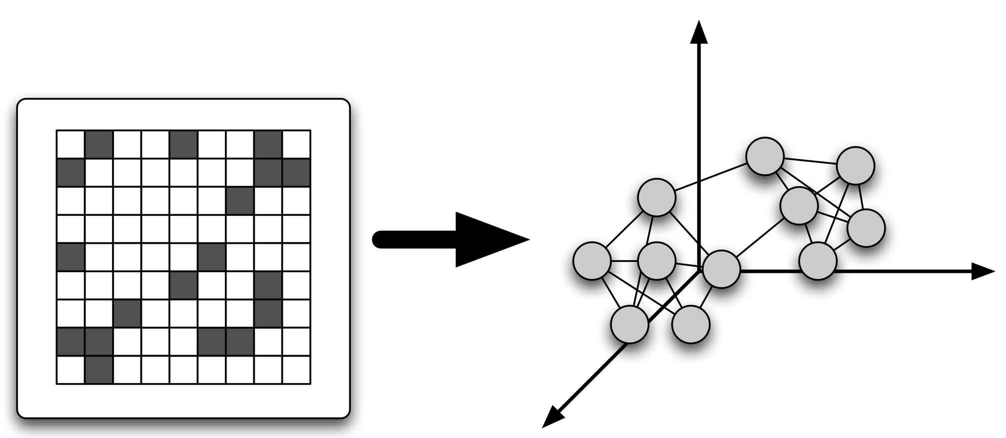
Structure Preserving Embedding

Blake Shaw
Tony Jebara
Columbia University

Introduction

Graph Embedding from Connectivity information

Given only connectivity information describing which nodes in a graph are connected, can we learn a set of low-dimensional coordinates for each node such that these coordinates can easily be used to reconstruct the original structure of the network?



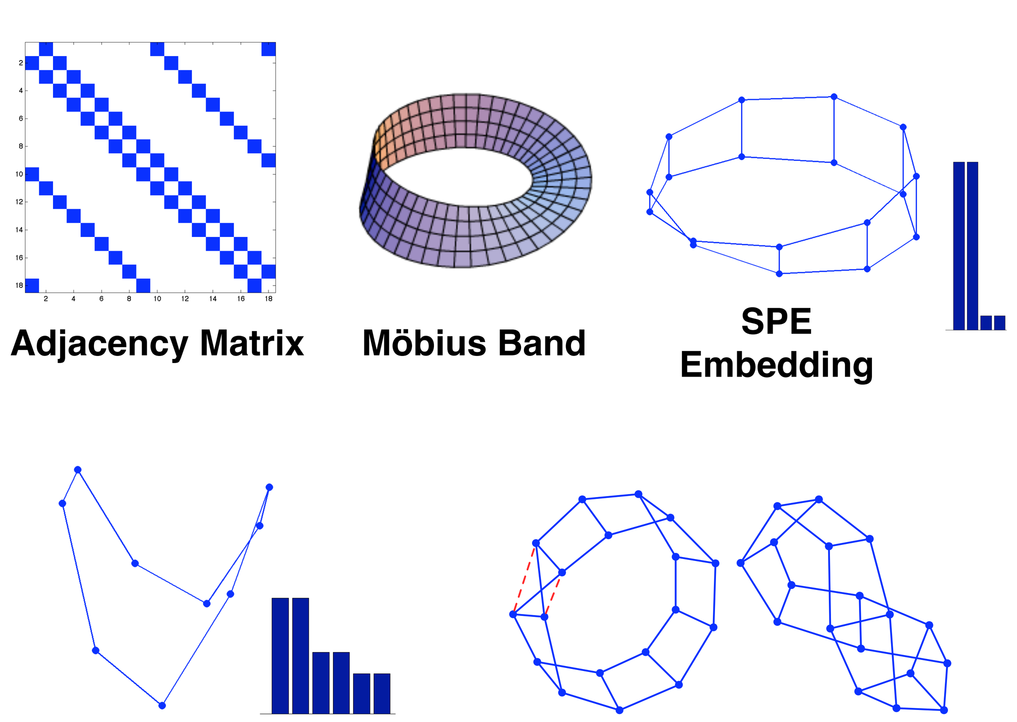
Input: binary adjacency matrix $A \in \mathbb{B}^{N \times N}$

Output: coordinates for each node $\vec{y}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$

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.
Spring embedding - simulate physical system where edges are springs, use Hookes law to compute forces.

