Manifold Expectation Maximization Algorithm Experiments

Presented By: Xinyuan Cao, Mark Dijkstra, Linyun He, Yuze Zhou

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

COLUMBIA | ENGINEERING The Fu Foundation School of Engineering and Applied Science

Data, Models, and Code:
https://github.com/Geophagus96/Manifold-EM

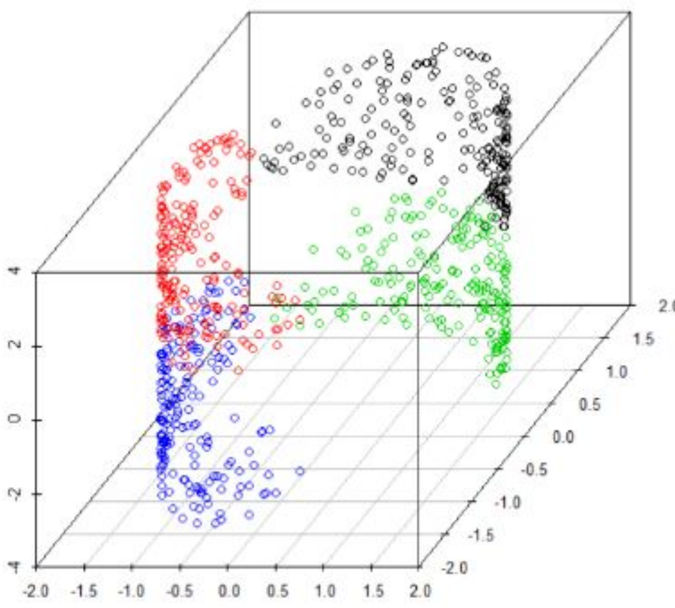
Introduction & Problem Statement

We were interested in applying a variant of an expectation maximization algorithm based on a Manifold whose points are drawn from a Gaussian distribution.

$$f_i(x) \propto \frac{1}{\sigma_i^d} e^{-\frac{d_G^2(x,\mu_i)}{2\sigma_i^2}}$$

Our algorithm is influenced by previous work in the field, and is actually a relatively novel implementation.

Our experimentation involved working extensively on a generated data set with some intrinsic gaussian distribution on some manifold in 3 dimensions. We then used a variety of expectation maximization algorithms to try and cluster the manifold in 2 dimensions.



Expectation Maximization

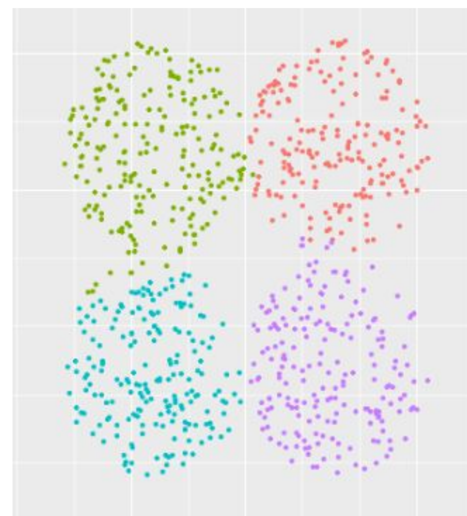


Figure 1: Manifold Data

Figure 2: Clustering of the Manifold

Our algorithm is an expectation maximization algorithm whose expectation step involves assigning clusters to the closest center with respect to the Gaussian distribution assumption, and whose Maximization step involves reweighting the parameters of the Gaussian Distribution with respect to the new cluster assignments.

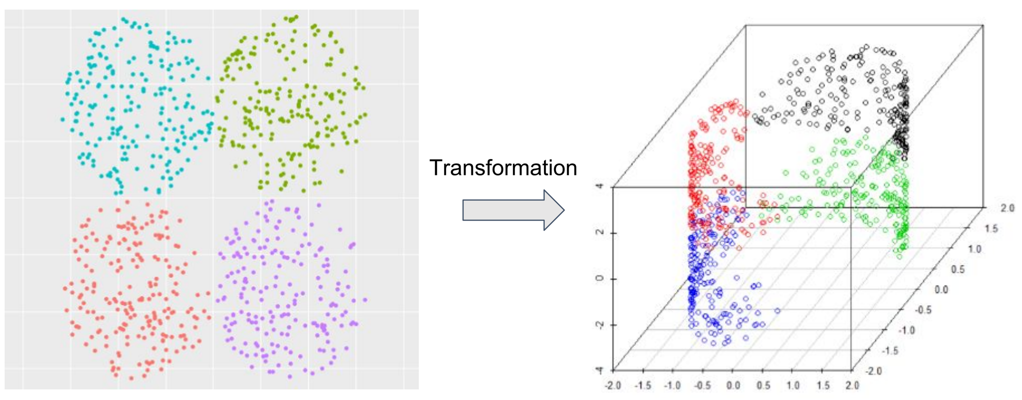
Algorithm 1 Expectation Maximization on Manifolds

