Maintaining Nets and Net Trees under Incremental Motion

Minkyoung Cho, David Mount, and Eunhui Park

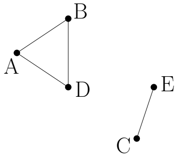
Department of Computer Science
University of Maryland, College Park

August 25, 2009

Latent Space Embedding (LSE)

- The probability of relational ties in a network may depend on similarity of characteristics of individuals.
- Subsets of individuals with many ties may indicate that the group is nearby in some social space of characteristics.
- Imagine that each actor i is associated with a position z_i in social space.
- This social space consists of unobserved latent characteristics, which influence the existence of ties in the network.
- LSE Estimation: Given a social network Y, estimate these positions Z.

	A	B	C	D	E
A	-	1	0	1	0
B	1	-	0	1	0
C	0	0	-	0	1
D	1	1	0	-	0
E	0	0	1	0	-



Latent Space

Latent Space Embedding (LSE)

Usefulness of LSE

- Provides a parsimonious model of network structure ($O(dn)$ rather than $O(n^2)$)
- Allows for natural interpretation of relational concepts, such as "betweenness," "surroundedness," and "flatness"
- Provides a means to perform visual analysis of network structure through spatial relationships (when dimension is low)
- The model is flexible and extensible.

Talk Overview

- LSE model and estimation
- WSPDs and static cost computation
- Efficient incremental cost computation
- Nets and net trees
- Incremental motion model
- Maintaining nets for moving points
- Concluding remarks

LSE — Stochastic Model [HRH02]

Input

- Y, an $n \times n$ sociomatrix ($y_{i,j} = 1$ if there is a tie between i and j)
- Additional covariate information X (ignored here)

Model Parameters

- Z: the positions of n individuals, $\{z_1, \ldots, z_n\}$
- α: real-valued scaling parameter

Stochastic Model

Ties are independent of each other, but depend on Z and α.

$$\Pr[Y \mid Z, \alpha] = \prod_{i \neq j} \Pr[y_{i,j} \mid z_i, z_j, \alpha]$$

LSE — Estimation

Objective

Given an $n \times n$ matrix Y, determine Z and α to maximize $\Pr[Y \mid Z, \alpha]$.

MCMC — Metropolis Hastings Algorithm

- An iterative algorithm for drawing a sequence of samples $Z_0, Z_1, Z_2, \ldots$ from a distribution [MRR+53]
- Simplified View: For $k = 0, 1, 2, \ldots$
  - Sample a proposal Z from some distribution $J(Z \mid Z_k)$
  - Evaluate the decision variable

$$\rho = \frac{\Pr[Y \mid Z, \alpha_k]}{\Pr[Y \mid Z_k, \alpha_k]}$$

  - Accept Z as Z_{k+1} with probability $\min(1, \rho)$
- Convergence may require many iterations. Efficiency is critical.

LSE — Cost Computation

Metropolis-Hastings Main Loop

- Perturb point positions $\rightarrow Z$
- Evaluate decision variable $\rightarrow \rho(Y, Z, \alpha)$ ($\leftarrow$ bottleneck)
- Accept/Reject

Evaluating the Decision Variable

Introduce a parameterization $\eta(Z, \alpha)$ [HRH02]:

$$\begin{aligned}
\eta_{i,j} &= \log \mathrm{odds}(y_{i,j} = 1 \mid z_i, z_j, \alpha) \\
&= \alpha - \mathrm{dist}(z_i, z_j) \\
\log \Pr[Y \mid \eta] &= \sum_{i \neq j} (\eta_{i,j} y_{i,j} - \log(1 + e^{\eta_{i,j}}))
\end{aligned}$$

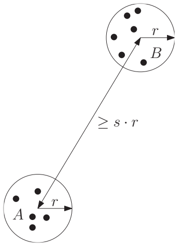
Computing this log-likelihood naively requires $O(n^2)$ time.

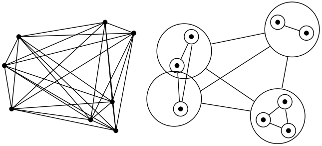
Talk Overview

Well-Separated Pair Decomposition (WSPD)

Well-Separated Pair Decomposition

- n points determine $O(n^2)$ pairs
- A and B are s-well separated if they can be enclosed in balls of radius r that are separated by at least $s \cdot r$
- A WSPD of a point set P is a collection of well-separated pairs (A_i, B_i) covering all pairs of the set
- An n-element point set in dimension d has a WSPD of size $O(s^d n) = O(n)$ [CaK95]
- Distances approximated to a relative error of $(1 + \frac{1}{s})$.





