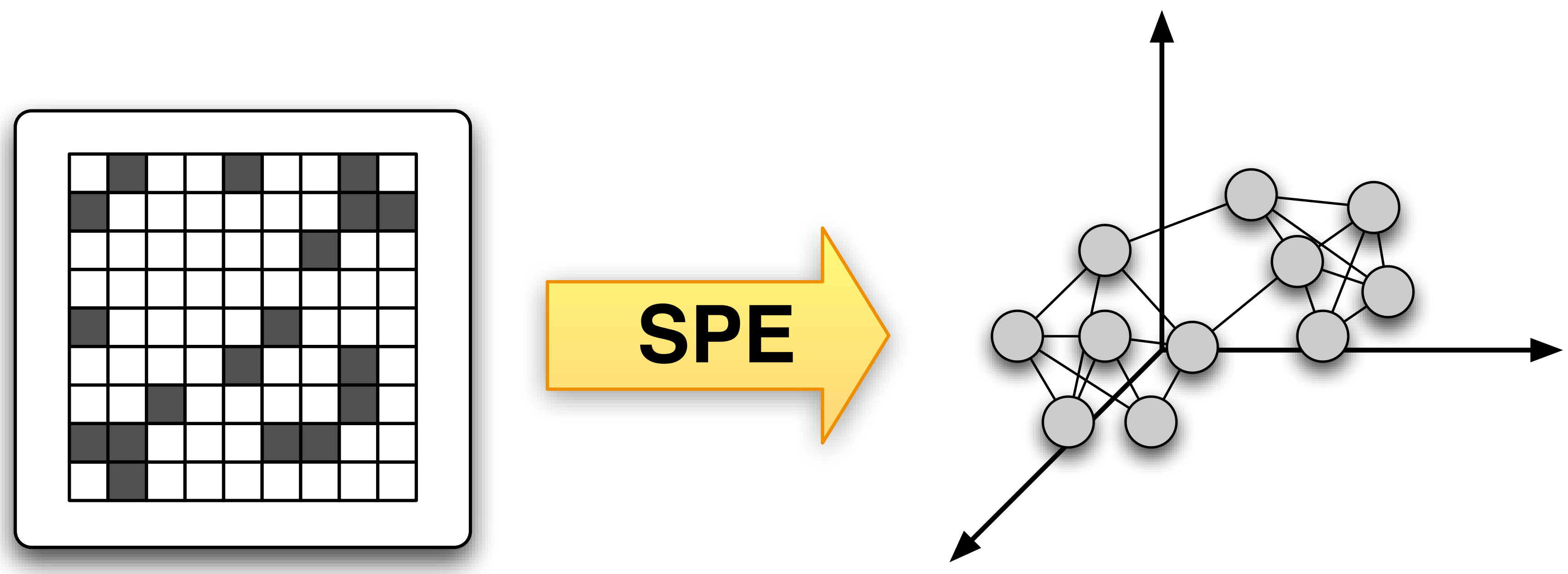
Structure Preserving Embedding

Blake Shaw and Tony Jebara
Columbia University
6/15/2009

Introduction

SPE is a graph embedding algorithm



Input: binary adjacency matrix

$A \in \mathbb{B}^{N \times N}$

Output: point coordinates for each node

$\vec{y}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$

Graph Embedding

Applications

- Many different objectives for graph embedding [Chung '97][Battista et al. '99]
    - Planarity - drawing graphs such that edges never cross, possible for some graphs in 2D, possible for all in 3D
    - Approximating NP-hard sparsest cut problem [Arora '04]
- Our focus: visualization and compression
    - Real-world data from observing binary interactions, e.g. links between websites, and synthetic data such as interesting classical graphs

Graph Embedding

Background

- **Spring embedding** - simulate physical system where edges are springs, use Hooke's law to compute forces, converges to local optimium
- **Spectral embedding** - decompose adjacency matrix A with an SVD and use eigenvectors with highest eigenvalues for coordinates
- **Laplacian eigenmaps** [Belkin, Niyogi '02] - form graph laplacian from adjacency matrix, $L = D - A$, apply SVD to L and use eigenvectors with smallest non-zero eigenvalues for coordinates

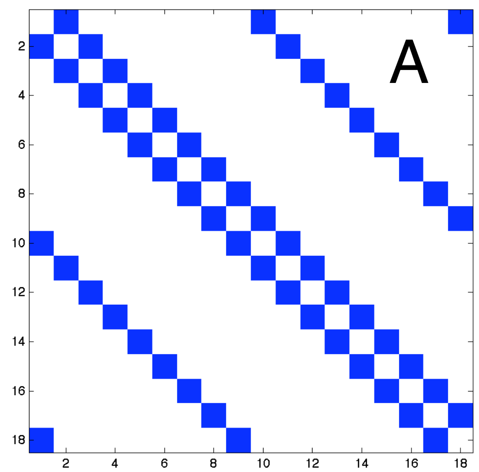
Structure Preserving Embedding

SDP & SVD

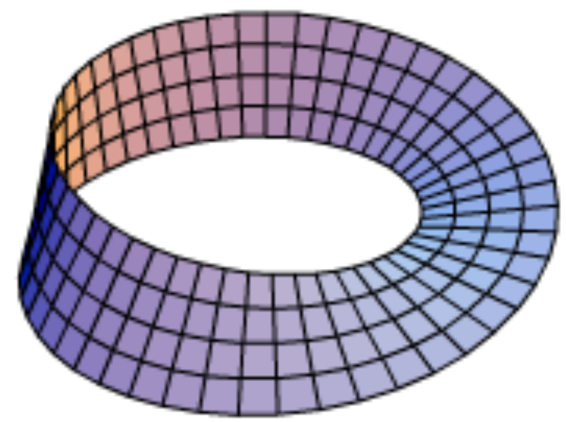
1. SDP to learn an embedding K from A
   - Linear constraints on K preserve the global topology of the input graph
   - Convex objective favors low-rank K close to the spectral solution, ensuring low-dimensional embedding
2. Use eigenvectors of K with largest eigenvalues as coordinates for each node

