UML Project Outline

Linyun He

December 4, 2018

1 Isomap and EM on Manifolds

Outline

First determine the distance matrix through k-NN method or NNG (ϵ-nearest-neighbor). Secondly implement an adjusted EM algorithm and compare the outcome with that of a classic EM. The adjusted part is in the M-step, we assign the medoid of the data cluster to the "mean" parameter instead of sample means. Details can be found at **EMmanifold.pdf**.

Data Sets for testing

Test on a toy example has been finished. The adjusted algorithm beats the classic EM, but it's important to point out that the overall performance is hugely affected by the choice of intial points in EM algorithm. In other words, the choice of k or ϵ is the key.

Further test on Iris or Mnist. (The list may be updated.)

2 Rank-based distance extension

2.1 Rank-based distance to select initial points

First filter the data set and pick out the "important" or representative points. With a relatively big k, we set another threshold s, and define a data node important if it's in more than s other points' k-nearest neighbourhood. We can similarly set such a definition in an ϵ-NN graph.

Secondly, we can merge some near "important" node to obtain a set of clusters and "centers". Set these points as initial center guess in EM algorithm may help improve the outcome.

It's necessary to note that this paper *A rank-order distance based clustering algorithm for face tagging* provides much inspiration. A similar D^R and D^N distance can also help to choose the initial representatives and inspires us to put forward the next potential direction.

2.2 D^N-like distance and merge algorithm

Check some new distance if it can serve well to tell us whether a node is representative. Or to do some generalization of distances in other fields to this problem. To note the manifold setting would be abandoned in background as in this method.

3 Max-cut or min-cut extension

The discussion in this part still focuses on the selection of initial presentative points.

First to construct the graph by defining some new weight on each edges, while the weights reflect the similarity of two nodes. The D^R and D^N are good examples. After the construction of nearest neighbor graph, we may implement a k min cut algorithm on this graph, which provides k clusters with minimized sum of between-cluster distances. This can be interpreted as k groups with the minimized similarity. Some known approximate algorithms have help us solve the k min cut problem.

We can also choose the distance to present the differences between two data points, and thus the key part is to solve a max cut problem. Since it's NP-complete, we may not have a promising prospect, but at least this method worth a try.