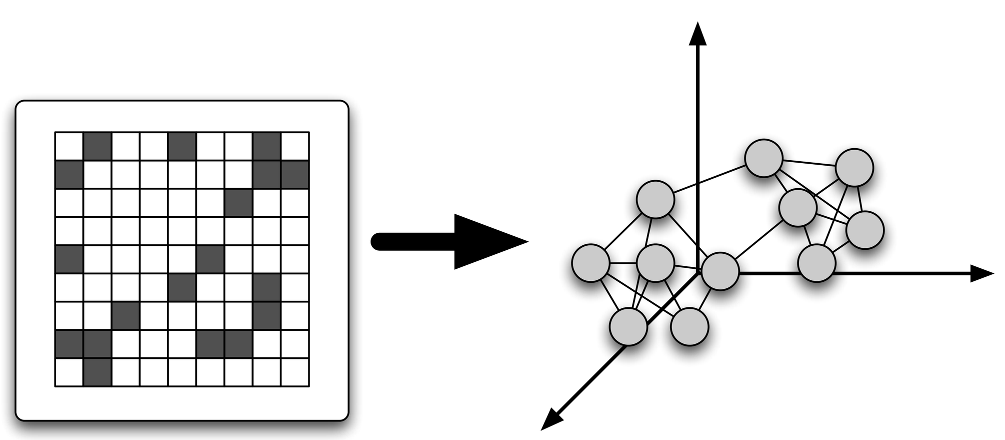
Visualizing Social Networks with Structure Preserving Embedding

Blake Shaw
Tony Jebara
Columbia University

Introduction

Social Network Visualization as Low-Dimensional Graph Embedding

From 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 $\mathbf{A} \in \mathbb{B}^{n \times n}$

Output: coordinates for each node $\mathbf{L} \in \mathbb{R}^{d \times n}$

Spectral embedding - decompose adjacency matrix $\mathbf{A}$ with an SVD and use eigenvectors with highest eigenvalues for coordinates.

Laplacian Eigenmaps (Belkin, Niyogi '02) - form graph laplacian from adjacency matrix, $\mathcal{L} = \mathbf{D} - \mathbf{A}$, apply SVD to $\mathcal{L}$ 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.

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 $\mathbf{K} = \mathbf{L}^\top \mathbf{L}$ specifying an embedding $\mathbf{L}$ and returns an adjacency matrix, we call an embedding *structure preserving* if the application of $\mathcal{G}$ to $\mathbf{K}$ exactly reproduces the input graph: $\mathcal{G}(\mathbf{K}) = \mathbf{A}$.

Constraints are linear in $\mathbf{K}$

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

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

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

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

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

The Algorithm

Structure Preserving Embedding optimized via SDP + SVD

SPE for greedy nearest-neighbor constraints solves the following SDP:

$$\max_{\mathbf{K} \in \mathcal{K}} \operatorname{tr}(\mathbf{KA})$$
$$D_{ij} > (1 - A_{ij}) \max_m (A_{im} D_{im}) \;\; \forall_{i,j}$$

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

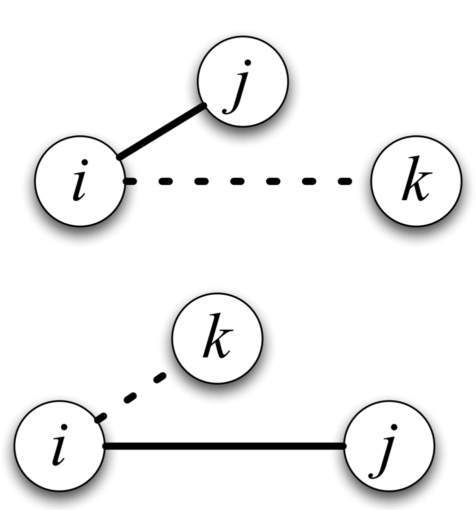
SPE $\quad \mathbf{A} \xrightarrow{\text{SDP}} \mathbf{K} \xrightarrow{\text{SVD}} \mathbf{L}$

Structure Preserving Embedding optimized via SGD

As first proposed, SPE learns a matrix $\mathbf{K}$ via a semidefinite program (SDP) and then decomposes $\mathbf{K} = \mathbf{L}^\top \mathbf{L}$ by performing singular value decomposition. We propose optimizing $\mathbf{L}$ directly using stochastic gradient descent (SGD).