- Similar to MVU and MVE
  [Weinberger et al. '05] [Shaw, Jebara '07]

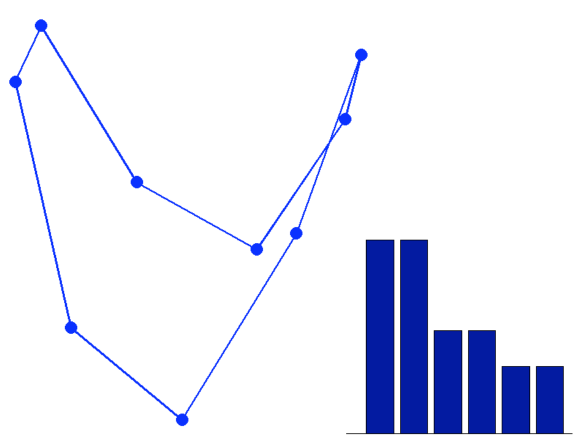
Möbius Ladder Graph



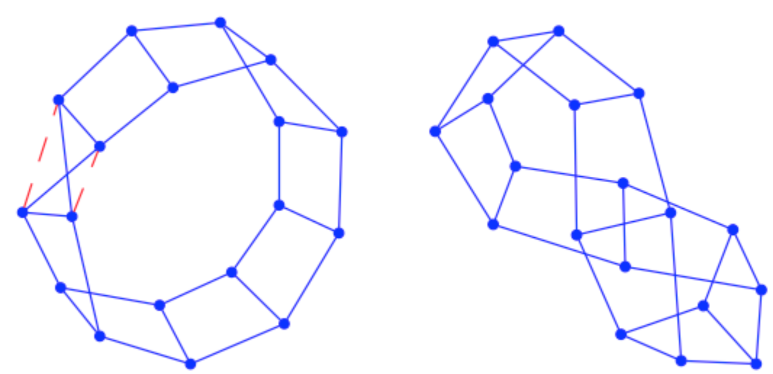
Möbius Ladder Graph



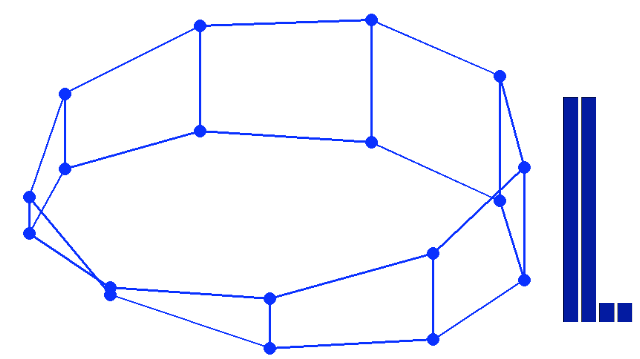
Möbius Band



Spectral Embedding



Two Spring Embeddings



SPE Embedding

Outline

- Introduction
  - Applications, background, SDP + SVD, Möbius example
- Structure Preserving Embedding
  - A low-rank objective
  - Graph topology from linear constraints
  - Algorithm details, implementation
- Experiments
  - Classical graphs, molecules, political blogs
- Dimensionality Reduction
- Review

Low-Rank Objective

- Low-rank K corresponds to low-dimensional embedding

SPE objective:

$$\max_{K \in \mathcal{K}} \ \mathrm{tr}(KA)$$

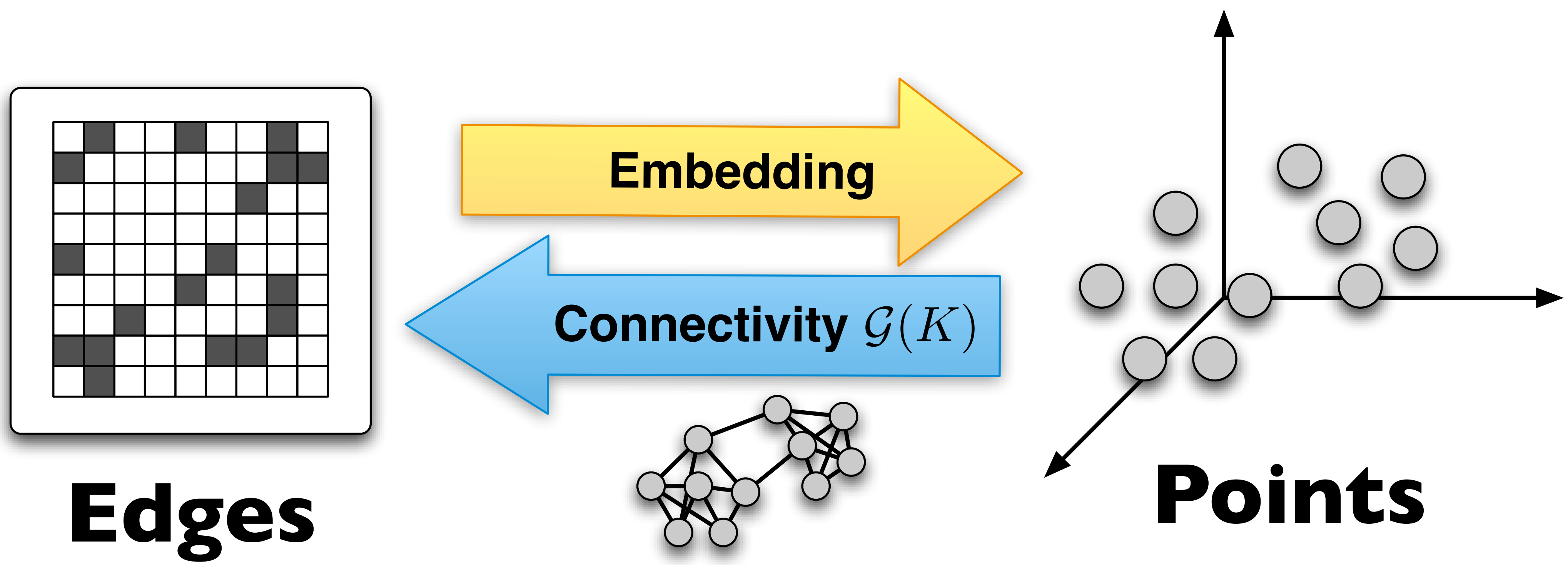
$$\mathcal{K} = \{K \succeq 0, \mathrm{tr}(K) \leq 1, \sum_{ij} K_{ij} = 0\}$$

