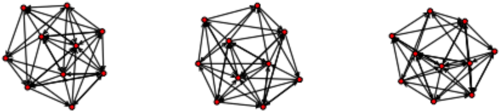
Case Studies in Dynamic Network Models: (Xu and Hero, 2013) (Sarkar and Moore, 2005)

Alex Loewi

February 25, 2014

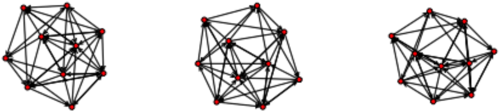
Dynamic Models (Commonalities)

- Dynamic usually means “harder”: e.g. in physics
- In networks, “dynamic” means “several data points”
- You have a “hint” when you’re fitting a static model



Dynamic Models (Commonalities)

- Dynamic usually means “harder”: e.g. in physics
- In networks, “dynamic” means “several data points”
- You have a “hint” when you’re fitting a static model



- NOTE: These models are *not temporally generative*: You can’t generate (interesting) *future* networks from them (like, for example, the preferential attachment model).

Dynamic Models (Commonalities)

- EM
- Gibbs Sampling
- (Extended) Kalman Filter
- Conjugate Gradient Ascent
- SVD, MDS, PCA
- KD-trees
- Procrustean Transform
- Spectral Clustering
- Label Switching
- ...

Static LSM Refresher

- You want locations **X**

Static LSM Refresher

- You want locations $\mathbf{X}$
- Which are embedded in a matrix of dot products $\mathbf{X}\mathbf{X}^{\mathbf{T}}$

Static LSM Refresher

- You want locations $\mathbf{X}$
- Which are embedded in a matrix of dot products $\mathbf{X}\mathbf{X}^{\mathbf{T}}$
- Which are expressible as a function of distances

$$d_{ij}^2 = x_i^T x_i + x_j^T x_j - 2x_i^T x_j$$

Static LSM Refresher

- You want locations $\mathbf{X}$
- Which are embedded in a matrix of dot products $\mathbf{XX^T}$
- Which are expressible as a function of distances
$$d_{ij}^2 = x_i^T x_i + x_j^T x_j - 2x_i^T x_j$$
- Which you can actually get from the network

Static LSM Refresher

- You want locations $\mathbf{X}$
- Which are embedded in a matrix of dot products $\mathbf{XX^T}$
- Which are expressible as a function of distances
$$d_{ij}^2 = x_i^T x_i + x_j^T x_j - 2x_i^T x_j$$
- Which you can actually get from the network
- ...

Static LSM Refresher

- You want locations $\mathbf{X}$
- Which are embedded in a matrix of dot products $\mathbf{X}\mathbf{X}^{\mathbf{T}}$
- Which are expressible as a function of distances
$$d_{ij}^2 = x_i^T x_i + x_j^T x_j - 2x_i^T x_j$$
- Which you can actually get from the network
- ...
- (Once you decide what distances you want to use.)

Static LSM Refresher

- You want locations $\mathbf{X}$
- Which are embedded in a matrix of dot products $\mathbf{X}\mathbf{X}^{\mathbf{T}}$
- Which are expressible as a function of distances
$$d_{ij}^2 = x_i^T x_i + x_j^T x_j - 2x_i^T x_j$$
- Which you can actually get from the network
- ...

$$G = f(\text{distances} = g(\text{dot products} = h(\ \textbf{latent positions!})))$$

Static LSM Refresher

- Take a random stroll (≥ 3 edges away) to determine inter-node distances $\mathbf{D}$
- Use $\mathbf{D}$ to minimize a set of distances expressed as dot products $\min_X |\mathbf{D} - \mathbf{X}\mathbf{X}^\mathbf{T}|_F$
- Extract the positions.

$$G = f(\text{distances} = g(\text{dot products} = h(\textbf{ latent positions!})))$$

Static LSM Refresher

- The likelihood is not convex, so you have to have a good first guess if you're going to maximize iteratively (or you'll get stuck on the way).
- For the first step, find a linear solution (MDS/PCA), then iterate on that. Hoff et. al use Metropolis-Hastings, Sarkar and Moore use conjugate gradient descent.
- PROBLEM is, the likelihood is a function of DISTANCES, and you want absolute POSITIONS. So, the correct solution upside down has the same likelihood as the true solution.
- SOLUTION: Choose a position, always rotate back as close to it as possible.

Dynamic LSMs

The biggest difference is the addition of X_{t-1} in the conditioning set

$$X_t = \arg\max_X p(X|G_t, X_{t-1}) = \arg\max_X p(G_t|X)p(X|X_{t-1})$$

The most "dynamic" part of this model is the "don't move much" rule

$$\log p(X_t|X_{t-1}) = -\sum_{i=1}^{n} |X_{i,t} - X_{i,t-1}|^2 / 2\sigma^2 + const$$

The more the new positions are like the old ones, the greater the likelihood

A Neat Trick, Once You Recognize It

(Sarkar and Moore, 2013):

$$\arg\min_{X} |D - XX^T|_F$$

(Hoff, Raftery, and Handcock, 2002):

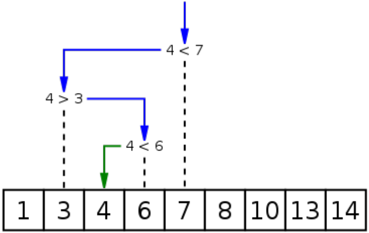
$$\arg\min_{TZ} tr(Z_0 - TZ)^T(Z_0 - TZ)$$

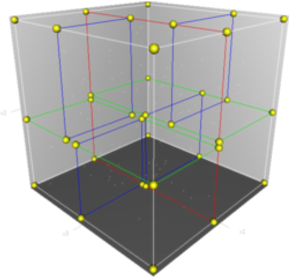
Why would you think of this, but –

$$tr(A^T A) = tr\begin{pmatrix} a^2 + c^2 & ab + cd \\ ab + cd & b^2 + d^2 \end{pmatrix} = \sum_{ij} a_{ij}^2 = |A|_F$$

KD-Trees: I Thought This Was Pretty Cool

Because of the range-restricted ties, the likelihood can be calculated in local pieces.





