

Visualizing Mercer Kernel Feature Spaces Via Kernelized Locally-Linear Embeddings

Dennis DeCoste

Machine Learning Systems Group

Jet Propulsion Laboratory / California Institute of Technology
M/S 126/347, 4800 Oak Grove Drive, Pasadena, CA 91109, USA

decoste@aig.jpl.nasa.gov

Abstract

A new technique for projecting high-dimensional data to low-dimensional spaces, called locally linear embedding (LLE), has recently been introduced. LLE offers many benefits over traditional alternatives, such as principal component analysis (PCA) and multi-dimensional scaling (MDS). In this paper, we generalize LLE to use Mercer kernels, resulting in a method we call KLLE. Mercer kernels have recently become very popular, due in large part to many recent successes in applying kernel methods such as support vector machines (SVMs) and kernel PCA to many real world problems. KLLE provides a powerful new tool for visualizing how Mercer kernels (implicitly) project data from input space to kernel feature space, which is an open and critical issue for better understanding how kernel methods work and how to best apply them.

ICONIP 2001 Session 7.11: Special session on *Support vector machines and kernel methods*.

1. Introduction

Recently, many traditional linear methods have been generalized to powerful corresponding nonlinear forms using *Mercer kernels*, including Principal Component Analysis (PCA) [12], kmeans clustering [11] nearest-neighbors [11], and Fisher discriminates [8]. Also, new methods employing kernels, including SVM classification [1] and SVM regression [13] have been introduced. Many recent applications have successfully demonstrated the power of such kernel methods (e.g. see [6]).

In this paper, we apply Mercer kernels to a very recent promising dimensionality reduction method called *locally-linear embedding* (LLE) [9], resulting

in a new kernelized form of LLE which we call KLLE.

Visualizing what various kernels do to high-dimensional data, as they project the data into even higher-dimensional feature spaces, is a critical outstanding issue in current kernel methods research [14]. KLLE enables us to easily visualize the impact of a given kernel on a given dataset. It could also provide an interesting alternative to other dimensionality reduction methods (e.g. PCA) for speeding up any machine learning method.

2. Locally-Linear Embedding (LLE)

Two popular traditional methods for projecting data into lower-dimensional spaces are multi-dimensional scaling (MDS) [2] and principal component analysis (PCA). PCA finds linear projections of greatest variance, by computing the eigenvectors of the data covariance matrix with highest eigenvalues. MDS finds projections which minimize the “stress” of violating metric pairwise distances between examples. When the distance metric is Euclidian, PCA and MDS give the same result.

Whereas PCA and MDS perform linear dimensionality reduction, the new LLE method [9] offers a means of nonlinear dimensionality reduction. Unlike other nonlinear methods, LLE is computationally very simple, involving closed-form linear algebraic operations and suffering no local minima problems.

LLE essentially tries to find local linear patches around each example on a low-dimensional manifold embedded in the high-dimensional space. When such a low-dimensional manifold exists, LLE can be very effective in discovering it.

Given ℓ -by- D data X (where ℓ is the number of examples and D is the input dimensionality), a desired embedding dimension d (typically 2 or 3 when

cally much more slower (when ℓ is much larger than D).

The *polynomial* kernel is defined by a non-linearly squashed dot product of the following form:

$$K(u, v) = (u \cdot v + r)^d, \quad (9)$$

with polynomial degree parameter d . Varying the continuous offset parameter r changes the relative weighting of the (implicit) terms in the non-linear polynomial feature space. We will refer to instances of this kernel as “POLY d r ”.

One of the most popular kernels is the *radial basis function* (RBF) kernel:

$$K(u, v) = e^{-\frac{\|u-v\|^2}{2\sigma^2}}, \quad (10)$$

with variance parameter σ , giving a different non-linear squashing of the dot product of the two examples.¹ We will refer to instances of this kernel as “RBF g ”, where $g = \frac{1}{2\sigma^2}$.

