITY