Lecture 14, 15 and 16 Oct. 20,23 and 25 -2006

**Data Visualization**

STAT 442 / 890, CM 462

Lecture: Ali Ghodsi

# 1 Unified Framework

All of the algorithms presented so far can be cast as kernel PCA,

A straightforward connection between LLE and Kernel PCA has been shown in [2] and [6]. Let $\lambda_{max}$ be the largest eigenvalue of $L = (I - W)^T(I - W)$. Then define the $LLE$ kernel to be:

$$K_{LLE} = \lambda_{max} I - L \tag{1}$$

This kernel is, in fact, a similarity measure based on the similarity of the weights required to reconstruct two patterns in terms of $k$ neighbouring patterns. The leading eigenvector of $K_{LLE}$ is $e$, and the eigenvectors $2, \ldots, d+1$ provide the LLE embedding.

An alternative interpretation of LLE as a specific form of Kernel PCA has been discussed in [1] in details. Based on this discussion, performing Kernel PCA on pseudo-inverse $L^\dagger$ is equivalent to LLE up to scaling factors.

$$K_{LLE} = L^\dagger \tag{2}$$

It has been also shown [5] that metric MDS can be interpreted as kernel PCA. Given a distance matrix $D$, one can define $K_{MDS}$ as:

$$K_{MDS} = -\frac{1}{2}(I - ee^T)D(I - ee^T) \tag{3}$$

where $e$ is a column vector of all ones.

In the same fashion, given the geodesic distance $D^{(\mathcal{G})}$ used in Isomap, $K_{Isomap}$ can be defined as [1]:

$$K_{Isomap} = -\frac{1}{2}(I - ee^T)D^{(\mathcal{G})}(I - ee^T) \tag{4}$$

The eigenvectors of (3) and (4) yield solutions identical to MDS and Isomap, up to scaling factor $\sqrt{\lambda_p}$, where $\lambda_p$ is the $p$-th eigenvector.

The connection between kernel PCA and Semidefinite Embedding (SDE) ,that will be introduced shortly, is even more obvious. In fact, SDE is an instance of kernel PCA and the only difference is that SDE learns a kernel from data which is suitable for manifold discovery, while classical kernel PCA chose a kernel function a priori.

## 2 Semidefinite Embedding (SDE):

In 2004, Weinberger and Saul introduced semidefinite embedding (SDE) [4, 3] (See Figure 1 for an example). SDE (also known as Maximum Variance Unfolding) can be seen as a variation on kernel PCA, in which the kernel matrix is also learned from the data. This is in contrast with classical kernel PCA which chooses a kernel function a priori. To derive SDE, Weinberger and Saul formulated the problem of learning the kernel matrix as an

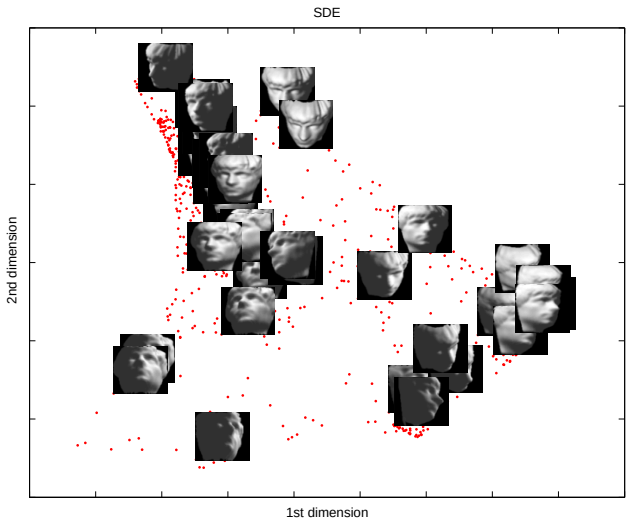


Figure 1: *SDE applied* ($k = 5$) *to the same data set. A two-dimensional projection is shown, with a sample of the original input images.*