3.2 Kernel Distances

The (Euclidian) distance between examples x_i and x_j in the feature space of the kernel is, by definition:

$$d_{ij} \equiv \text{dist}(\phi(x_i), \phi(x_j)) \equiv \sqrt{\|\phi(x_i) - \phi(x_j)\|^2}. \quad (11)$$

Distances can be computed directly from kernel values:

$$d_{ij} \equiv \sqrt{K_{ii} - 2K_{ij} + K_{jj}}. \quad (12)$$

3.3 KLLE

Kernelized LLE is based on a simple idea: use kernel distance to find the nearest neighbors in the kernel’s feature space, instead of finding neighbors in the original input space (as LLE does).

To find nearest neighbors (i.e. each set $\eta(x_i)$) in time sub-linear in ℓ , one can employ various efficient indexing methods that support general metric distances. Specifically, we use vantage-point (VP) trees [16].

To find the optimal weights for reconstructing example x_i from its K neighbors, we must now compute the covariance matrix in feature space, which is:

$$\forall x_j, x_k \in \eta(x_i) \quad C_{jk} = K_{ii} - K_{ij} - K_{ik} + K_{jk}. \quad (13)$$

¹Where 2-norm defined as $\|u - v\|^2 \equiv (u \cdot u - 2u \cdot v + v \cdot v)$.

This results from replacing each term x_a in Equation 3 with $\phi(x_a)$, i.e.:

$$C_{jk} = (\phi(x_i) - \phi(x_j)) \cdot (\phi(x_i) - \phi(x_k)), \quad (14)$$

and replacing each resulting term of form $\phi(x_a) \cdot \phi(x_b)$ with the corresponding kernel value $K_{ab} = K(x_a, x_b)$.

Given these covariance matrices C_{jk} for each example x_i , we can compute the weights matrix W and the embedding matrix Y in the same way as for LLE earlier.

3.4 Semi-Supervised KLLE

In many applications, at least some of the examples have known class labels. One can take advantage of such labels when using KLLE by searching for the kernel for which the classes are best separated in the lower d -dimensional projections produced by KLLE. In our experiments, we have trained linear SVMs on the projected (2-dimensional) data for various kernels, and selected the kernel for which the resulting misclassification rate is lowest. One might further use such search to also find the best neighborhood size (K) and projection dimension (d) for a given domain.

4. Examples

To demonstrate the performance of KLLE versus LLE and PCA, we use the well-known MNIST handwritten digit data [7]. MNIST is a common benchmark dataset in SVM research, for which SVMs have recently been shown to achieve the best known test classification rates [4]. Each digit image is 28-by-28 pixels (i.e. input dimensionality is $D = 784$), with 256 grayscale levels.

Figures 1, 2, and 3 show some results in projecting MNIST data for digits “3” and “8” to two dimensions, using LLE, KLLE, and PCA respectively. For easier visualization, we did this for only the first 200 examples of each of these two digits.

The three dashed lines in each figure show the margins of a linear SVM trained on the projected two-dimensional data. Notice that KLLE provides projected data for which the linear SVM can better separate the two classes than for projected data from LLE (i.e. KLLE with a linear kernel).

Furthermore, both KLLE and LLE provide projections which are more useful for the linear SVM than that of PCA. As Figure 3 illustrates, PCA tends to

Nn=200, Np=200; KLL: kernel='poly 2 .001', K=8, secs=3.17; SVM: kernel='linear', C=2, secs=0.04, errs=33, rate=0.083

