# Efficient Algorithms for Latent Space Embedding

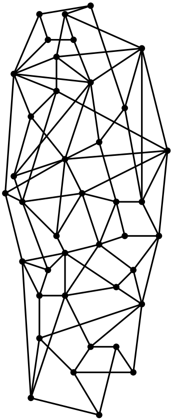
Minkyoung Cho, David Mount, and Eunhui Park

Department of Computer Science
University of Maryland, College Park

MURI Meeting – Nov 12, 2010

# Motivation

- Social networks exhibit various structural features:
  - Transitivity
  - Homophily on attributes
  - Clustering
- Analysis of social networks seeks to uncover deeper structure, as evidenced by network ties.
- The likelihood of a tie is often correlated with the similarity of attributes of the actors. (E.g., geography, age, ethnicity, income).
- Attributes may be observed or unobserved (latent).
- Motivating Question: Through analysis of network structure, can we recover an understanding of these, possibly hidden, attributes?



# Latent Space Embedding (LSE)

## Hypothesis

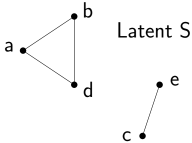
The likelihood of relational ties in social networks depends on the similarity of attributes in an unobserved latent space.

## Problem Statement

Given a network $Y = [y_{i,j}]$ with $n$ nodes, estimate a set of positions $Z = \{z_1, \ldots, z_n\}$ in $\mathbb{R}^d$ that best describes this network relative to some model.

Network

| | 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 $O(n^2)$ size)
- Allows for natural interpretation of geometric relations, such as "betweenness," "surroundedness," and "dimensionality"
- Can be used for cluster analysis of nodes
- Provides a means to perform visual analysis of network structure through spatial relationships (when dimension is low), and outlier detection.

# LSE — Stochastic Model

## Input

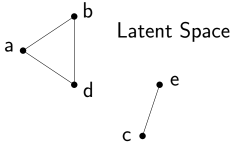
- $Y$: An $n \times n$ sociomatrix ($y_{i,j} = 1$ if there is a tie between $i$ and $j$)

## Model Parameters

- $Z$: The positions of $n$ individuals, $\{z_1, \ldots, z_n\}$ in latent space
- $\alpha$: Real-valued scaling parameter

Network

| | 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 | - |



# LSE — Stochastic Model

## Logistic Regression Model [HRH02]

Hypotheses: Ties are statistically independent, and the odds of a tie decreases exponentially with attribute distance.

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

$$\log \text{odds}(y_{i,j} = 1 \mid z_i, z_j, \alpha) \;=\; \alpha - \|z_i - z_j\|.$$

Defining $\eta_{i,j} = \alpha - \|z_i - z_j\|$, we have

$$\log \Pr[Y \mid \eta] \;=\; \sum_{i \neq j} (\eta_{i,j} y_{i,j} - \log(1 + e^{\eta_{i,j}})).$$

# LSE — Stochastic Model

## LSE Model

Let $\eta_{i,j} = \alpha - \|z_i - z_j\|$.

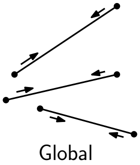
$$\log \Pr[Y \mid \alpha, \eta] = \sum_{i \neq j} (\eta_{i,j} y_{i,j} - \log(1 + e^{\eta_{i,j}})).$$

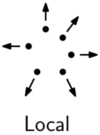
Global Component:

$\sum_{i \neq j} \eta_{i,j} y_{i,j} \Rightarrow$ Avoid long edges

Local Component:

$-\sum_{i \neq j} \log(1 + e^{\eta_{i,j}}) \Rightarrow$ Encourage dispersion



# LSE — MCMC Algorithm

## Markov-Chain Monte-Carlo (MCMC)

- For $k = 0, 1, 2, \ldots$
  - Perturbation: Sample a random perturbation $Z_*$ of $Z_k$.
  - Evaluation: Compute the decision variable

$$\rho = \frac{\Pr[Y \mid Z_*, \alpha]}{\Pr[Y \mid Z_k, \alpha]} \qquad \leftarrow \text{(Computational bottleneck)}$$

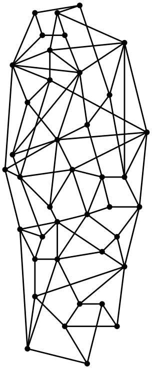
  - Decision: Accept $Z_*$ as $Z_{k+1}$ with probability $\min(1, \rho)$

Convergence requires many iterations (tens of thousands and more).