28 pairs

12 well separated pairs (separation factor = 1)

Computing Costs (Statically)

Approximating the log-likelihood

$$
\begin{aligned}
\Pr[Y \mid \eta] &= \sum_{i \neq j} \{\eta_{i,j} y_{i,j} - \log\left(1 + e^{\eta_{i,j}}\right)\} \\
&\approx \sum_{i \neq j} \left\{\eta_{i,j} y_{i,j} - \left(\log 2 + \frac{\eta_{i,j}}{2} + \frac{\eta_{i,j}^2}{8}\right)\right\} \\
&\approx \sum_{(\mathrm{A,B}) \in \mathrm{WSPD}} \left\{\eta_{\mathrm{A,B}} y_{\mathrm{A,B}} - \left(\log 2 + \frac{\eta_{\mathrm{A,B}}}{2} + \frac{\eta_{\mathrm{A,B}}^2}{8}\right)\right\}
\end{aligned}
$$

- Precompute $\eta_{\mathrm{A,B}}$, $\eta_{\mathrm{A,B}}^2$ and $y_{\mathrm{A,B}}$ in $O(n(\log n + s^d))$ time (Good!)
- After perturbation, need to rebuild spatial index and WSPD (Bad!)

Talk Overview

- LSE model and estimation
- WSPDs and static cost computation
- Efficient incremental cost computation
- Nets and net trees
- Incremental motion model
- Maintaining nets for moving points
- Concluding remarks

Computing Costs (Incrementally)

Incremental Hypothesis

If point perturbations are small, then relatively few changes to WSPD structure.

Incremental Approach

(After each perturbation):

- Update spatial index ($\longleftarrow$ this talk)
- Update WSPD structure
- Update decision variable

Nets

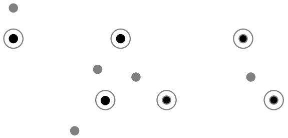
Net

P is a finite set of points in a $\mathbb{R}^d$. Given $r > 0$, an r-net for P is a subset $X \subseteq P$ such that,

$$\max_{p \in M} dist(p, X) < r \quad \text{and}$$
$$\min_{\substack{x, x' \in X \\ x \neq x'}} \mathrm{dist}(x, x') \geq r.$$

Features

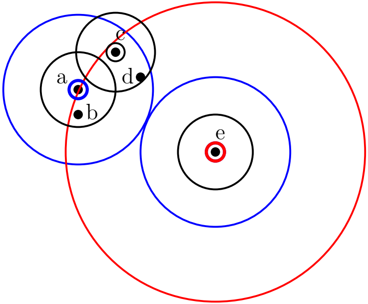
- Intrinsic: Independent of coord. frame
- Stable: Relatively insensitive to small point motions

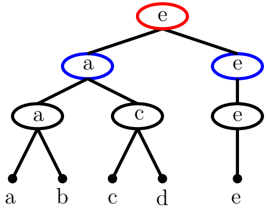


Net Tree

Net Tree

- The leaves of the tree consists of the points of P.
- The tree is based on a series of nets, $P^{(1)}, P^{(2)}, \ldots, P^{(h)}$, where $P^{(i)}$ is a (2^i)-net for $P^{(i-1)}$.
- Each node on level $i-1$ is associated with a parent, at level i, which lies lies within distance 2^i.





Talk Overview

- LSE model and estimation
- WSPDs and static cost computation
- Efficient incremental cost computation
- Nets and net trees
- Incremental motion model
- Maintaining nets for moving points
- Concluding remarks

Incremental Motion — Observer-Builder Model

Incremental (Black-Box) Motion

- Motion occurs in discrete time steps
- All points may move
- No constraints on motion, but processing is most efficient when motion is small or predictable

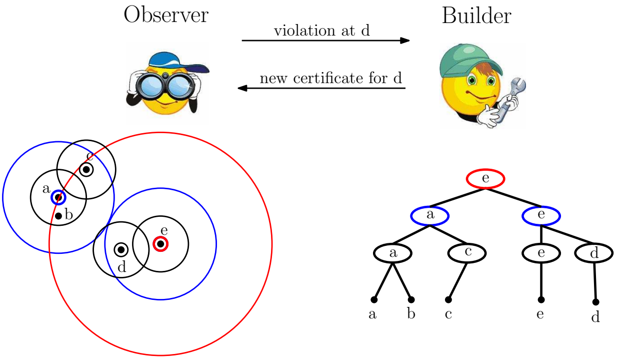
Observer-Builder Model

- Two agents cooperate to maintain data structure [MNP+04,YiZ09]
  - Observer: Observes points motions
  - Builder: Maintains the data structure
- Certificates: Boolean conditions, which prove structure's correctness

Incremental Model — Observer-Builder Model

Communication Protocol

- Builder maintains structure and issues certificates
- Observer notifies builder of any certificate violations
- Builder then fixes the structure and updates certificates



Observer-Builder — Cost Model

Cost Model

- Computational cost is the total communication complexity (e.g., number of bits) between the observer and builder.
- Builder's goal: Issue certificates that will be stable against future motion.
- Builder's and observer's overheads are not counted:
  - Builder's overhead: Is small.
  - Observer's overhead: Observer can exploit knowledge about point motions to avoid re-evaluating certificates.