Lemma 1. *The objective function* $\max_{K \succeq 0} \mathrm{tr}(KA)$ *subject to* $\mathrm{tr}(K) \leq 1$ *recovers a low-rank version of spectral embedding.*

- Proof in paper/poster

Preserving Structure

A connectivity algorithm $G(K)$ such as *k*-nearest neighbors should be able to recover the edges from the coordinates such that $G(K) = A$



Preserving Graph Topology

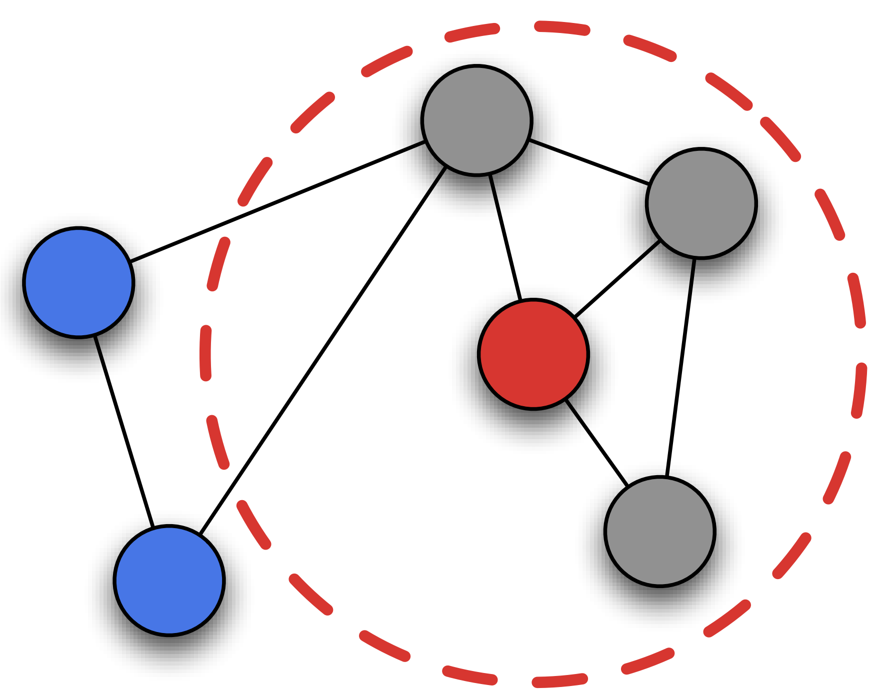
Using linear constraints

$\mathcal{G}(K)$	Linear constraints on K
k-nearest neighbors	$D_{ij} > (1 - A_{ij}) \max_m (A_{im} D_{im})$
ϵ-neighborhoods	$D_{ij}(A_{ij} - \frac{1}{2}) \leq \epsilon(A_{ij} - \frac{1}{2})$

Distance is linear function of K

$$D_{ij} = K_{ii} + K_{jj} - 2K_{ij}$$

Constraints prevent blue nodes from invading the neighborhood of the red node



Structure Preserving Embedding

Algorithm for nearest-neighbor graphs

Input	$A \in \mathbb{B}^{N \times N}$, connectivity algorithm $\mathcal{G}$, and parameter C.
Step 1	Solve SDP $\tilde{K} = \arg\max_{K \in \mathcal{K}} \mathrm{tr(KA) - C\xi}$ s.t. $D_{ij} > (1 - A_{ij}) \max_m (A_{im} D_{im}) - \xi$
Step 2	Apply SVD to $\tilde{K}$ and use the top eigenvectors as embedding coordinates

Preserving Graph Topology

Using linear constraints

$\mathcal{G}(K)$	Linear constraints on K
b-matching or max-weight spanning tree	???

b-matching:

$$\mathcal{G}(K) = \text{argmax}_{\tilde{A}} \sum_{ij} W_{ij}\tilde{A}_{ij} \quad \text{s.t.} \quad \sum_j \tilde{A}_{ij} = b_i, \tilde{A}_{ij} = \tilde{A}_{ji}, \tilde{A}_{ii} = 0, \tilde{A}_{ij} \in \{0, 1\}$$

max weight spanning tree:

$$\mathcal{G}(K) \;=\; \arg\max_{\tilde{A}} \sum_{ij} W_{ij}\tilde{A}_{ij} \text{ s.t. } \tilde{A} \in \mathcal{T}$$

Weight is linear function of K

$$W_{ij} = -D_{ij} = -K_{ii} - K_{jj} + 2K_{ij}$$

Preserving Graph Topology

Using linear constraints

$\mathcal{G}(K)$	Linear constraints on K
b-matching or max-weight spanning tree	$\sum_{ij} W_{ij} A_{ij} \geq \sum_{ij} W_{ij} \tilde{A}_{ij} \;\; \mathrm{s.t} \;\; \tilde{A} \in \mathcal{G}$

- Exponential number of constraints of this form
- Use cutting-plane technique to avoid enumeration, similar to SVM-struct [Finley, Joachims '08]
  - Iterate SDP adding worst violating constraint at each iteration