Embedding the Möbius Ladder

Traditional graph embedding algorithms such as spectral embedding and spring embedding do not explicitly preserve structure according to our definition and thus in practice perform poorly in accurately visualizing many simple classical graphs such as the Mobius ladder:



Adjacency Matrix — Möbius Band — SPE Embedding — Spectral Embedding — Two Spring Embeddings

The Algorithm

Structure Preserving Embedding

SPE is an efficient convex optimization based on semidefinite programming for finding an embedding $K \in \mathbb{R}^{N \times N}$ such that K is both low-rank and structure preserving.

Using the eigenvectors of K with the largest eigenvalues as coordinates for the nodes, we get a low-dimensional embedding.

$$\boxed{A} \xrightarrow{\text{SDP}} \boxed{K} \xrightarrow{\text{SVD}} \boxed{\vec{y}_i}$$

Structure Preserving Constraints

Given a connectivity algorithm $\mathcal{G}$ (such as k-nearest neighbors, b-matching, or maximum weight spanning tree) which accepts as input a kernel K specifying an embedding and returns an adjacency matrix, we call an embedding *structure preserving* if the application of $\mathcal{G}$ to K exactly reproduces the input graph: $\mathcal{G}(K) = A$.

Constraints are linear in K

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

$\mathcal{G}(K) \rightarrow$ **k-nearest neighbors:**

$$D_{ij} > (1 - A_{ij}) \max_m (A_{im} D_{im})$$

$\mathcal{G}(K) \rightarrow$ **epsilon-balls:**

$$D_{ij}(A_{ij} - \tfrac{1}{2}) \leq \epsilon(A_{ij} - \tfrac{1}{2})$$

$\mathcal{G}(K) \rightarrow$ **maximum weight subgraphs:**

b-matching:

$$\mathcal{G}(K) = \arg\max_{\tilde{A}} \sum_{ij} W_{ij} \tilde{A}_{ij}$$

$$\text{s.t.} \sum_j \tilde{A}_{ij} = b_i, \tilde{A}_{ij} = \tilde{A}_{ji}, \tilde{A}_{ii} = 0, \tilde{A}_{ij} \in \{0, 1\}$$

max-weight spanning tree:

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

exponential number of constraints of form:

$$\sum_{ij} W_{ij} A_{ij} \geq \sum_{ij} W_{ij} \tilde{A}_{ij} \quad \text{s.t } \tilde{A} \in \mathcal{G}$$

avoid enumeration w/ cutting-plane algorithm

Low-rank Objective

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

Proof. Let $K = U \Lambda U^T$ and $A = V \tilde{\Lambda} V^T$ and insert into the objective function:

$$\begin{aligned}
\max_{K \succeq 0} \operatorname{tr}(KA) &= \max_{\Lambda \in \mathcal{L}, U \in \mathcal{O}} \operatorname{tr}(U \Lambda U^T V \tilde{\Lambda} V^T) \\
&= \max_{\Lambda \in \mathcal{L}, U \in \mathcal{O}} \operatorname{tr}((V^T U) \Lambda (V^T U)^T \tilde{\Lambda}) \\
&= \max_{\Lambda \in \mathcal{L}, R \in \mathcal{O}} \operatorname{tr}(R \Lambda R^T \tilde{\Lambda}) \\
&= \max_{\lambda \geq 0, \lambda^T \mathbf{1} \leq 1} \lambda^T \tilde{\lambda} \\
&= \tilde{\lambda}_1.
\end{aligned}$$

Thus, the maximization problem reproduces spectral embedding while pushing all the spectrum weight into the top eigenvalue. The (rank 1) solution must be $K = vv^T$ where v is the leading eigenvector of A. $\square$

Algorithm Overview

Table 1: Structure Preserving Embedding algorithm for k-nearest neighbor constraints.