Talk Overview

- LSE model and estimation
- WSPDs and static cost computation
- Efficient incremental cost computation
- Nets and net trees
- Incremental motion model
- Maintaining nets for moving points
- Concluding remarks

Incremental Online Algorithm for Maintaining an r-Net

What the Builder Maintains

- The point set, P
- The r-net, X
- For each $p \in P$:
  - A representative $\mathrm{rep}(p) \in X$, where $\mathrm{dist}(p, x) \leq r$
  - A candidate list $\mathrm{cand}(p) \subseteq X$ of possible representatives for p

Certificates

- For $p \in P$, Assignment Certificate(p): $\mathrm{dist}(p, \mathrm{rep}(p)) \leq r$ (representative is close enough)
- For $x \in X$, Packing Certificate(x): $|b(x, r) \cap X| \leq 1$ (no other net-point is too close)

Incremental Online Algorithm for Maintaining an r-Net

Assignment Certificate Violation(p)

Point p has moved beyond distance r from its representative:

- If cand(p) has a representative x within distance r, x is now p's new representative.
- Otherwise, make p a net point (add it to X) and add p to candidate lists of points within distance r of p

Packing Certificate Violation(x)

There exists another net point within distance r of x:

- Remove all net points within radius r of x. (This may induce many assignment violations)
- Handle all assign certificate violations

Competitive Ratio

Competitive Ratio

- We establish the efficiency through a competitive analysis
- Given an incremental algorithm A and motion sequence $\mathcal{P}$, define

$$\begin{aligned} C_A(P) &= \text{Total communication cost of running } A \text{ on } \mathcal{P} \\ C_{OPT}(P) &= \text{Total communication cost of optimal algorithm on } \mathcal{P} \end{aligned}$$

  The optimal algorithm may have full knowledge of future motion

- Competitive Ratio:

$$\max_{\mathcal{P}} \frac{C_A(P)}{C_{OPT}(P)}$$

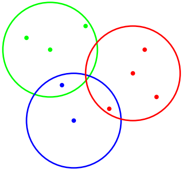
Slack Net

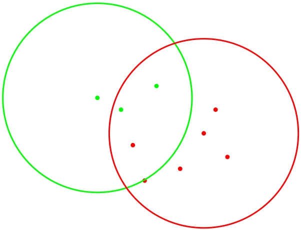
Slack Net

- To obtain a competitve ratio, we relaxed the r-net definition slightly.
- Given constants $\alpha, \beta \geq 1$, an (α, β)-slack r-net is a subset $X \subseteq P$ of points such that

$$\max_{p \in M} dist(p, X) < \alpha r \qquad \text{and} \qquad \forall x \in X, \; |\{X \cap b(x, r)\}| \leq \beta.$$

Covering radius larger by factor α. Allow up to β net points to violate packing certificate.





Our Results

Theorem: (Slack-Net Maintenance)

There exists an incremental online algorithm, which for any real $r > 0$, maintains a $(2, \beta)$-slack r-net for any point set P under incremental motion. Under the assumption that P is a $(2, \beta)$-slack $(r/2)$-net, the algorithm achieves a competitive ratio of $O(1)$.

Theorem: (Slack-Net Tree Maintenance)

There exists an online algorithm, which maintains a $(4, \beta)$-slack net tree for any point set P under incremental motion. The algorithm achieves a competitive ratio of at most $O(h^2)$, where h is the height of the tree.

Concluding Remarks

Summary

- LSE is a flexible and powerful method for producing a geometric point model for a given social network
- It estimates point positions in an unobserved social space based on a stochastic model relating network ties to distances
- Introduced a computational model for incremental motion.
- Showed how to improve efficiency of LSE computations based on MCMC approaches through the use of WSPDs (statically) and an online incremental algorithm (dynamically).

Concluding remarks

- Tighten competitive ratio bounds (or show they are tight)
- Show how to use this to update WSPDs and MCMC cost functions.
- Implementation and testing on real data sets

Thank you!

Bibliography

- [CK95] P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. Assoc. Comput. Mach.*, 42:67–90, 1995.
- [HRH02] P. D. Hoff, A. E. Raftery, and M. S Handcock. Latent space approaches to social network analysis. *J. American Statistical Assoc.*, 97:1090–1098, 2002.
- [MNP+04] D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu. A computational framework for incremental motion. In *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, pages 200–209, 2004.

Bibliography

- [MRR+53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21:1087–1092, 1953.
- [YZ09] K. Yi and Q. Zhang. Multi-dimensional online tracking. In *Proc. 20th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 1098–1107, 2009.