Lecture 17	Oct. 27 -2006

Data Visualization

STAT 442 / 890, CM 462

Lecture: Ali Ghodsi — Scribes: Stefan Pintilie

1 Landmark SDE

One of the major problems with SDE is that it requires the use of semidefinite programming which is computationally intensive. SDE is therefore limited to no more than 2000 data points which use no more than 6 neighbors. If these bounds are breached then the runtime of the algorithm becomes unreasonably long. Landmark SDE attempts to solve the problem by estimating the matrix K by first decomposing it as $K \approx QLQ^T$. Where L is a positive semidefinite matrix of size m by m and Q is a known matrix of size n by m. The idea is to pick an m that is significantly smaller than n because we want to learn the matrix L. If L is small then it will be easier to compute.

The matrix L is a subset $\{u_\alpha\}_{\alpha-1}^m$ of the entire data set $\{x_i\}_{i=1}^n$. This subset is a skeleton of the data from which we will try to reconstruct the entire data set. Each point in the data set will be a linear combination of these landmarks. We need to find a Q such that:

$$\hat{x_i} = \sum_{\alpha}^{m} Q_{i\alpha} u_\alpha$$

where $\hat{x_i}$ is the reconstruction of x_i.

The assumption is that once we have this matrix Q we can use it with the low dimensional map to reconstruct all of the data in the lower dimensional space. For instance,

$$\hat{y}_i = \sum_{\alpha}^{m} Q_{i\alpha} l_\alpha$$

Where $\{l_\alpha\}_{\alpha-1}^{m}$ is the low dimensional mapping of $\{u_\alpha\}_{\alpha-1}^{m}$.

We can write each entry in the matrix K as:

$$K_{ij} = y_i^T y_j$$

Then:

$$\hat{Y}^T \hat{Y} = Q l^T l Q^T$$

Now let $L = l^T l$ and then:

$$\hat{Y}^T \hat{Y} = Q L Q^T$$

Our job here is to find L and Q in the above equation. We begin with the derivation for Q. Recall that in LLE we can find a K-Nearest Neighbor of the graph by finding the KNN for each data point and then solving for:

$$\min_W \sum_i ||x_i - \sum_j W_{ij} x_j||^2$$

In LLE we had a matrix Φ such that:

$$\Phi = (I_n - W)^T (I_n - W)$$

In this case Φ is composed of four blocks.

$$\Phi = \begin{pmatrix} \Phi_1 & \Phi_2 \\ \Phi_2^T & \Phi_3 \end{pmatrix}$$

where Φ_1 is an m by m matrix.

Skipping the details it can be shown that:

$$Q = \begin{pmatrix} I_m \\ \\ -\Phi_3^{-1}\Phi_2^T \end{pmatrix}_{n,m}$$

Finding L is not as easy. We need to find $\max_L Tr(QLQ^T)$ subject to the constraints:

1. K is positive semi-definite: $K \succeq 0$
2. $\sum_{i,j}[QLQ^T]_{ij} = 0$
3. $[QLQ^T]_{ii} - 2[QLQ^T]_{ij} + [QLQ^T]_{jj} = ||x_i - x_j||^2$.

We do know the value of Q. We can used semi-definite programming to find L. The advantage of this approach is that L is now only m by m where m is much smaller than n. This significantly reduces the computation required to estimate K.