Visualizing Social Networks with Structure Preserving Embedding

Blake Shaw and Tony Jebara
Department of Computer Science, Columbia University, New York, NY 10027
{blake, jebara}@cs.columbia.edu

1 Introduction

We propose an adaptation to Structure Preserving Embedding (SPE) based on stochastic gradient decent that allows for visualization of large social network datasets. SPE finds a low-dimensional representation of nodes in a network which is *structure-preserving*, meaning a connectivity algorithm such as k-nearest neighbors will recover the original connectivity pattern of the network exactly from only the coordinates of the nodes in the low-dimensional embedding. There are many possible goals for network visualization algorithms, such as minimizing edge crossings, bringing neighbors close, pushing away unconnected nodes, highlighting clusters, and preserving graph distances. We propose that accurate visualizations of social networks should preserve the underlying topological structure of the network. In previous work, we have presented Structure Preserving Embedding (SPE) [3], an algorithm based on semidefinite programming and singular value decompositions designed to find such embeddings. In this abstract, we present a low-rank approximation to the original algorithm, implemented using a fast custom solver based on projected stochastic gradient descent, which allows the technique to scale to larger networks.

2 Algorithm

Given a network of n nodes represented as a graph with adjacency matrix $\mathbf{A} \in \mathbb{B}^{n \times n}$, SPE finds an embedding $\mathbf{L} \in \mathbb{R}^{d \times n}$ such that d is small and running a connectivity algorithm such as k-nearest neighbors on $\mathbf{L}$ returns $\mathbf{A}$. 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. In contrast, this article proposes optimizing $\mathbf{L}$ directly. Although for $d < N$, this problem is now non-convex, because of the stochastic nature of the optimizer we have found the algorithm does not suffer from local minima in practice.

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

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

where $D_{ij} = K_{ii} + K_{jj} - 2K_{ij}$ and $\mathcal{K} = \{\mathbf{K} \succeq 0,\ \operatorname{tr}(\mathbf{K}) \leq 1, \sum_{ij} K_{ij} = 0\}$. The constraints require the embedding of each node to be more distant from its non-neighbors than its neighbors. Let $S = \{\mathbf{C}_1, \mathbf{C}_2, ...\mathbf{C}_m\}$ be the set of all triplet constraints, 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 distance constraints above, and allows each individual constraint to be written as $\operatorname{tr}(\mathbf{C}_l \mathbf{K}) > 0$ where $\operatorname{tr}(\mathbf{C}_l \mathbf{K}) = K_{jj} - 2K_{ij} + 2K_{ik} - K_{kk}$. Temporarily dropping the centering and scaling constraints, we can now formulate the SDP above as maximizing the following objective function over $\mathbf{L}$:

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

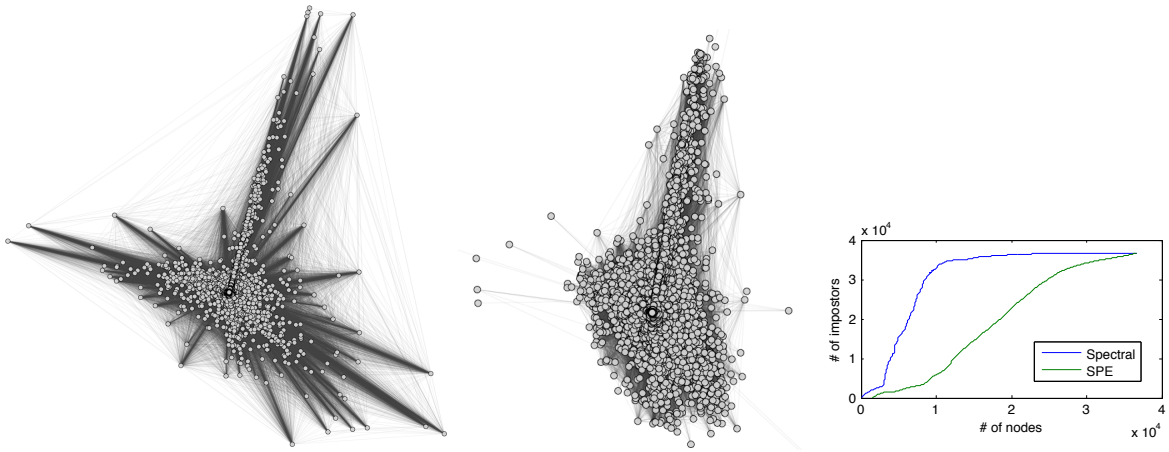
Note that we have introduced a Lagrange multiplier λ as an additional parameter which trades-off between the loss term and regularization term. We will maximize $f(\mathbf{L})$ via projected stochastic subgradient decent. Define the subgradient in terms of a single randomly chosen triplet:

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

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. $\mathbf{L}$ is initialized either randomly or from the solution of spectral embedding or Laplacian eigenmaps [1]. The



(a) Spectral embedding (b) SPE-SGD (c) Impostors

Figure 1: The Enron email network embedded into 2D by spectral embedding (a), and SPE-SGD (b). The plot on the right shows how many nodes have fewer than x impostors. We see that embedding this network into 2D yields many impostors; however on average nodes in the SPE embedding have many fewer impostors than nodes in the spectral embedding.

algorithm terminates when $|\mathbf{L}_{t+1} - \mathbf{L}_t| < \epsilon$, where ϵ is an input parameter.

In practice, instead of optimizing over a single randomly chosen triplet at each iteration, we find it useful to randomly select a node at each iteration, and use the gradient computed from all *impostor* triplets, since it is only for these triplets that a gradient step is taken. As shown in Figure 2 an impostor is a node which violates the neighborhood of another node. For each impostor triplet $\{i, j, k\}$, i is the randomly chosen target node, j is the furthest connected neighbor of i and k is a node unconnected to i but currently closer than j.

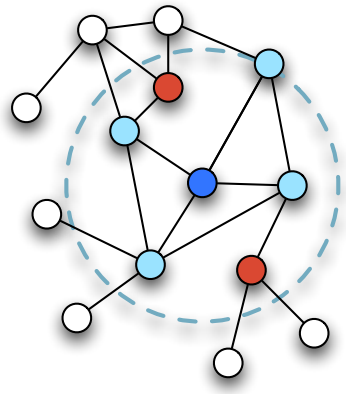


Figure 2: The red nodes are identified as impostors to the neighborhood of the center node (dark blue), because the impostors (red) are closer than the furthest of the connected nodes (light blue).

3 Experiments

In Figure 1 we see two embeddings of the Enron email network [2]. Each of the 36692 nodes in the network represents a person, and there exist edges between each pair of people who have communicated via email. Because of the high degree of many of the nodes in the network, it is likely impossible to find a 2D embedding which preserves topology exactly – meaning all nodes have zero impostors. The network may require a higher dimensional embedding. However we see that the 2D visualization produced by SPE has far fewer impostors than that produced by spectral embedding, and thus provides a more accurate visualization.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2002.

[2] J. Leskovec, J Kleinberg, and C Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005.

[3] B. Shaw and T. Jebara. Structure preserving embedding. In *Proc. of the 26th International Conference on Machine Learning*, 2009.