Input	$A \in \mathbb{B}^{N \times N}$, connectivity algorithm $\mathcal{G}$, and parameter C.
Step 1	Solve SDP $\tilde{K} = \arg\max_{K \in \mathcal{K}} \operatorname{tr}(KA) - C\xi$ 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

Table 2: Structure Preserving Embedding algorithm with cutting-plane constraints.

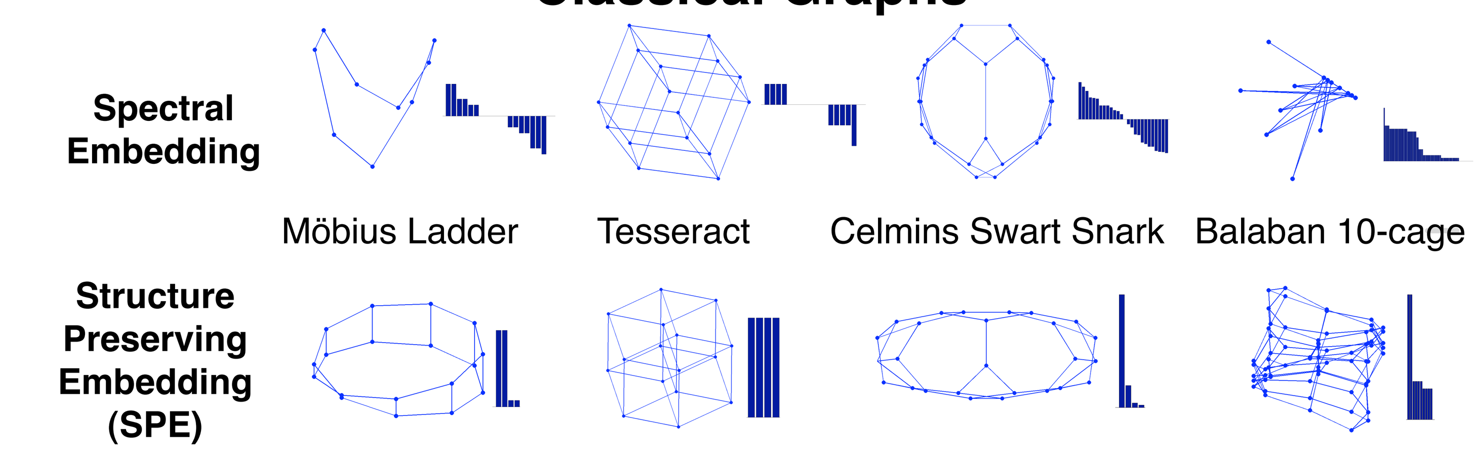
Input	$A \in \mathbb{B}^{N \times N}$, connectivity algorithm $\mathcal{G}$, and parameters C, ϵ.
Step 1	Solve SDP $\tilde{K} = \arg\max_{K \in \mathcal{K}} \operatorname{tr}(KA) - C\xi$.
Step 2	Use $\mathcal{G}$, $\tilde{K}$ to find biggest violator $\tilde{A} = \arg\max_A \operatorname{tr}(\tilde{W}A)$.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