SPE-SGD $\quad \mathbf{A} \xrightarrow{\text{SGD}} \mathbf{L}$

Structure preserving constraints can be written as a set of matrices $S = \{\mathbf{C}_1, \mathbf{C}_2, ...\mathbf{C}_m\}$, where each $\mathbf{C}_l$ is a constraint matrix corresponding to a triplet (i,j,k) such that $A_{ij} = 1$ and $A_{ik} = 0$. This set of all triplets clearly subsumes the SPE distance constraints, and allows each individual constraint to be written as $\operatorname{tr}(\mathbf{C}_l \mathbf{K}) > 0$ where $\operatorname{tr}(\mathbf{C}_l \mathbf{K}) = K_{jj} - 2K_{ij} + 2K_{ik} - K_{kk}$.



$\operatorname{tr}(\mathbf{C}_l \mathbf{K}) < 0$

$\operatorname{tr}(\mathbf{C}_l \mathbf{K}) > 0$

Temporarily dropping the centering and scaling constraints, we can now formulate the SDP above as maximizing the following objective function over $\mathbf{L}$:

$$f(\mathbf{L}) = \lambda \operatorname{tr}(\mathbf{L}^\top \mathbf{L} \mathbf{A}) - \sum_{l \in S} \max(\operatorname{tr}(\mathbf{C}_l \mathbf{L}^\top \mathbf{L}), 0).$$

Structure Preserving Embedding optimized via SGD cont.

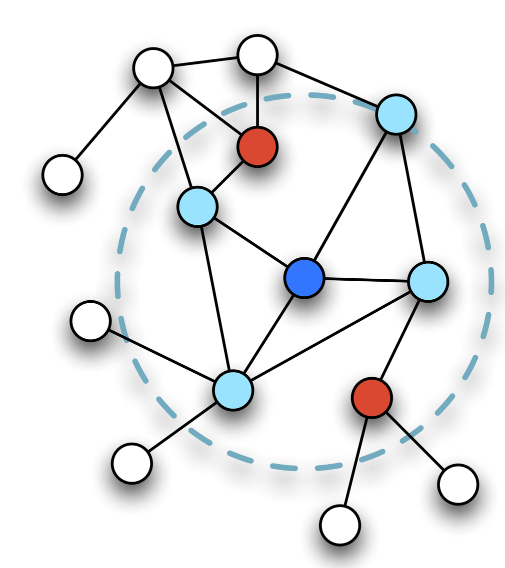
We will maximize $f(\mathbf{L})$ via projected stochastic subgradient decent. Define the subgradient in terms of a single randomly chosen triplet:

$$\Delta(f(\mathbf{L}), \mathbf{C}_l) = \begin{cases} 2\mathbf{L}(\lambda \mathbf{A} - \mathbf{C}_l) & \text{if } \operatorname{tr}(\mathbf{C}_l \mathbf{L}^\top \mathbf{L}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

and for each randomly chosen triplet constraint C_l, if $\operatorname{tr}(\mathbf{C}_l \mathbf{L}^\top \mathbf{L}) > 0$ then update $\mathbf{L}$ according to:

$$\mathbf{L}_{t+1} = \mathbf{L}_t + \eta \Delta(f(\mathbf{L}_t), \mathbf{C}_l)$$

where the step-size $\eta = \frac{1}{\sqrt{t}}$. After each step, we can use projection to enforce that $\operatorname{tr}(\mathbf{L}^\top \mathbf{L}) \leq 1$ and $\sum_{ij} (\mathbf{L}^\top \mathbf{L})_{ij} = 0$, by subtracting the mean from $\mathbf{L}$ and dividing each entry of $\mathbf{L}$ by its Frobenius norm.



Impostors
Red nodes incur hinge loss violations because they are impostors of the blue nodes neighborhood.