Structure Preserving Embedding

Algorithm for maximum-weight subgraphs

Input	$A \in \mathbb{B}^{N \times N}$, connectivity algorithm $\mathcal{G}$, and parameters C, ϵ.
Step 1	Solve SDP $\tilde{K} = \arg\max_{K \in \mathcal{K}} \mathrm{tr}(\mathrm{KA}) - \mathrm{C}\xi$.
Step 2	Use $\mathcal{G}$, $\tilde{K}$ to find biggest violator $\tilde{A} = \arg\max_{A} \mathrm{tr}(\tilde{\mathrm{W}}\mathrm{A})$.
Step 3	If $\lvert \mathrm{tr}(\tilde{\mathrm{W}}\tilde{\mathrm{A}}) - \mathrm{tr}(\tilde{\mathrm{W}}\mathrm{A}) \rvert > \epsilon$, add constraint $\mathrm{tr}(\mathrm{WA}) - \mathrm{tr}(\mathrm{W}\tilde{\mathrm{A}}) \geq \triangle(\tilde{\mathrm{A}}, \mathrm{A}) - \xi$ and go to Step 1
Step 4	Apply SVD to $\tilde{K}$ and use the top eigenvectors as embedding coordinates

Implementation

- MATLAB
- Using CSDP and SDP-LR [Borchers '99][Burer, Monteiro '03]
- Complexity similar to SDPs for dimensionality reduction
    - $O(N^3 + C^3)$ where C is the number of constraints
    - Many inactive constraints, working-set method
    - SDP-LR takes advantage of low-rank objective
    - Run on graphs with up to 1000 nodes

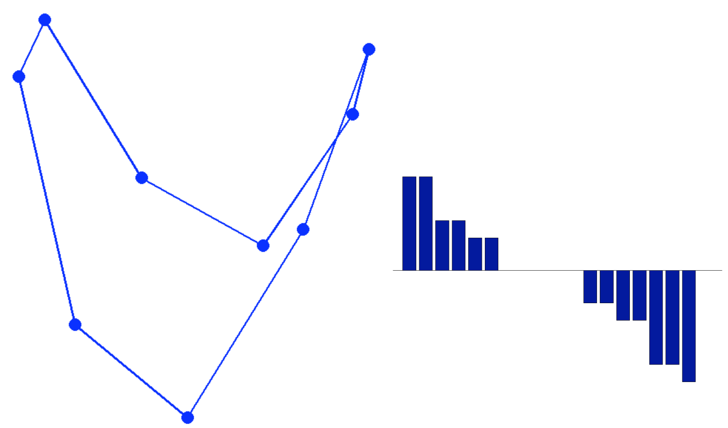
Outline

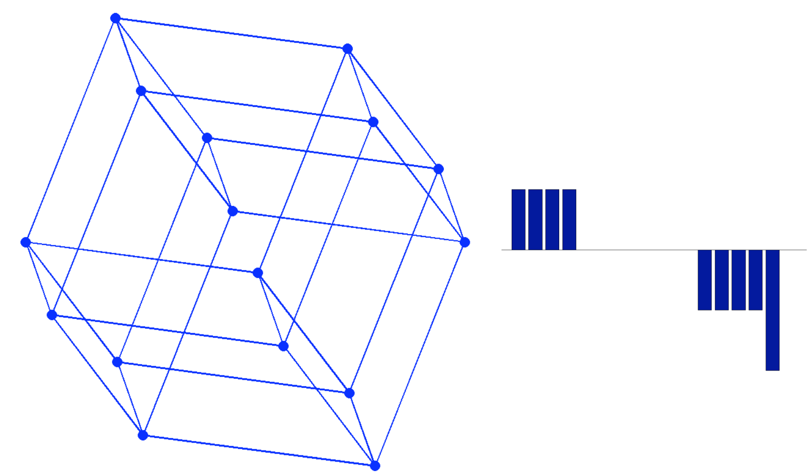
- Introduction to graph embedding
  - Applications, background, SDP + SVD, Möbius example
- Structure Preserving Embedding
  - Graph topology as linear constraints
  - A low-rank objective
  - Algorithm details, implementation
- Experiments
  - Classical graphs, molecules, political blogs
- Dimensionality Reduction
- Review

Experiments

Classical graphs

Spectral Embedding

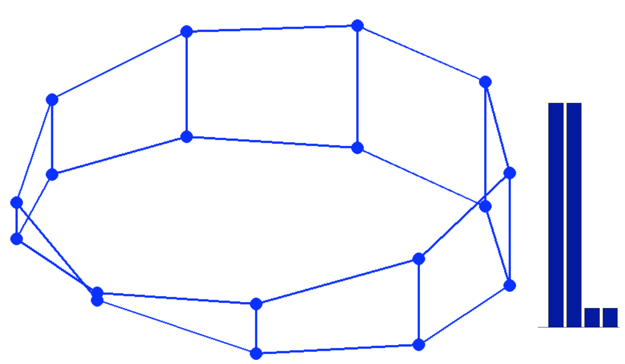


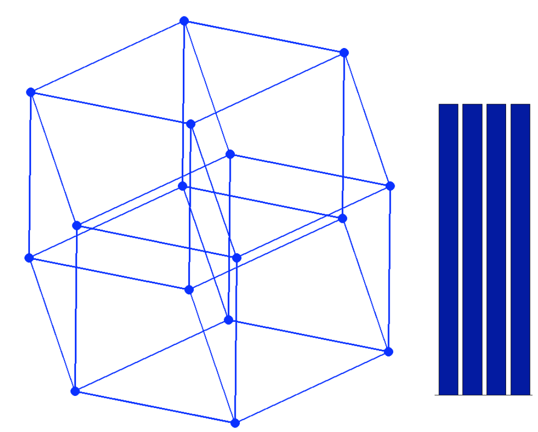


Möbius Ladder

Tesseract

Structure Preserving Embedding (SPE)