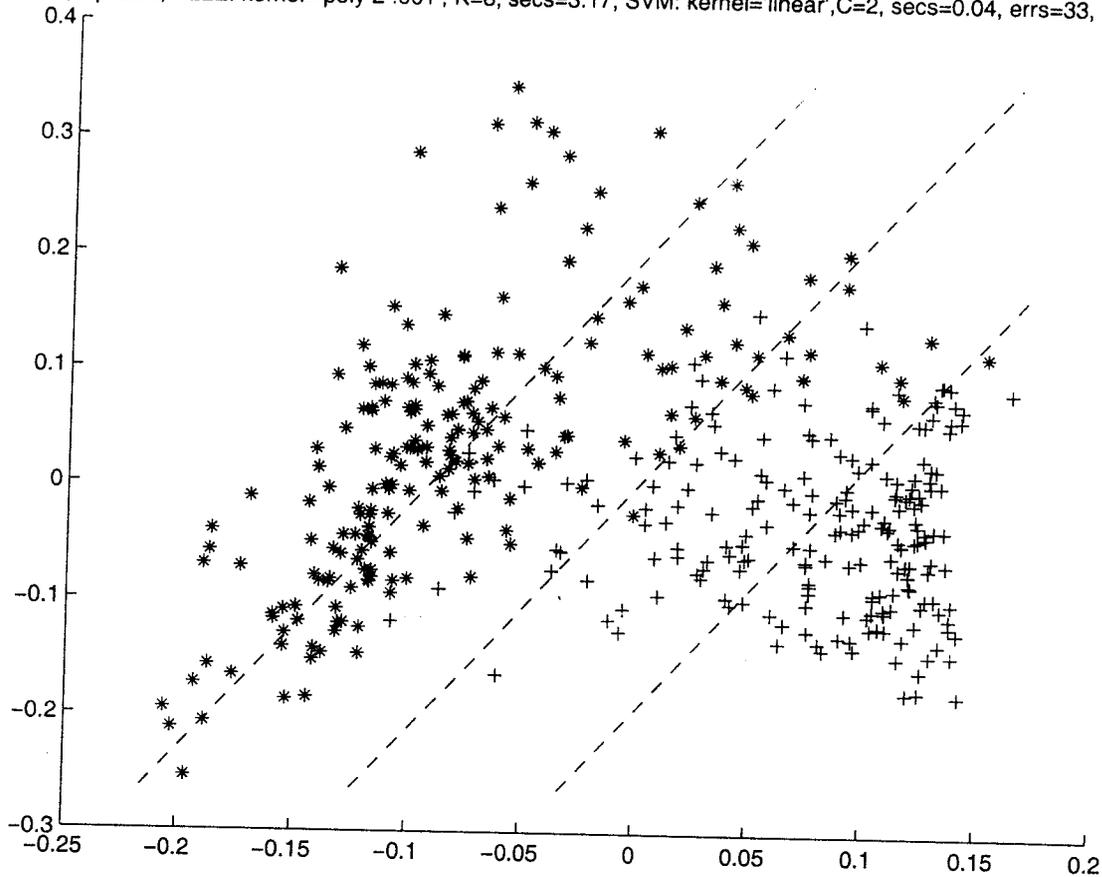


Figure 2. KLL projection ("poly 2 0.001" kernel).

Same as Figure 1, but using KLL with polynomial degree 2 kernel. The misclassification rate by a linear SVM in the projected 2-D space is 8.3%.

More experiments are required to see just how well the qualitative nature of KLL projections indeed tend to correlate with the qualitative nature of kernel methods such as SVMs, but this appears to be a very promising avenue for future work.

We are currently exploring use of KLL methods to visualize SVM and kernel performance on challenging NASA applications, such as classifying massive MISR Earth image datasets [15].

6. Acknowledgements

This research was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

- [1] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.
- [2] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994.
- [3] D. DeCoste and M. C. Burl. Distortion-invariant recognition via jittered queries. In *Computer Vision and Pattern Recognition (CVPR-2000)*, June 2000.
- [4] D. DeCoste and B. Schölkopf. Training invariance support vector machines. *Machine Learning*, 2001. In press.
- [5] D. DeCoste and K. Wagstaff. Alpha seeding for support vector machines. In *International Conference on Knowledge Discovery and Data Mining (KDD-2000)*, August 2000.