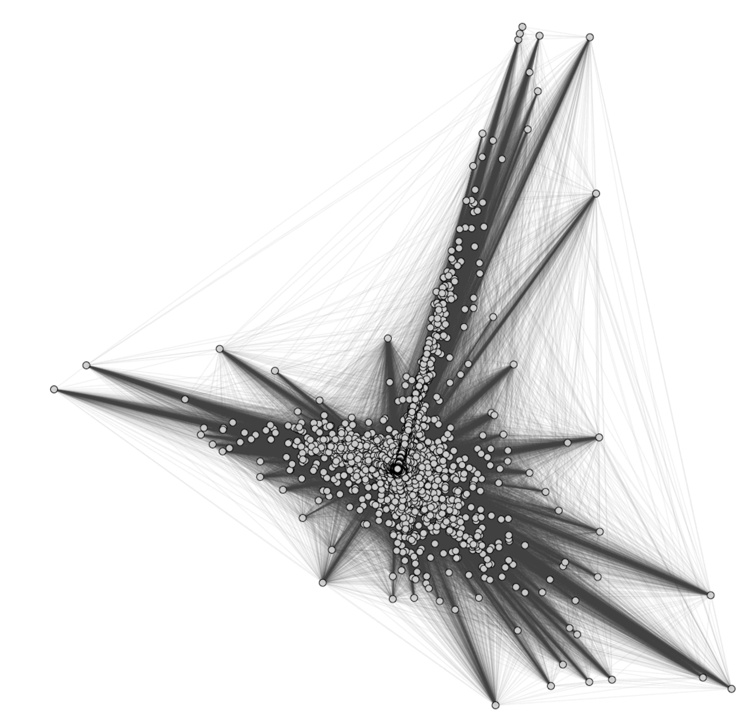
Algorithm 1 Large-Scale Structure Preserving Embedding

Require: $\mathbf{A} \in \mathbb{B}^{n \times n}$, dimensionality d, regularizer parameter λ, and maximum iterations T
Initialize $\mathbf{L}_0 \leftarrow \operatorname{rand}(d, n)$
(or optionally initialize to spectral embedding or Laplacian eigenmaps solution)
$t \leftarrow 0$
repeat
$\quad \eta_t \leftarrow \frac{1}{\sqrt{t+1}}$
$\quad i \leftarrow \operatorname{rand}(1 \dots n)$
$\quad j = \arg\min_j \| L_i - L_j \|_2 \;\; \forall_j$ s.t. $A(i,j) = 1$
$\quad \mathbf{C} \leftarrow \operatorname{zeros}(n \times n)$
$\quad \mathbf{C}_{jj} \leftarrow 1, \mathbf{C}_{ij} \leftarrow -1, \mathbf{C}_{ji} \leftarrow -1$
$\quad$**for all** k s.t. $\| L_i - L_k \|_2 < \| L_i - L_j \|_2$ AND $A(i,k) = 0$ **do**
$\quad\quad \mathbf{C}_{ik} \leftarrow 1, \mathbf{C}_{ki} \leftarrow 1, \mathbf{C}_{kk} \leftarrow -1$
$\quad$**end for**
$\quad \nabla_t \leftarrow 2\mathbf{L}_t (\lambda \mathbf{A} - \mathbf{C})$
$\quad \mathbf{L}_{t+1} \leftarrow \mathbf{L}_t + \eta_t \nabla_t$
$\quad$ {Subtract out mean}
$\quad \mathbf{L}_{t+1} = \frac{\mathbf{L}_{t+1}}{\|\mathbf{L}_{t+1}\|_2}$ {Project on to unit sphere}
$\quad t \leftarrow t + 1$
until $t \geq T$
return $\mathbf{L}$

