Graph Embedding with Global Structure Preserving Constraints

Blake Shaw – blake@cs.columbia.edu
Tony Jebara – jebara@cs.columbia.edu
Columbia University

September 4th, 2008

Abstract

We present Structure Preserving Embedding (SPE), a method for embedding graphs in low-dimensional Euclidean space such that the embedding preserves the graph's global topological properties. In particular, topology is preserved if a connectivity algorithm can recover the original graph only from the coordinates of its nodes after embedding. Given a graph and an algorithm for linking embedded nodes, SPE uses a semidefinite program to learn a low-rank kernel matrix, constrained by linear inequalities that capture the structure of the input graph. The SPE cost function ensures that the learned kernel is low-rank. The result is a small set of coordinates for each node that reproduce the original graph when processed by the connectivity algorithm (any pre-specified maximum weight subgraph algorithm). SPE improves visualization and lossless compression of graphs, outperforming popular methods such as spectral embedding and spring embedding. Many classical graphs and networks are embedded in only a few dimensions. Furthermore, the SPE constraints can be incorporated into dimensionality reduction algorithms to produce accurate embeddings of high-dimensional data.

1 Algorithm

We are given an input graph G consisting of N nodes and E edges, represented by a symmetric adjacency matrix $A \in \{0,1\}^{N^2}$ specifying which pairs of nodes are connected. The goal of SPE is to find a low-dimensional embedding of G in Euclidean space given by $\vec{y_i} \in \Re^d$ for $i = 1, \ldots, N$ such that the embedding is *structure-preserving* and accurately encodes the graph topology of G. Given a connectivity algorithm $\mathcal{A}$ for recovering a graph G *from* an embedding, (for example k-nearest neighbors and maximum weight spanning tree), a *structure preserving* embedding of a graph G is an embedding which, when processed by connectivity algorithm $\mathcal{A}$, produces exactly the graph G. SPE first learns a positive semidefinite kernel matrix K, where the input adjacency matrix A dictates a precise set of constraints on K. The embedding of the graph is then created by taking the top eigenvectors of K corresponding to the largest eigenvalues. SPE solves the following optimization to find K:

$$\max_{K \succeq 0} \ \mathrm{tr(KA_0)} - \mathrm{C}\xi \quad \text{s.t.} \ \begin{cases} \xi > 0 \\ \mathrm{tr(K)} \leq 1 \\ \sum_i \sum_j K_{ij} = 0 \end{cases}$$

For graphs that simply connect the closest nodes by some greedy algorithm $\mathcal{A}$ like k-nearest neighbors or epsilon-balls, we add the following constraints, and solve the SDP:

$$K_{ii} + K_{jj} - K_{ij} - K_{ji} > \max_m A_{im}(K_{ii} + K_{mm} - K_{im} - K_{mi}) - \xi \quad \forall_{i,j} \ \textit{when} \ A_{ij} = 0.$$

For maximum weight subgraph algorithms, such as b-matchings or maximum weight spanning trees, we iterate solving the SDP with adding cutting plane constraints until convergence:

$$(\mathrm{tr(W(K)A_0)} - \mathrm{tr(W(K)A_i)}) \geq \triangle(\mathrm{A_i}, \mathrm{A_0}) - \xi \quad \forall \mathrm{A_i} \in \mathcal{G}$$

where $\triangle(A_i, A_0) = \frac{1}{N^2}\sum_{ij}|A_i - A_0|$, $\mathcal{G}$ is a family of graphs such as b-matchings or trees, and $W(K)$ defines a weighted adjacency matrix derived from K by taking the negated distances of K. We start by running the optimization without any structure preserving constraints and then add a constraint at each iteration corresponding to the biggest violator. Given a learned kernel $\hat{K}$ from the last iteration, we find the biggest violator by computing the connectivity A_i that maximizes $\mathrm{tr(W(\hat{K})A_i)s.t.A_i} \in \mathcal{G}$ using a maximum weight subgraph method. We then add the constraint to our optimization that $(\mathrm{tr(W(K)A_0)-tr(W(K)A_i)}) \geq \triangle(\mathrm{A_i, A_0})-\xi$. The first iteration yields a rank 1 solution, which typically violates many constraints, but after several iterations, the algorithm converges when $|\mathrm{tr(WA_i)} - \mathrm{tr(WA_0)}| \leq \epsilon$, where ϵ is an input parameter,.

2 Examples

The following examples depict the usefulness of SPE and show how most of the variance (as a percentage of the eigenspectrum) in the embeddings is captured in just two dimensions (as opposed to spectral embedding methods which need many more dimensions, or spring embedding which can produce varied results due to local minima).

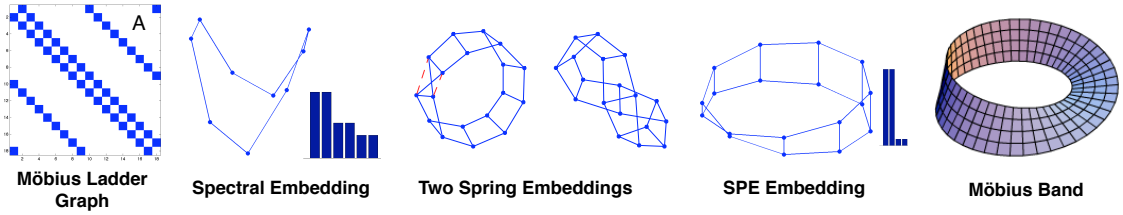

Figure 1: Embedding the classical Mobius Ladder Graph. Given the adjacency matrix (left), the visualizations produced by spectral embedding and spring embedding (middle) do not accurately capture the graph topology, furthermore spring embeddings are plagued by local minima. SPE more accurately embeds the graph in Euclidean space (right).

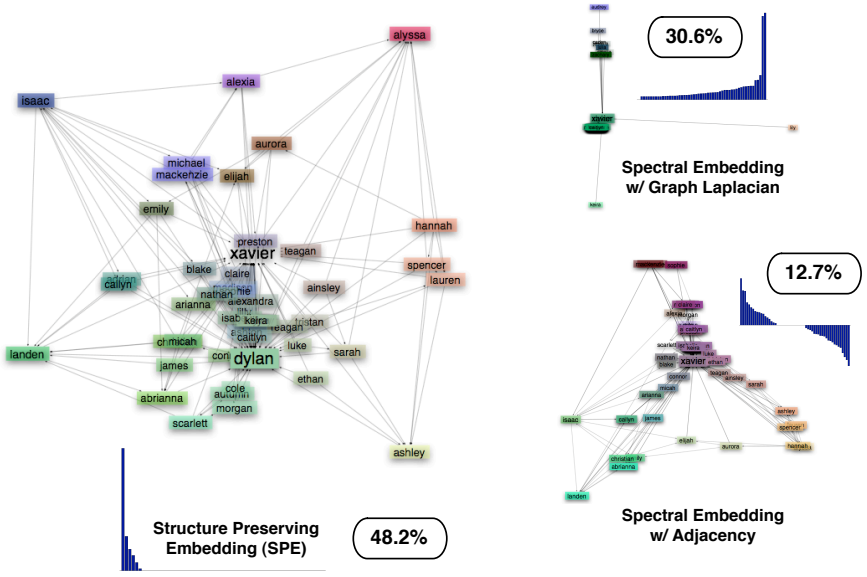

Figure 2: Visualizations of a facebook user's social network. Next to each embedding we see the corresponding eigenvalues, and the percentage of information captured in the 2D visualization.