Machine Learning (CS 5350/CS 6350) *05 Nov 2009*

Manifold Learning

Often we want to find a representation of data from $\mathbb{R}^D$ in some lower-dimenaional space, $\mathbb{R}^F$, for $F \ll D$. For $F \in \{2, 3\}$, this is useful for visualization. For other F, it's useful if we believe that the data is noisy, or not ideal for our learning algorithm (eg., kNN).

There are two varieties of dimensionality reduction techniques: linear and non-linear. We've seen PCA for doing linear d.r.

Locally Linear Embedding

Locally linear embedding is a "manifold learning" algorithm. A manifold is like a F dimensional space warped to fit into a $D > F$ dimensional space. Think about a partially folded piece of paper (swiss roll).

We want to "unfold" the manifold so that it lies in its true dimensionality, F. Of course, the problem is that (a) we only have data from the manifold and (b) the data is noisy.

LLE attempts to unfold the manifold by assuming local linearity.

The algorithm works by considering each data point independently, and only in the context of its k nearest neighbors. Then, we want to be able to *reconstruct* the original data point based only on its neighbors, using a linear function. We then use these linear functions to project the data into low dimensional space.

Algorithm:

1. For each data point x_n, calculate the set S_n of its k nearest neighbors (exclusing itself).
2. For each n, compute a weight vector w_n that minimizes:

$$\left|\left|x_n - \sum_{m \in S_n} w_{nm} x_m\right|\right|^2$$

   subject to $\sum_m w_{nm} = 1$ for all n.

3. For each n, compute an embedded y_n to minimize:

$$\sum_n \left|\left|y_n - \sum_{m \in S_n} w_{nm} y_m\right|\right|^2$$

The optimal weights (from (2)) are invariant to rotations, rescalings and translations of a data point and its neighbors.

The second step is a bunch of least squares problems, one for each data point. The third step can be solved using eigen techniques. (Details in the paper.)

IsoMap

IsoMap relies on a different principle. We want our low-dimensional Euclidean distances to "match" the distances along the manifold as closely as possible. To estimate distances along the manifold, we construct a graph and compute all-pairs shortest-paths. This is then fed as input to MDS (the last step of LLE).