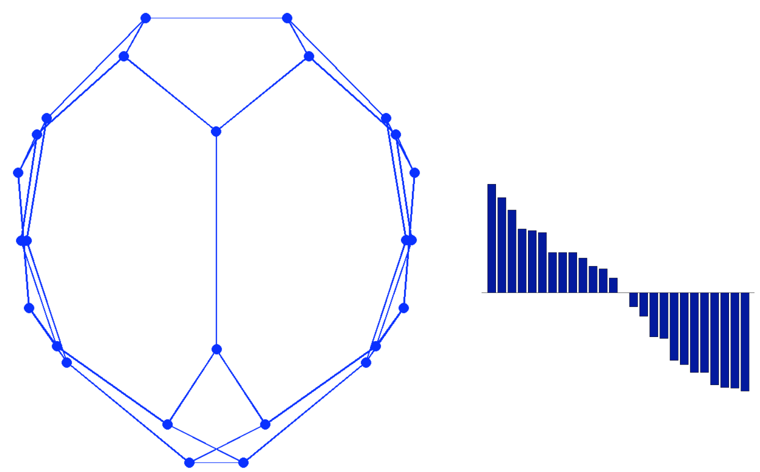


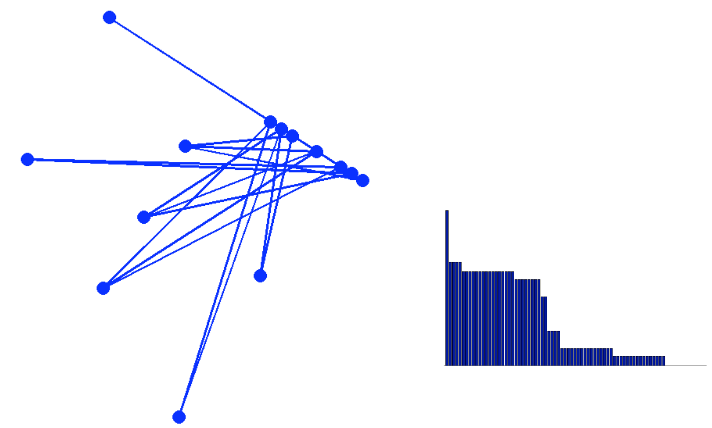


Experiments

Classical graphs

Spectral Embedding

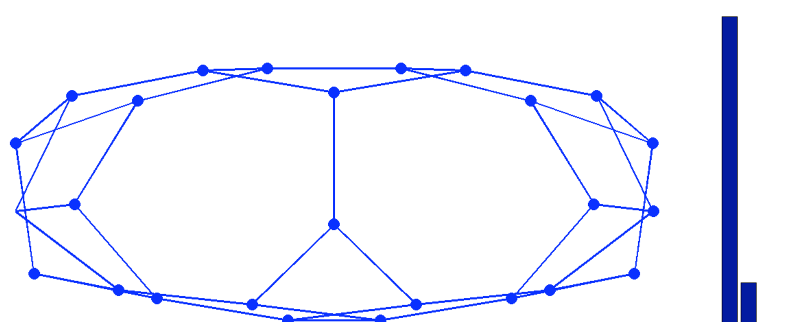


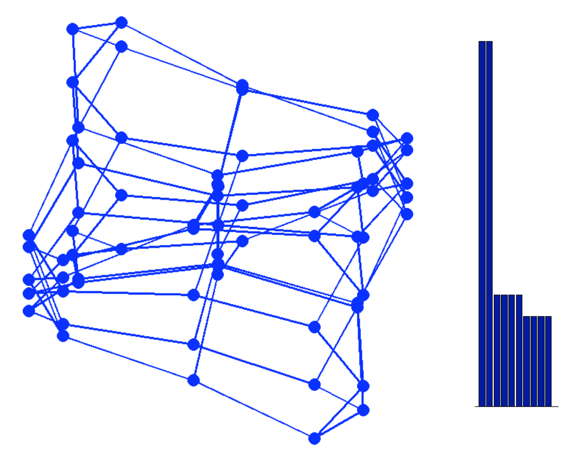


Celmins Swart Snark

Balaban 10-cage

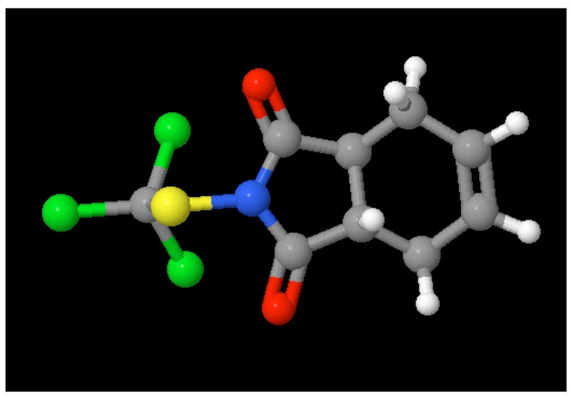
Structure Preserving Embedding (SPE)