instance of semidefinite programming. Since the kernel matrix $K$ represents inner products of vectors in a Hilbert space it must be positive semidefinite. Also the kernel should be centered, i.e. $\sum_{ij} K_{ij} = 0$. Finally, SDE imposes constraints on the kernel matrix to ensure that the distances and angles between points and their neighbours are preserved under the neighbourhood graph $\eta$. That is, if both $x_i$ and $x_j$ are neighbours (i.e. $\eta_{ij} = 1$) or are common neighbours of another input (i.e. $[\eta^T \eta]_{ij} > 0$), then the distance should be preserved

$$||\Phi(x_i) - \Phi(x_j)||^2 = ||x_i - x_j||^2.$$

It is straightforward to see:

$$
\begin{aligned}
||\Phi(x_i) - \Phi(x_j)||^2 &= \Phi(x_i)^T\Phi(x_j) - \Phi(x_j)^T\Phi(x_i) - \Phi(x_i)^T\Phi(x_i) + \Phi(x_j)^T\Phi(x_j) \\
&= K_{ii} - K_{ij} - Kji + K_{jj} \\
&= K_{ii} - 2K_{ij} + K_{jj} \\
&= ||x_i - x_j||^2
\end{aligned}
$$

Therefore, in terms of the kernel matrix, this constraint can be written as:

$$K_{ij} - 2K_{ij} + K_{jj} = ||x_i - x_j||^2.$$

By adding an objective function to maximize $\mathrm{Tr}(K)$ which represents the variance of the data points in the learned feature space, SDE constructs a semidefinite program for learning the kernel matrix $K$.

Note that the ideal objective function would be one that minimizes the rank of the matrix $K$. Unfortunately, this problem (minimizes the rank of the matrix $K$) is intractable. The alternative objective function i.e. $\mathrm{Tr}(K)$ is in fact the sum of the eigenvalues of $K$. Recall from PCA that each eigenvalue represents the variation of the data in the corresponding learned direction. Thus maximizing the sum of the eigenvalues is the same as maximizing the variance of the data points in the learned feature space.

The last detail of SDE is the construction of the neighbourhood graph $\eta_{ij}$. This graph is constructed by connecting the $k$ nearest neighbours using a similarity function over the data, $||x_i - x_j||$. In its last step, SDE runs kernel PCA on learned kernel $K$. The algorithm is summarized in Algorithm SDE (Table 1).

**Algorithm: SDE**

**Construct neighbours, $\eta$, using $k$-nearest neighbours.**

**Maximize** $\mathrm{Tr}(K)$ **subject to** $K \succeq 0$, $\sum_{ij} K_{ij} = 0$, **and**

$\forall ij \quad \eta_{ij} > 0 \vee [\eta^T \eta]_{ij} > 0 \Rightarrow$

$$K_{ii} - 2K_{ij} + K_{jj} = ||x_i - x_j||^2$$

**Run Kernel PCA with learned kernel, $K$.**

Table 1: *SDE Algorithm*

**Open Questions:** Experiments with SDE show that maximizing $Tr(K) = \sum_i \lambda_i$ lead to a low rank solution.

- Why does maximizing the variance lead to a low rank solution?
- Under what conditions do our solutions converge to the right solution?

**Strengths of SDE:**

- Eigenvalues reveal the true dimensionality of the data.
- We have a low dimensional map which preserve local isometry.

**Weaknesses of SDE:**

- SDE limited to an isometric map. The isometric property of SDE is both an advantage and a disadvantage.
- The algorithm is computationally intensive. SDE is polynomial time but the degree of the polynomial is high so it is really only feasible to work with no more than 2000

data points and no more than 6 neighbours.

# References


[1] J. Ham, D. Lee, S. Mika, and Schölkopf B. A kernel view of the dimensionality reduction of manifolds. In *International Conference on Machine Learning*, 2004.

[2] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, Massachusetts, 2002.

[3] K. Weinberger and L. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the International Conference on Machine Learning*, pages 839–846, 2004.

[4] K. Weinberger and L. Saul. Unsupervised learning of image manifolds by semidefinite programing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 988–995, 2004.

[5] C. K. I. Williams. On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19, 2002.

[6] P. Vincent Y. Bengio and J.-F. Paiement. Learning eigenfunctions of similarity: Linking spectral clustering and kernel pca. Technical Report 1232, Universite de Montreal, 2003.