Minimum Volume Embedding

Blake Shaw, Tony Jebara – Columbia University

September 7th, 2007

Abstract

Non-linear dimensionality algorithms are invaluable tools for working with many kinds of high-dimensional data ranging from images to biological data to social networks. Minimum Volume Embedding (MVE) has been shown to offer significant improvements over existing dimensionality reduction techniques. Following a similar approach to Semidefinite Embedding (SDE), MVE uses a semidefinite program (SDP) and matrix factorization to find a low-dimensional embedding that still accurately preserves the local distances in a high-dimensional dataset. However unlike SDE, MVE uses an intuitive objective function, explicitly maximizing the eigenspectrum energy available to the low dimensional embedding. In terms of visualization where one is typically limited to using only 2 or 3 dimensions, MVE offers significant improvements because it directly maximizes the amount and accuracy of information that can be displayed. We present here a variety of examples of using MVE on different datasets, the basic details of the algorithm, and a brief discussion of our recent work studying the convergence of the algorithm.

1 The Algorithm

Most nonlinear dimensionality reduction techniques begin with the following goal: given N points in a high-dimensional space $\vec{x_i} \in \Re^D$ for $i = 1 \dots N$ find a low-dimensional representation of corresponding points $\vec{y_i} \in \Re^d$ for $i = 1 \dots N$ such that $d \ll D$ which preserves local relationships or distances in the data. Given input data $\vec{x_i} \in \Re^D$ for $i = 1 \dots N$, MVE forms an affinity matrix A using any choice of pairwise affinity metrics such as a linear kernel, or an RBF kernel. A is then used to generate a connectivity matrix C where typically each point is connected to its k-nearest neighbors, and k is a parameter of the algorithm. This connectivity structure enforces which local distances will be preserved. MVE then learns a kernel matrix K by minimizing Equation 1, while preserving the constraints $\mathcal{K}$ that $K_{i,i} + K_{j,j} - K_{i,j} - K_{j,i} = A_{i,i} + A_{j,j} - A_{i,j} - A_{j,i} \forall_{i,j}$ where $C_{i,j} = 1$, and K must be positive semidefinite and centered. K is then used in KPCA, to get a set of eigenvectors $\vec{v_i} \in \Re^d$ for $i = 1 \dots N$, where typically the number of eigenvectors corresponding to large eigenvalues is much less than the dimensionality of the data.

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

In essence, MVE tries to unfold the data in the d dominant directions while simultaneously minimizing the variance along the other dimensions, thus preventing a key degeneracy in algorithms such as SDE which simply try to pull apart the data in all directions.

To minimize Equation 1, while preserving the constraints on K, we use an iterated SDP to minimize a variational bound on the original problem. Although we have not proven that MVE will converge to a global minimum, we show that MVE's cost is monotonically decreasing. Our recent work has been focused on studying the convergence of MVE. Figure 1 shows MVE's cost function over 10 random initializations for two different settings of the target dimensionality d. We note that for $d = 1$, MVE consistently reaches the same unique solution. Further research will be focused on finding a tight bound on the rate of convergence.

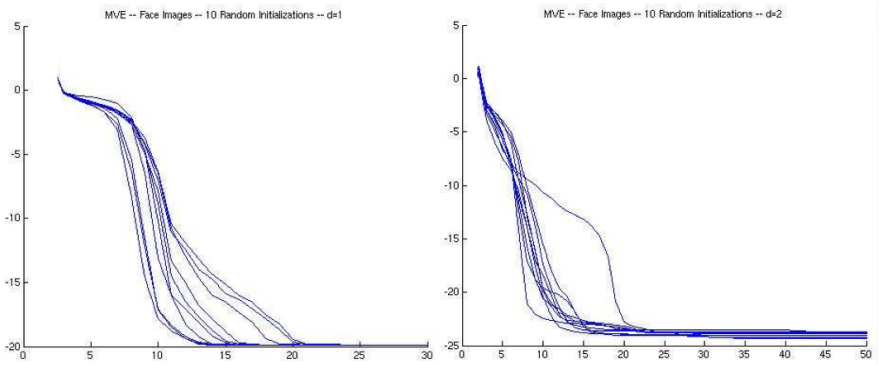

Figure 1: Plots of the convergence of MVE using different values for the target dimension parameter d

2 Examples

Figure 2 shows some examples of using MVE on real world datsets and the table below summarizes further results. Note how MVE is able to capture more of the eigenvalue energy then the other algorithms, and therefore more accurately captures local distances while using only a few dimensions. Furthermore from these plots one can see that the dimensions that MVE captures correspond to meaningful inherent features in the data.

Percentage of eigenvalue energy captured in 2D

	MVE	SDE	KPCA
Hubs and Spokes	100%	29.9%	95.0%
Spiral (% in 1D)	99.9%	99.9%	45.8%
Twos	97.8%	88.4%	18.4%
Faces	99.2%	83.6%	31.4%
Social Networks	77.5%	41.7%	29.3%

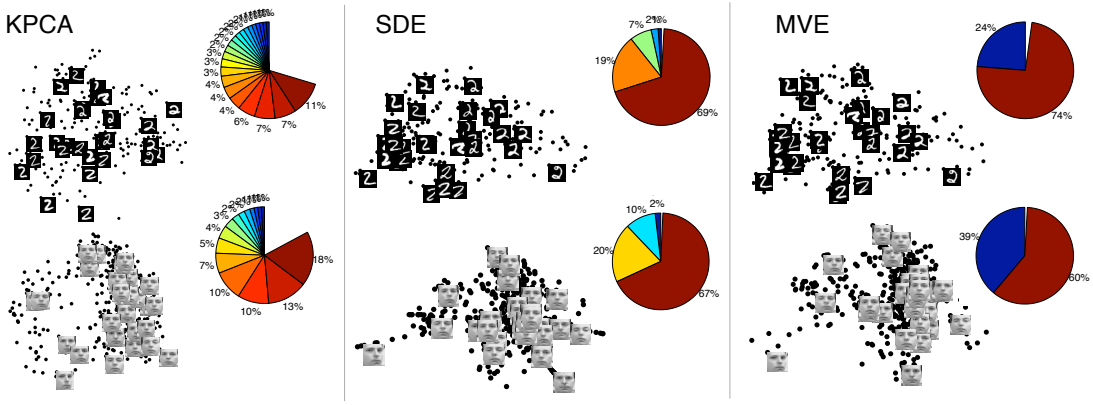

Figure 2: A comparison of the embeddings for the faces (top) and twos (bottom) image datasets. KPCA (left), SDE (middle), and MVE (right). The pie charts indicate the eigenvalue spectra corresponding to each embedding. It is clear that MVE captures more of the variance in the top 2 dimensions providing a more accurate 2D embedding.