procedure MYPROCEDURE
- $D_G \leftarrow ApproximateGeodesicDistance$
- $C \leftarrow MergeInitilization$
- **while** not converged **do**
  - **for** $j = 1, \ldots, n$ **do**
    - $y_j = argmax_{1 \leq i \leq k} \frac{1}{(\sigma_i^2)^{\frac{d}{2}}} e^{-\frac{d_G^2(x_j,\mu_i)}{2\sigma_i^2}}$
  - **for** $i = 1, \ldots, k$ **do**
    - $\pi_i = \frac{|C_i|}{\sum_{j=1}^k |C_j|}$
    - $\mu_i = argmin_{x \in C_i} \sum_{x_j \in C_i} d_G^2(x, x_j)$
    - $\sigma_i^2 = \frac{\sum_{x_j \in C_i} d_G^2(x_j \mu_i)}{d|C_i|}$

Experiments

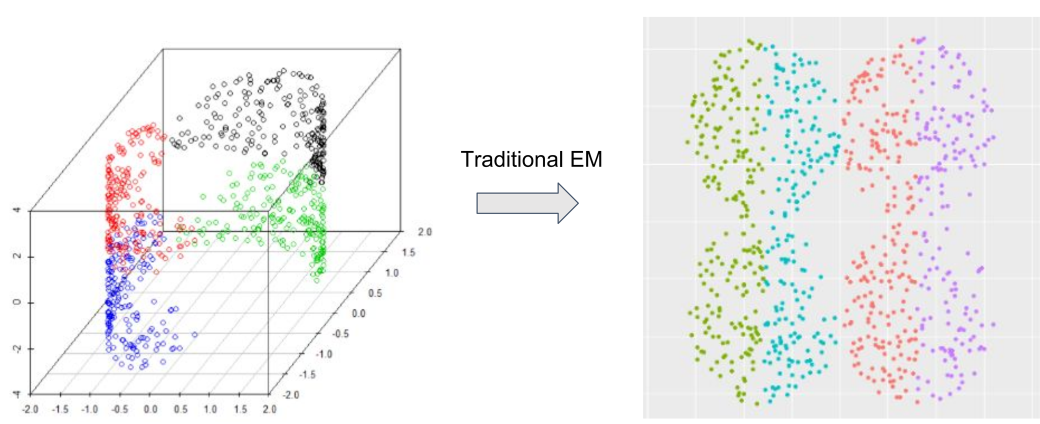
Setup

Our toy data set was generated based on a 2 dimensional set of points already partitioned into four clusters. We then apply a transformation with some added gaussian noise to "fold" the 2 dimensional data into the manifold in 3-dimensions.



Motivation

Running the traditional expectation maximization algorithm on this manifold dataset resulted in some visually unappealing clusters. This suggested that there must be another way to be able to extract clusters in a manner that produced clusters that looked visually similar to the ones that the manifold data was transformed from



Our initial implementation of our expectation maximization algorithm involved a random initialization of the clusters centers. This resulted in very varied clusterings, some of which proved to be quite note and others, not quite. However, the indication that our EM algorithm was capable of producing clusterings that looked similar to the dataset from which the manifold was transformed from was indicative of our algorithm's validity (see figure 2)..

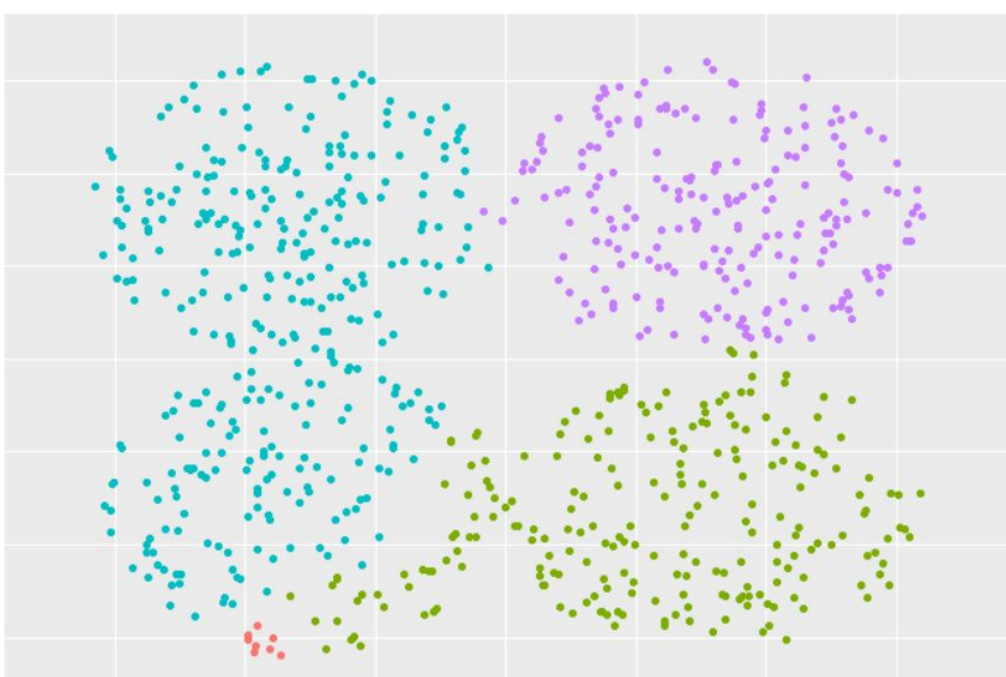


Figure 3: Arbitrarily Bad Clustering by our EM Algorithm

In addition to the standard expectation maximization algorithm and our expectation maximization algorithm we also tried spectral clustering from an R package to see how other algorithms fare off in the clustering problem.

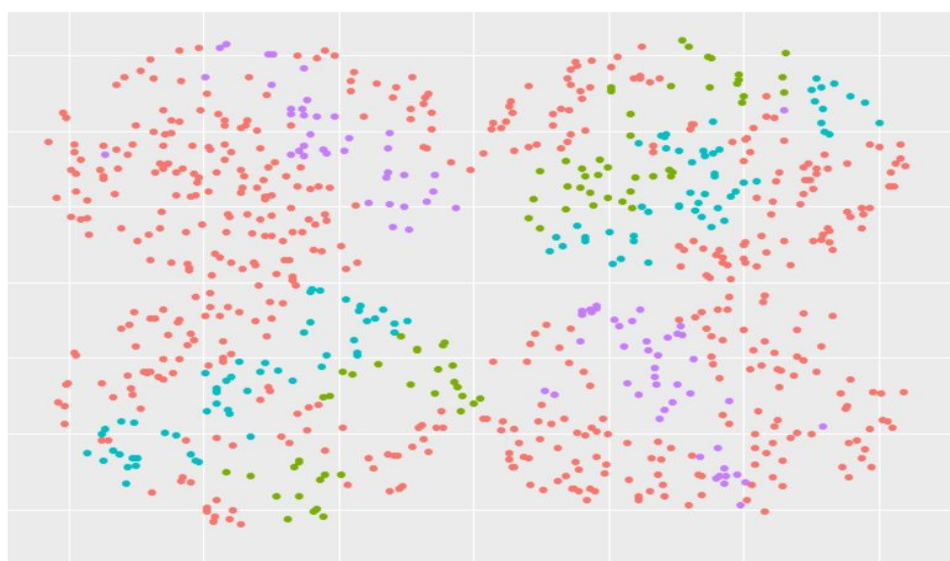


Figure 4: Arbitrarily Bad Clustering by Spectral Clustering

Initialization Algorithms

Since the initialization of the clusters prior to running EM significantly affected the resultant clusters. As a result we implemented three nearest neighbor merge algorithms to deal with the cluster initialization problem and these implementations significantly improved the output clusters.

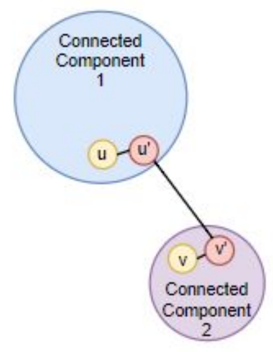
One such nearest neighbor merge at a high level works as such:

1. Compute Nearest Neighbor Graph and shortest path distances
2. Compute distance between connected components by

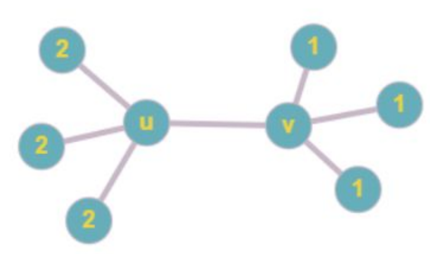
   $$d_G(u,v) = d_G(u,u') + d_x(u',v') + d_G(v,v')$$