Existing computational approaches, based on brute-force evaluation of probabilities, are unacceptably slow and do not scale to large networks.

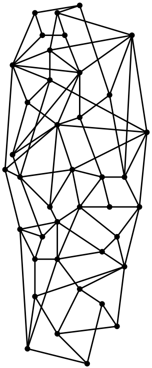
# LSE — Efficient LSE Computations

- Naive (exact) computation for each iteration requires quadratic time.
- Computation involves retrieval of spatial relations and distances.
- Need efficient geometric retrieval data structures.
- Important features:
  - Approximate: Exact structures are too slow.
  - Incremental: MCMC algorithms involve repeated perturbation of point positions.
  - Adaptable: Queries are highly non-uniform, and structures should adapt to these patterns.
  - Variational-Sensitive: Approximations must preserve small relational variations.



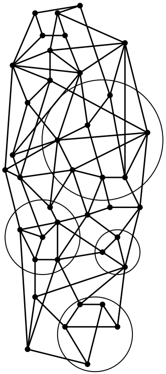
# LSE — Efficient LSE Computations

- Naive (exact) computation for each iteration requires quadratic time.
- Computation involves retrieval of spatial relations and distances.
- Need efficient geometric retrieval data structures.
- Important features:
  - Approximate: Exact structures are too slow.
  - Incremental: MCMC algorithms involve repeated perturbation of point positions.
  - Adaptable: Queries are highly non-uniform, and structures should adapt to these patterns.
  - Variational-Sensitive: Approximations must preserve small relational variations.



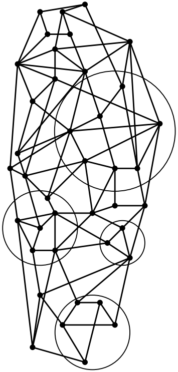
# LSE — Computational Challenges

- Almost all prior work on geometric data structures has focused on fairly static structures.
- Geometric MCMC is extremely dynamic:
  - Classical dynamics: Point insertion, point deletion.
  - Incremental dynamics: Many points change positions (but motion is small).
  - Block dynamics: Groups of points move in unison.
- Incremental dynamics, adaptivity, variational-sensitivity are unstudied in computational geometry.
- Such nimble data structures will be broadly applicable to a wide variety of settings.



# LSE — Computational Challenges

- Almost all prior work on geometric data structures has focused on fairly static structures.
- Geometric MCMC is extremely dynamic:
    - Classical dynamics: Point insertion, point deletion.
    - Incremental dynamics: Many points change positions (but motion is small).
    - Block dynamics: Groups of points move in unison.
- Incremental dynamics, adaptivity, variational-sensitivity are unstudied in computational geometry.
- Such nimble data structures will be broadly applicable to a wide variety of settings.



# LSE — Efficient LSE Computations

This MURI grant has enabled significant advances on these issues.

| High Priority | Prior Art | Current Art |
|---|---|---|
| Incremental | None | Kinetic net tree [ISAAC'09] |
| Dynamic | Limited (no range updates) | Dynamics + range search [SoCG'10] |
| Variation-Sensitivity | None | For block dynamics |
| **Middle Priority** | **Prior Art** | **Current Art** |
| Adaptable | None | Self-adjusting quadtree |
| Space/Time Tradeoffs | Suboptimal | Optimal at extremes [JACM'09], [SoCG'10] (submission to STOC'11) |
| **Desirable** | **Prior Art** | **Current Art** |
| Compressed | None | [AlgoSensor'09] [ESA'10] |
| Kinetic | Yes | Greater flexibility [CGTA'09] |

# LSE Algorithm Engineering

Objective: Achieve $\geq 90\%$ reduction in running time with $\leq 1\%$ error.