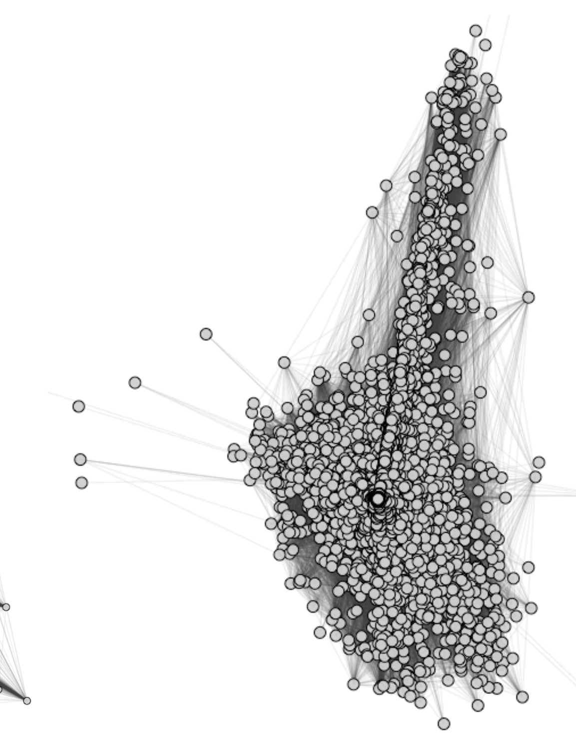
Results

Visualizing the Enron Email Network

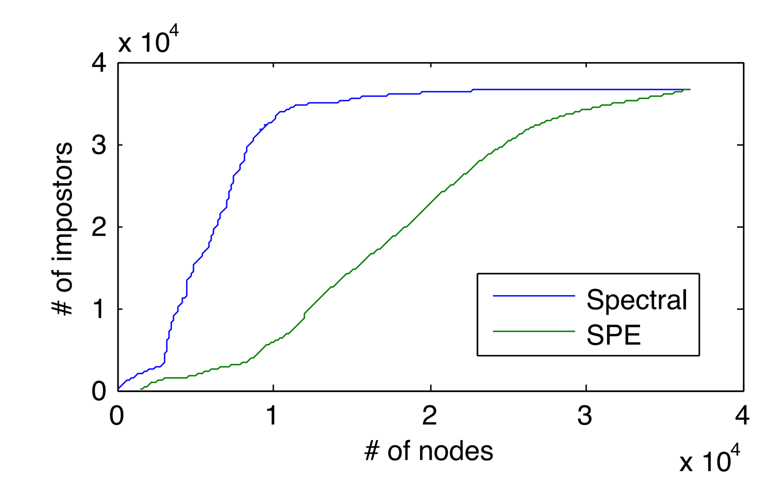
link structure of email correspondence between 36692 Enron employees



Spectral Embedding



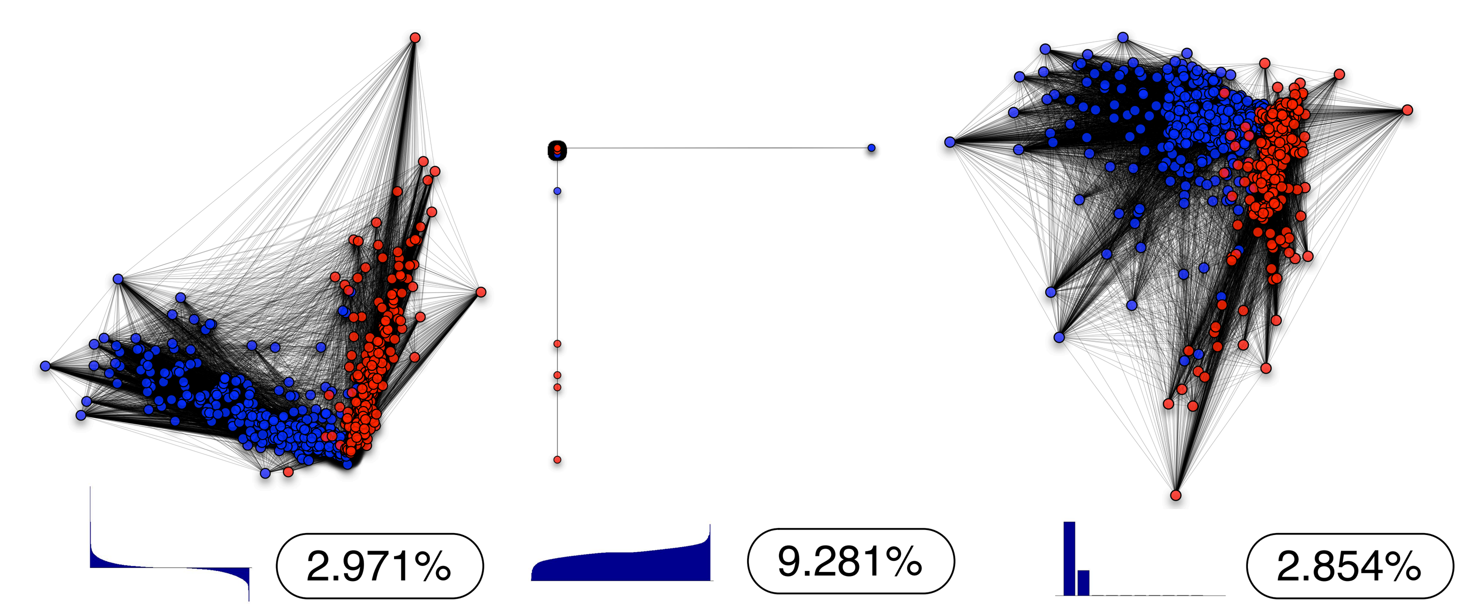
SPE-SGD



# of nodes with fewer than x impostors

Political Blogs

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



Spectral Embedding

Normalized Laplacian Eigenmaps

Structure Preserving Embedding (SPE)