   [Souvenir, Richard "Manifold Clustering"]

   this is demonstrated by figure below



3. Select "important" points that have more neighbors than a threshold
4. Find (u,v) if d(u,v) = min D_G (Important Points)
5. Take the sum of the weights of the edges between all neighbors of u and all neighbors of v and then remove the vertex from the set of important points that has the larger sum [Zhu, Chunhui]



6. Repeat 4 and 5 until only k important points, these are initial k clusters

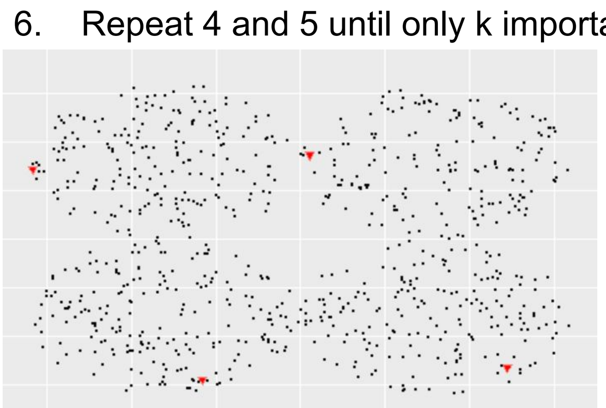


Figure 5: Locations of initial cluster centers given by Min-Nearest Neighbor Merge initialization

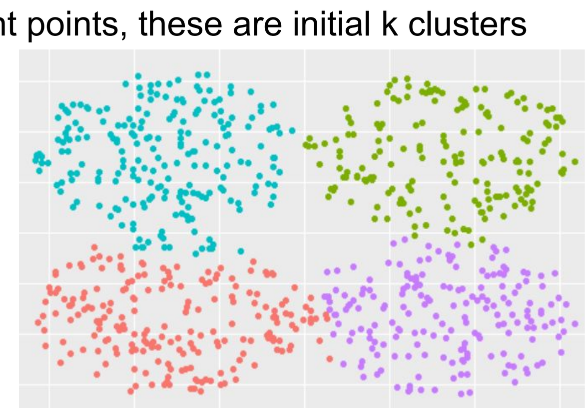


Figure 6: Resultant clustering given figure 4's initial cluster points

Numerical Results

The metric we used to calculate the accuracy of the clusters with respect to the initial data set in 2-dimensions from which we constructed the 3 dimensional manifold and the resultant clustering data based on the various algorithms implemented was the purity evaluation measure:

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

Where Ω is the set of clusters (our output) and C is the set of classes (the original points clusters that we transform into the manifold) and w_k is a cluster in Ω and c_j is a class of C. ["Evaluation of Clustering"]

Algorithm	Purity Measure
Traditional Expectation Maximization	0.53
Spectral Clustering	0.36
Min NN Merge EM	0.996

Discussion

Our algorithm has problems working with data matrices of low rank and high sparsity. Specifically, our algorithm has trouble working on the MNIST dataset for its high dimensionality and the reasons listed just prior. Furthermore, our algorithm sometimes returns fewer clusters than asked for due to numeric properties of the matrix we cannot control, including stationary points that exist within the dataset we can accidentally converge upon. This problem is also witnessed in the traditional expectation maximization algorithm.

Conclusion

We acknowledge that further research and experimentation with our algorithm is necessary to have more generalizable results. Our algorithm does phenomenally with respect to the purity measure; however, such a high value can also indicate some bias or overfitting, as we tried many different initialization algorithms to achieve such favorable results. Nonetheless, we believe we have implemented an algorithm such that it combines interesting features of Gaussian Mixture Modeling, Expectation Maximization, and Manifold learning that justifies further research and experimentation.

References

Archambeau, Cédric, John Aldo Lee, and Michel Verleysen. "On Convergence Problems of the EM Algorithm for Finite Gaussian Mixtures." ESANN. Vol. 3. 2003.

Souvenir, Richard, and Robert Pless. "Manifold clustering." Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Vol. 1. IEEE, 2005.

Stanford University, "Evaluation of Clustering" https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html?fbclid=IwAR37nxYYAxWw8VGrC8TVqShhyfvAJxjqwyoqa8UwgfwuuIxiqI1JSbP4Tc8

Wu, Yiming, and Kap Luk Chan. "An extended isomap algorithm for learning multi-class manifold." Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on. Vol. 6. IEEE, 2004.

Zhu, Chunhui, Fang Wen, and Jian Sun. "A rank-order distance based clustering algorithm for face tagging." Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011