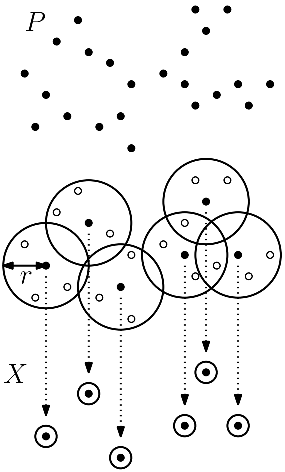
- Local Component:
  - Tapering: Keep all the “heavy hitters”
  - Sampling: Sample the rest
- Global Component:
  - Sparse networks: If sparser than local sampling rate, use brute force.
  - Dense networks: Apply dynamic net trees with block dynamics.

# Computational Tools – 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,

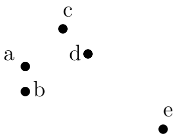
$$\max_{p \in M} dist(p, X) < r \quad \text{and}$$
$$\min_{\substack{x, x' \in X \\ x \neq x'}} \|x - x'\| \geq r.$$

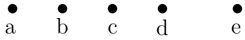


# Net Trees

## 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 $2^i$.

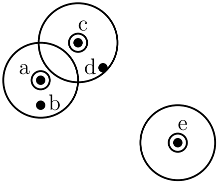


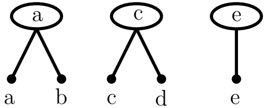



# Net Trees

## 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 $2^i$.

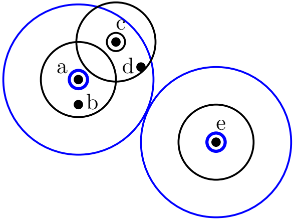


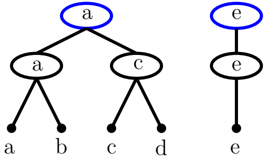



# Net Trees

## 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 $2^i$.

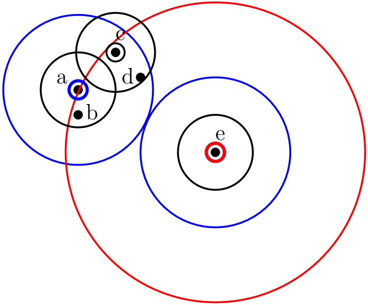


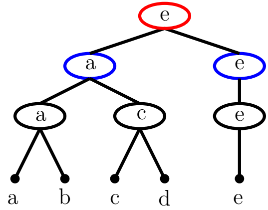



# Net Trees

## 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 $2^i$.

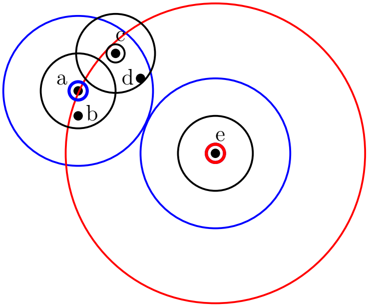


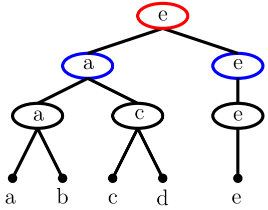



# Net Trees - Results

## Net Tree

- Presented algorithms for net trees under incremental motion.
- Proved bounds on the competitive ratio of our algorithm, relative to the optimal algorithm.
- Demonstrated that net trees can be applied to block dynamics in LSE computations.



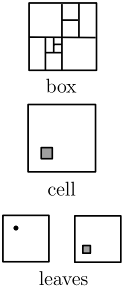



# Background: BD-tree

**Box Decomposition tree (BD-tree)**

A geometric data structure based on a hierarchical decompostion of space into $d$-dimensional axis-aligned rectangles.

- Each node is associated with a region of space, cell.
- Each cell has an outer box and optional inner box.
- Partition operations: split and shrink.
- Internal nodes: split nodes and shrink nodes.
- A leaf has a single point or a single inner box.



