

Local Multidimensional Scaling

Ashutosh Tiwari - CS18BTECH11049

Polamraju Aakarsh - ES18BTECH11027

April 1, 2019

Introduction :-

Local multidimensional scaling(LMDS) is a method of non-linear dimensional reduction.

In non-linear dimensional reduction we try to construct a set of points in three-dimensional or two-dimensional space, (which can be visualized) which best represents corresponding set of points(data set) in higher dimensional space.

We are said to "project" the higher dimensional data-set onto three-dimensional or two-dimensional space.

Introduction :-

In a visualization task, every nonlinear projection method needs to make a compromise between trustworthiness and continuity. In a trustworthy projection the visualized proximities hold in the original data as well, whereas a continuous projection visualizes all proximities of the original data.

Non-Linear Dimensional Reduction :-

In non-linear dimensionality reduction given a set of points, we assume that there exists a corresponding set of points in three-dimensional or two-dimensional space such that the proximities between the points are conserved.

In essence we assume that the data of interest lie on an embedded non-linear manifold(three-dimensional or two-dimensional) within the higher-dimensional space.

Multidimensional Scaling :-

Consider a set of N points in higher dimension, which is to be projected in lower dimension. Let the distance between the i 'th and j 'th point in higher dimension space be D_{ij} .

Our goal is to construct a set of points x_k (k ranges from 1 to N) such that distance between all x_i and x_j is as close to D_{ij} as possible.

In order to do this we minimize the stress function, given by

$$S(x_1, x_2, \dots, x_N) = \sum (D_{ij} - \|x_i - x_j\|)^2$$

What we have done till here is **Classical** Multidimensional Scaling.

Local Multidimensional Scaling :-

In Local Multidimensional Scaling we modify the stress function such that we give more importance to the pairwise distances of neighbouring points, i.e. points which are closer to each other in higher dimensional space.

Let the set of neighbouring pairs of points be χ . Low weightage, w is assigned to all the pairs of point not belonging to χ . Also for all such pairs D_{ij} is replaced by D .

Here w is taken to be very small and D is taken to be very large. In doing so we assume that points which are not neighbour are very far from each other,

Local Multidimensional Scaling :-

The modified stress function now looks like this:

$$S_L(x_1, x_2, \dots, x_N) = \sum_{(i,j) \in \chi} (D_{ij} - \|x_i - x_j\|)^2 + w \sum_{(i,j) \notin \chi} (\|x_i - x_j\| - D)^2$$

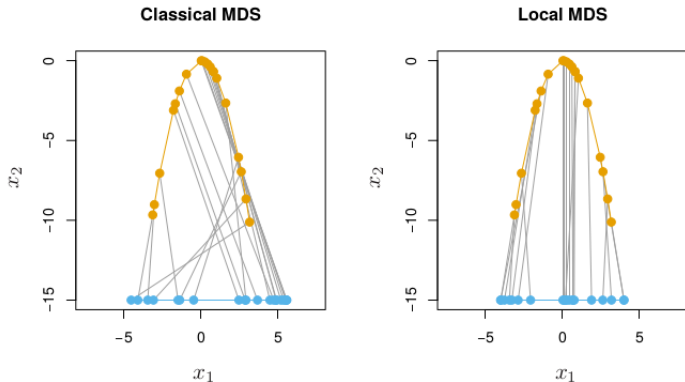
To simplify the expression, we take $w \sim 1/D$, and let $D \rightarrow \infty$.
Simplifying this gives:

$$S_L(x_1, x_2, \dots, x_N) = \sum_{(i,j) \in \chi} (D_{ij} - \|x_i - x_j\|)^2 + \tau \sum_{(i,j) \notin \chi} (\|x_i - x_j\|)$$

where $\tau = 2wD$

Local Multidimensional Scaling :-

The first term in the above equation tries to preserve local structure in the data, while the second term encourages the representations x_i, x_j for pairs (i, j) that are non-neighbors to be farther apart. Local MDS minimizes the stress function over x_i , for fixed values of the number of neighbors and the tuning parameter τ .



The orange points show data lying on a parabola, while the blue points shows multidimensional scaling representations in one dimension. Classical multidimensional scaling (left panel) does not preserve the ordering of the points along the curve, because it judges points on opposite ends of the curve to be close together. In contrast, local multidimensional scaling (right panel) does a good job of preserving the ordering of the points along the curve.

In experiments reported in Chen and Buja (2008), local MDS shows superior performance, as compared to other non dimensional reduction methods. They also demonstrate the usefulness of local MDS for graph layout. There are also close connections between the methods discussed here, spectral clustering and kernel PCA .

Thank You!