KD-trees allow you to efficiently find only the relevant pieces.

Models for Growing Networks

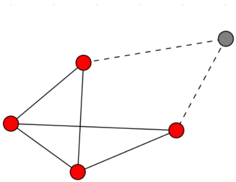
- Latent Space Models have a logic
- This logic should be no less valid for as-yet-unobserved nodes
- Using this logic helps helps reduce uncertainty about previously observed nodes
- If a new node N makes friends with a person P, but not their friend F, then F and N should be placed as far apart as possible.

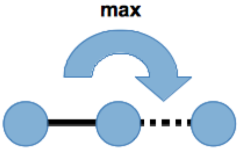
Models for Growing Networks

- Latent Space Models have a logic
- This logic should be no less valid for as-yet-unobserved nodes
- Using this logic helps helps reduce uncertainty about previously observed nodes
- If a new node N makes friends with a person P, but not their friend F, then F and N should be placed as far apart as possible.
-
$$\min_X |\mathbf{D} - \mathbf{X}\mathbf{X}^{\mathbf{T}}|_F - \lambda |X_{new} - X_{\notin N(new)}|_F$$

Models for Growing Networks

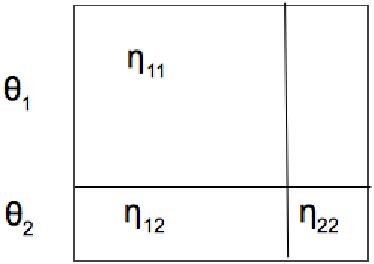
- If a new node N makes friends with a person P, but not their friend F, then F and N should be placed as far apart as possible.
- $$\min_{X} |\mathbf{D} - \mathbf{X}\mathbf{X}^{\mathbf{T}}|_F - \lambda |X_{new} - X_{\notin N(new)}|_F$$





Static SBM Refresher

- There are "blocks" of uniform densities
- There are two versions of the problem: with (a priori) and without (a posteriori) the class labels.
- If you know who's in what class, this is just estimating a Bernoulli.



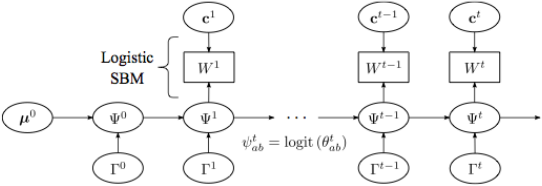
Static SBM Refresher

- A posteriori is the more interesting problem, but you have to check ALL 2^n possible labelings.
- This is not possible to do for more than a few nodes, but sampling approaches can be used.

$$
\begin{aligned}
&P(Y;\theta,\eta)\\
&= \sum_{x\in[1,2]^n} P(Y,X;\theta,\eta)\\
&= \sum_{x\in[1,2]^n} (1-\theta)^n \left(\frac{\theta}{1-\theta}\right)^{n_2} \tilde{\eta}_{11}^{\binom{n_1}{2}} \tilde{\eta}_{12}^{n_1 n_2} \tilde{\eta}_{22}^{\binom{n_2}{2}} \left(\frac{\eta_{11}}{\tilde{\eta}_{11}}\right)^{e_1 1} \left(\frac{\eta_{12}}{\tilde{\eta}_{12}}\right)^{e_1 2} \left(\frac{\eta_{22}}{\tilde{\eta}_{22}}\right)^{e}\\
&= \sum \theta^{n_1}(1-\theta)^{n_2} \prod_{b\in block} \eta_b^{e_b}(1-\eta_b)^{\binom{n_b}{e_b}-e_b}
\end{aligned}
$$

Dynamic SBMs

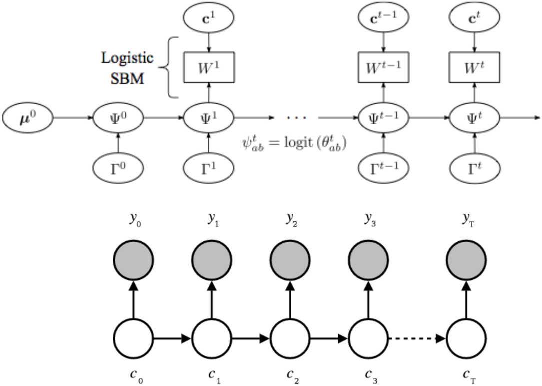
The dynamic part:



$$f(\psi^t | W^{(t)}) \propto f(W^{(t)} | \psi^t, W^{(t-1)}) f(\psi^t | W^{(t-1)})$$

Dynamic SBMs

The depth of my understanding: Kalman Filters are ways to solve complicated HMMs.



You don't *force* the class labels to only change a little, but if you guess that they only change a little, then you tend to get a pretty good fit. The previous parameter acts as a "hint."

Thank you!

aloewi@andrew.cmu.edu
Hamburg Hall, 3005