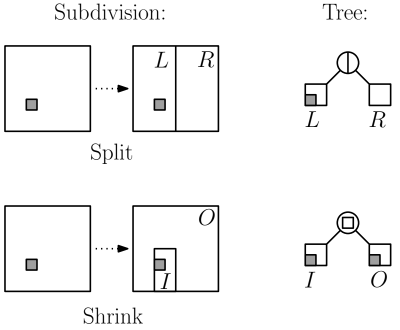
# BD-tree: Partitioning Operations

## Split

A split partitions a cell by an axis-orthogonal hyperplane that bisects the cell's longest side.

## Shrink

A shrink partitions a cell by a shrinking box, which lies within the cell.



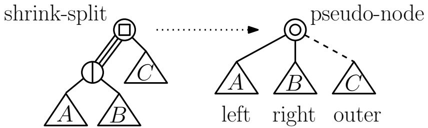
# Pseudo-nodes

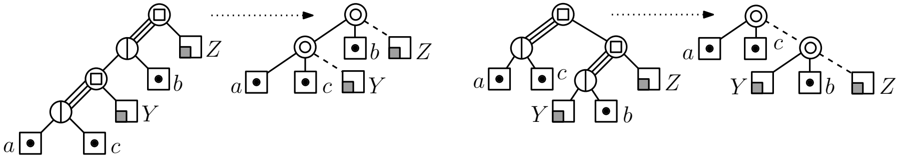
### Shrink-split property

The inner child of each shrink node is a split node.

### Pseudo-node

Merged shrink-split pair into a single node

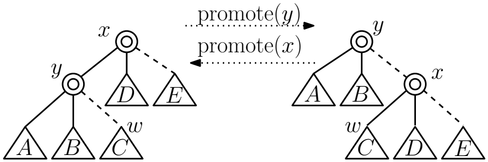


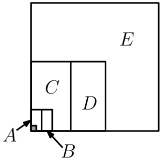




Examples of transformation to pseudo-nodes

# Promotion: Rotation on pseudo-nodes

- BD-trees can be rebalanced through rotation, called promotion.
- Promotion does not alter subdivision, just the tree structure.




## Results

- Developed a randomized balanced quad tree structure, quadtreap.
- Supports efficient insertion, deletion, approximate proximity queries.
- We are developing a self-adjusting variant of this structure, generalizing the 1-dimensional splay tree.

# Future/Further Work

We have significantly advanced the state of the art of dynamic geometric data structures, but many questions remain:

- Intrinsic (net-tree like) variants of the quadtreap and self-adjusting quadtree?
- Establishing variational-sensitivity for incremental dynamics?
- Efficient MCMC updates for networks of moderate density.
- Continued prototyping of algorithms, analysis, and subsequent dissemination of software.

Closing the circle: Apply the tools and algorithms we have developed for the analysis of actual networks.

# Some Work Supported by this Grant

- Storing and Retrieving Information from Dynamic Data Sets:
  - Maintaining Nets and Net Trees under Incremental Motion (with M. Cho and E. Park), ISAAC'09.
  - A Dynamic Data Structure for Approximate Range Searching (with E. Park), SoCG 2010.
- Efficient Algorithms and Data Structures for Geometric Retrieval:
  - Space-Time Tradeoffs for Approximate Nearest Neighbor Searching (with S. Arya and T. Malamatos), JACM'09.
  - Tight Lower Bounds for Halfspace Range Searching (with S. Arya and J. Xia), SoCG 2010 (invited to a special issue of DCG).
  - A Unifying Framework for Approximate Proximity Searching (with S. Arya and G. Fonseca), ESA 2010.
  - Approximate Polytope Membership Queries (with S. Arya and G. Fonseca), (submitted to STOC 2011).
- Compression and Retrieval of Kinetic Data:
  - Compressing Kinetic Data From Sensor Networks (with S. Friedler), AlgoSensors'09.
  - Spatio-Temporal Range Searching Over Compressed Sensor Data (with S. Friedler), ESA 2010.

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.
- [HRT07] M. S. Handcock and A. E. Raftery and J. M. Tantrum. Model-based clustering for social networks. *J. R. Statist. Soc. A*, 170, Part 2, 301–354, 2007.