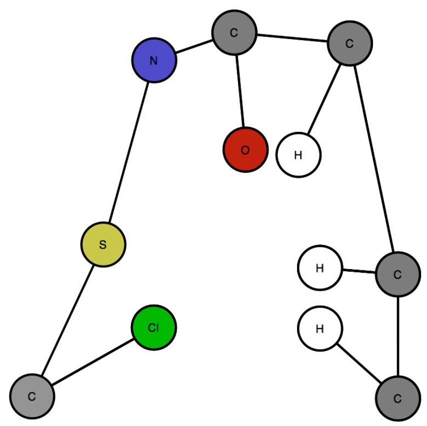


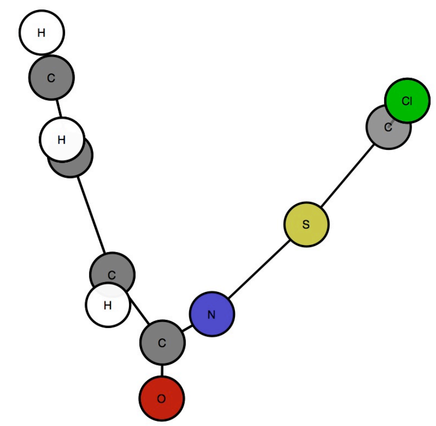
Experiments

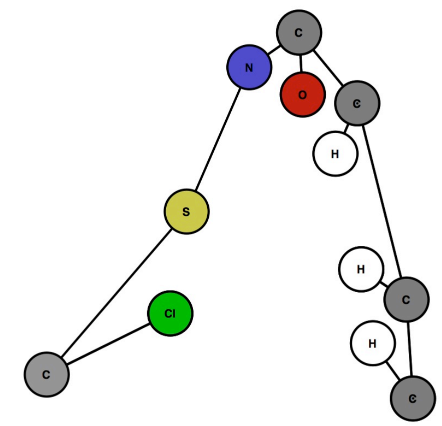
Molecules

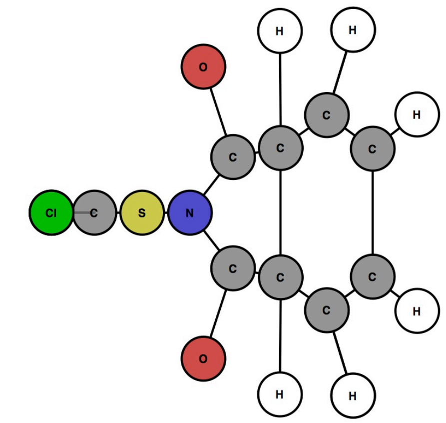


Molecule TR015







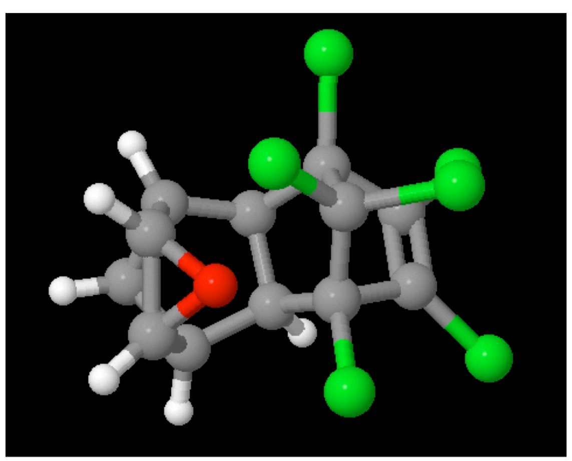


Spectral Embedding

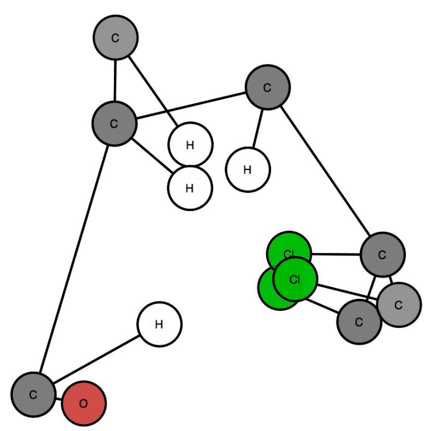
Laplacian Eigenmaps

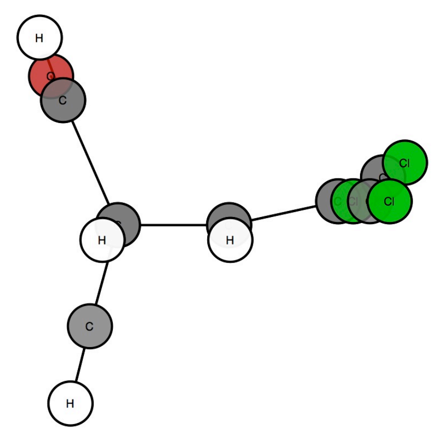
Normalized Laplacian Eigenmaps

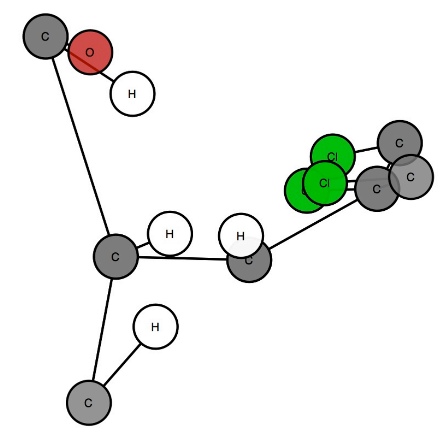
Structure Preserving Embedding (SPE)