Dimensionality Reduction

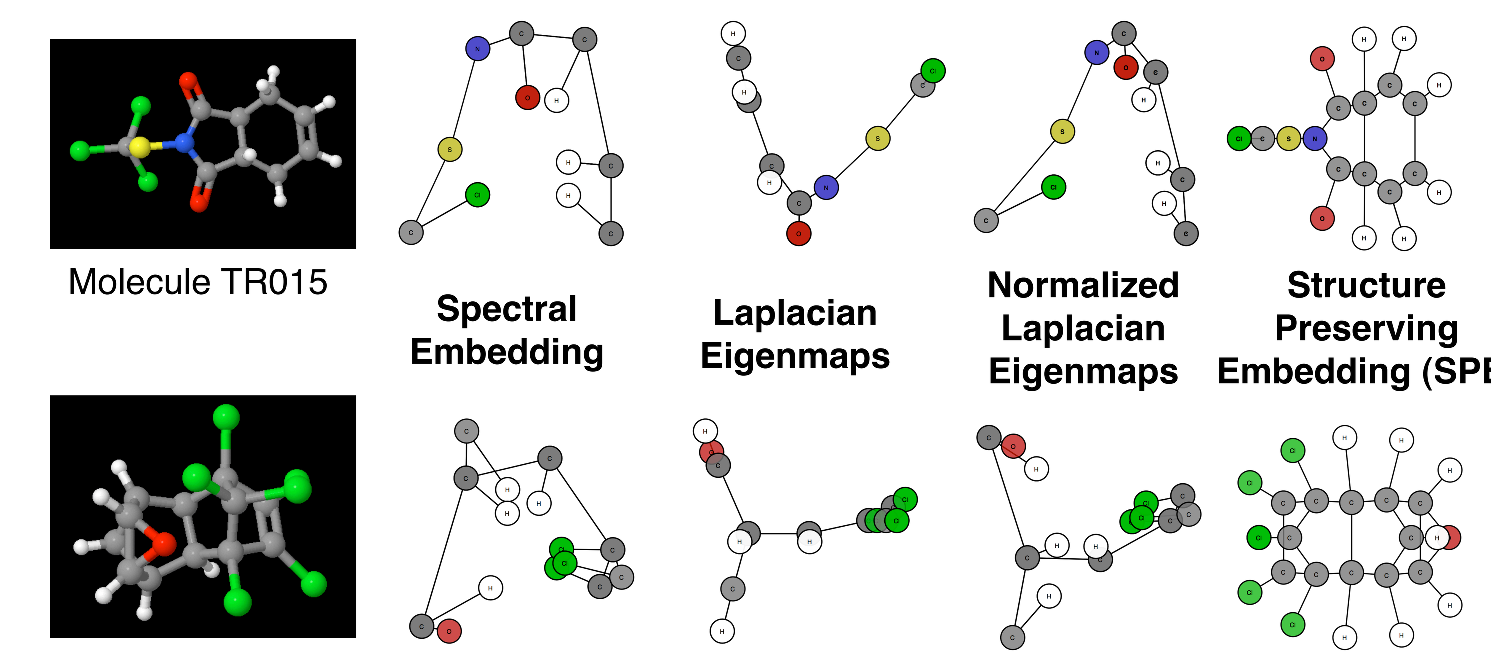
Structure preserving constraints can also benefit dimensionality reduction algorithms. These methods similarly find compact coordinates that preserve certain properties of the input data. Many of these manifold learning techniques preserve local distances but not graph topology. We show that adding explicit topological constraints to these existing algorithms is crucial for preventing folding and collapsing problems that occur in dimensionality reduction.