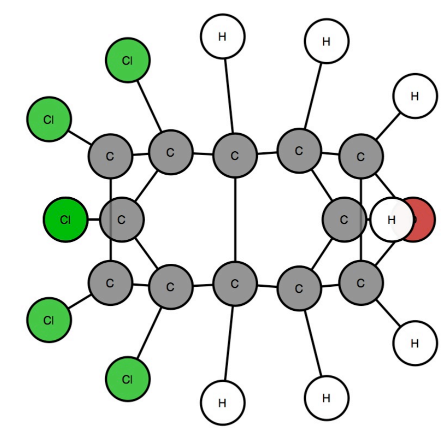


Molecule TR012





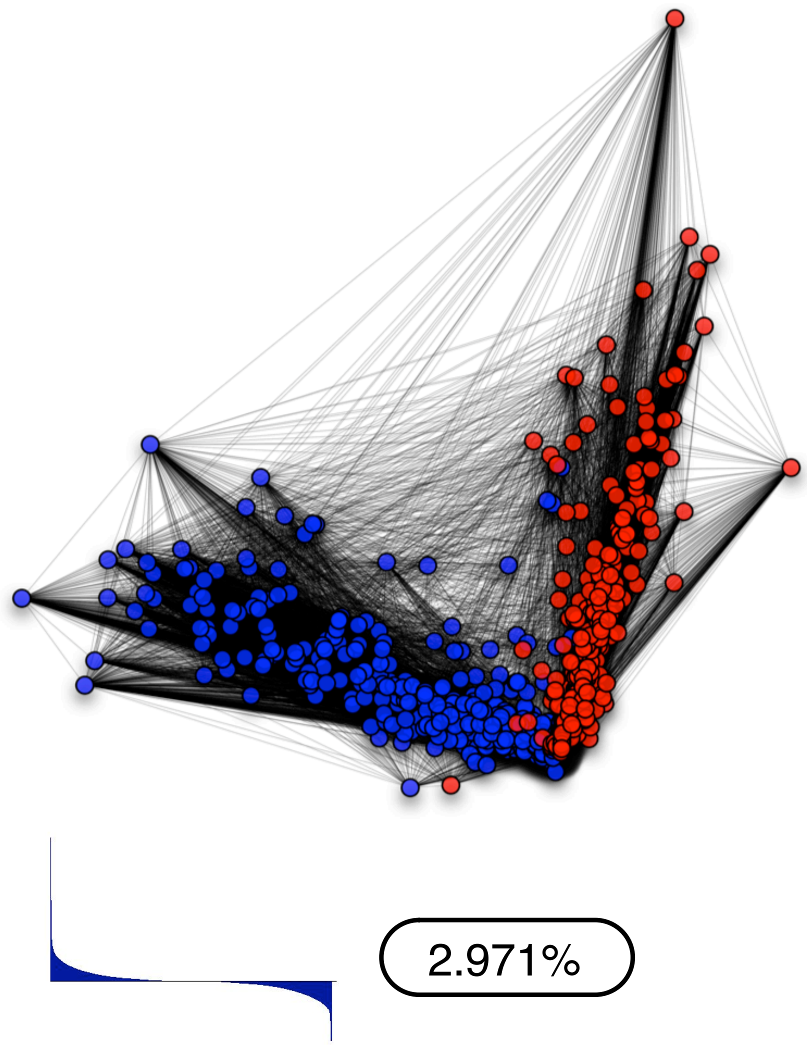




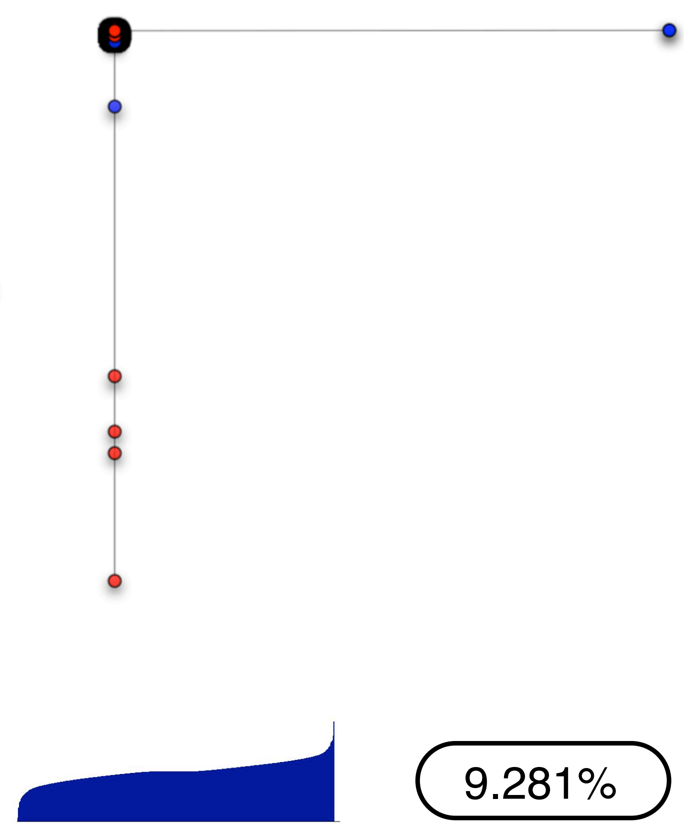
Experiments

Political blogs

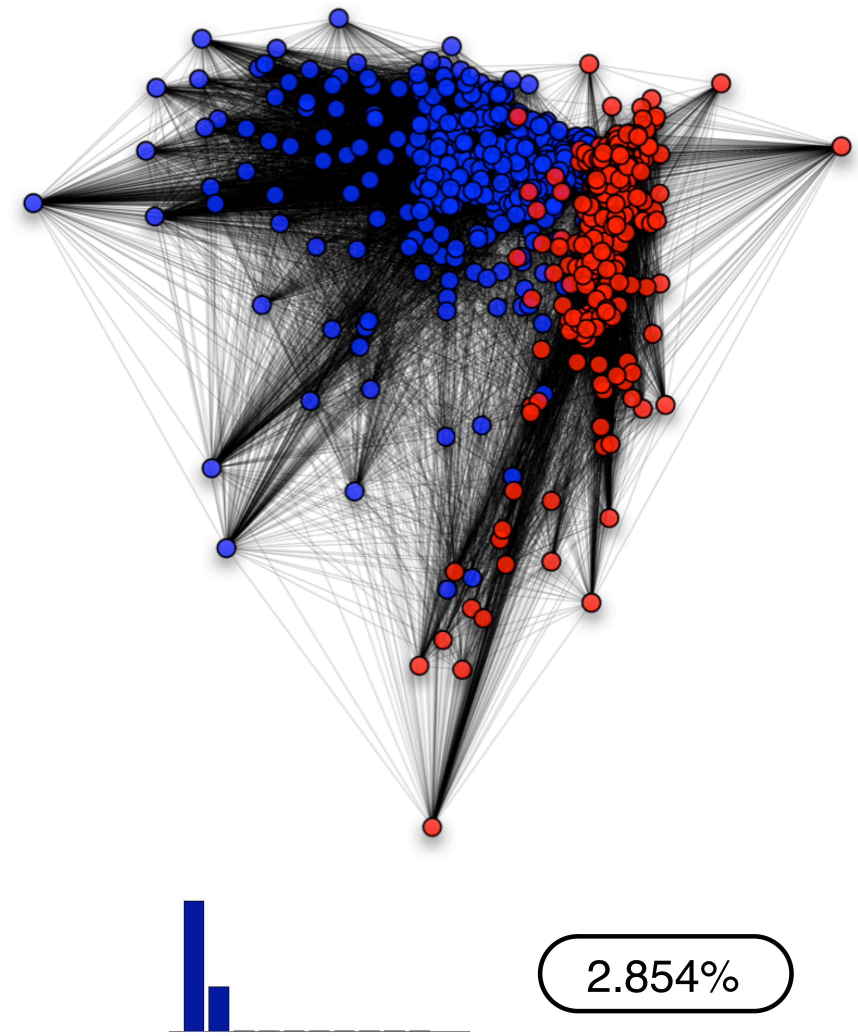
Link structure between 981 political blogs
Red is conservative, blue is liberal, reconstruction error shown as %



Spectral Embedding



Normalized Laplacian Eigenmaps

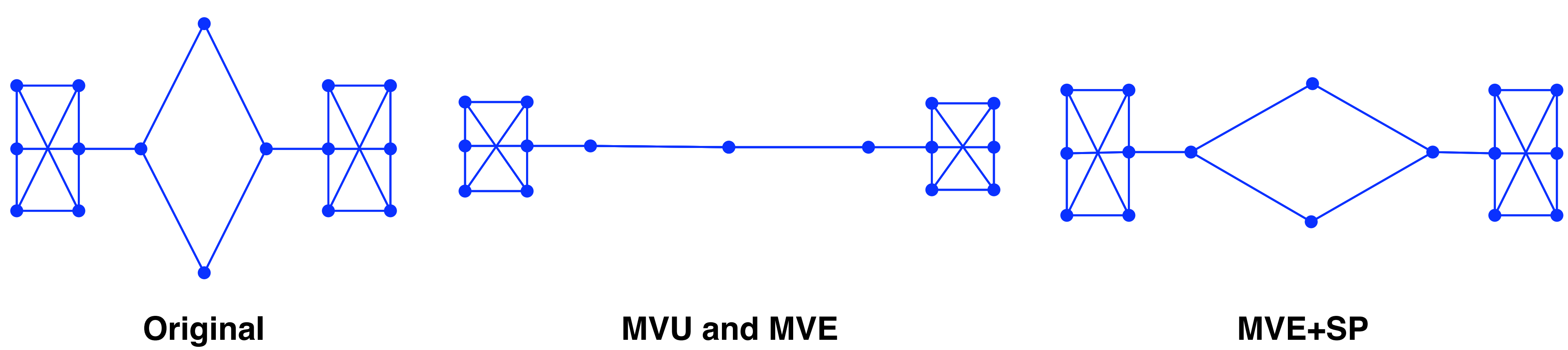


Structure Preserving Embedding (SPE)

Dimensionality Reduction

Preserving distances and graph topology

- SPE is similar to manifold-learning methods for dimensionality reduction such as LLE, MVU, MVE
  [Roweis, Saul '05] [Weinberger et. al '05] [Shaw, Jebara '07]
- These methods preserve pairwise distances between datapoints
- Adding topology-preserving constraints yields more accurate embeddings, prevents collapsing parts of the underlying manifold



Dimensionality Reduction

Preserving distances and graph topology

MVU objective

$$\max_{K \in \mathcal{K}} \operatorname{tr}(K)$$

MVE objective

$$\max_{K \in \mathcal{K}} \sum_{i=1}^{d} \lambda_i - \sum_{i=d+1}^{N} \lambda_i$$

$$\mathcal{K} = \left\{ \forall K \in \mathbb{R}^{N \times N} \;\middle|\; \begin{array}{l} K \succeq 0 \\ \sum_{ij} K_{ij} = 0 \\ K_{ii} + K_{jj} - K_{ij} - K_{ji} = W_{ii} + W_{jj} - W_{ij} - W_{ji} \\ \forall_{i,j} \text{ s.t. } A_{ij} = 1 \end{array} \right\}$$

For nearest-neighbor graphs

$$K_{ii} + K_{jj} - K_{ij} - K_{ji} > (1 - A_{ij}) \max_m (A_{im}(K_{ii} + K_{mm} - K_{im} - K_{mi})) \;\forall_{i,j}$$

For maximum-weight graphs

$$\sum_{ij} (-K_{ii} - K_{jj} + K_{ij} + K_{ji}) A_{ij} - \sum_{ij} (-K_{ii} - K_{jj} + K_{ij} + K_{ji}) \tilde{A}_{ij} \geq \triangle(\tilde{A}, A) - \xi \forall_{i,j}$$

Experiments

MVE+SP

- 1-nearest-neighbor classifier on UCI datasets
- Compare using 2 dimensions per point vs. using all dimensions

Accuracy % of 1NN classifier

	KPCA	MVU	MVE	MVE+SP	All-Dimensions
Ionosphere	66.0%	85.0%	81.2%	**87.1**%	78.8%
Cars	66.1%	70.1%	71.6%	**78.1**%	79.3%
Dermatology	58.8%	63.6%	64.8%	**66.3**%	76.3%
Ecoli	94.9%	95.6%	94.8%	**96.0**%	95.6%
Wine	68.0%	68.5%	68.3%	**69.7**%	71.5%
OptDigits 4 vs. 9	94.4%	99.2%	99.6%	**99.8**%	98.6%

Conclusion

- SPE finds low-dimensional representations of graphs that implicitly preserve topology
- Preserving local distances is insufficient for faithfully embedding graphs in low-dimensional space, need to preserve graph topology