Results

Classical Graphs

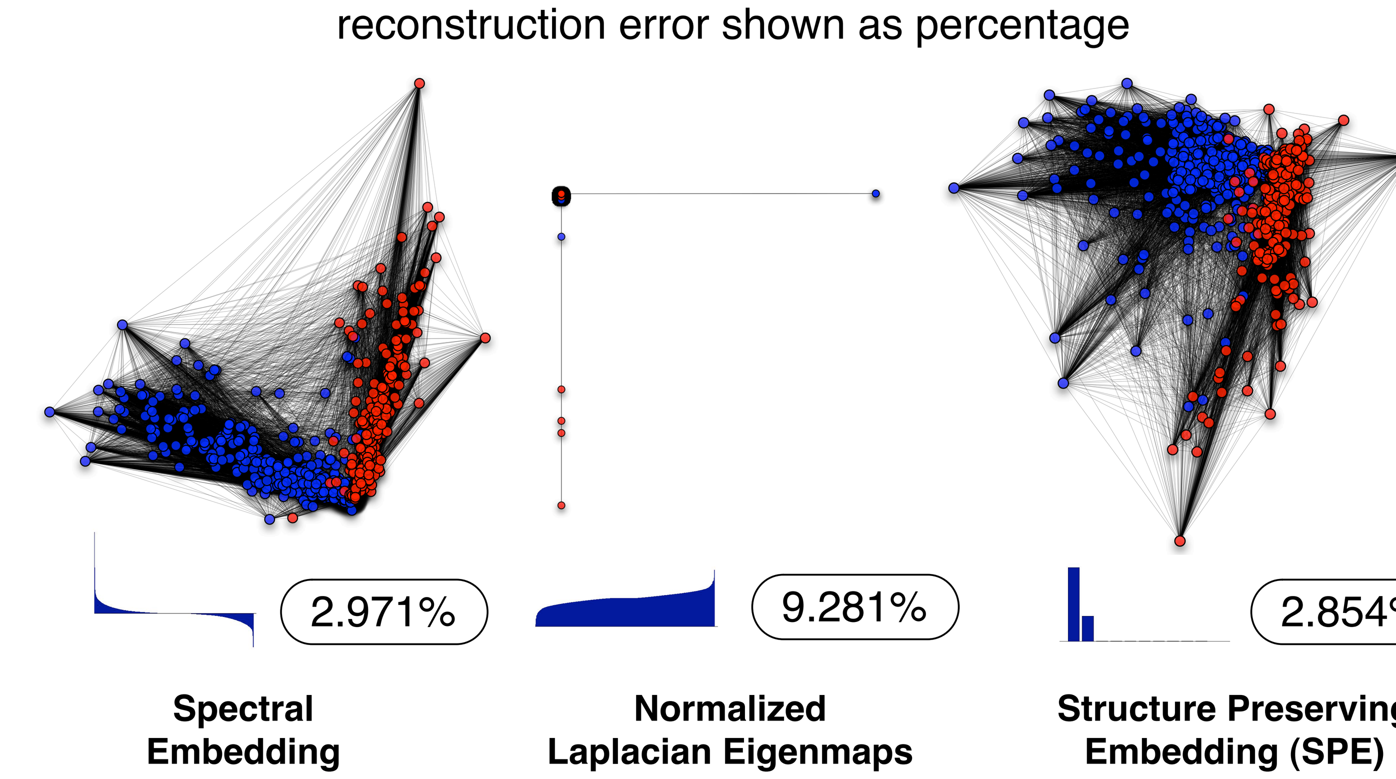


Molecules

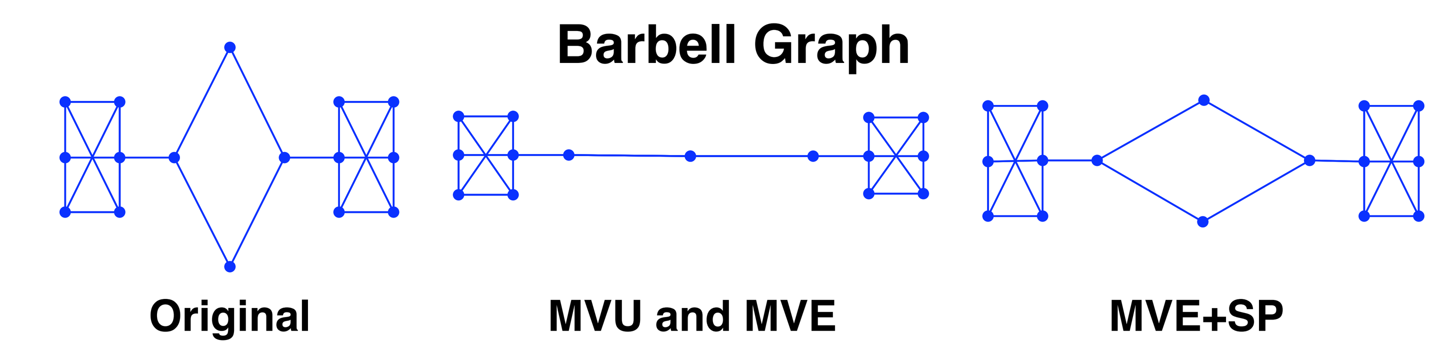


Political Blogs

link structure of 981 blogs, red is conservative, blue is liberal
reconstruction error shown as percentage



Barbell Graph



MVE+SP for UCI datasets

1-nearest neighbor classifer accuracy on 2D embeddings

	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%