Clustering Seminar

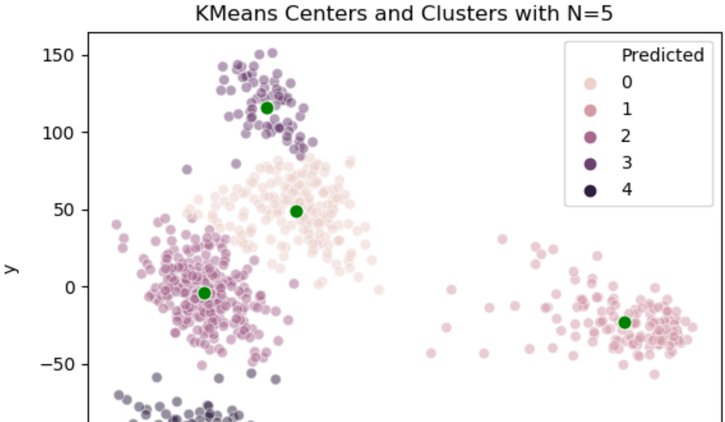
Organizational Meeting

September 10, 2019

Approaches to clustering

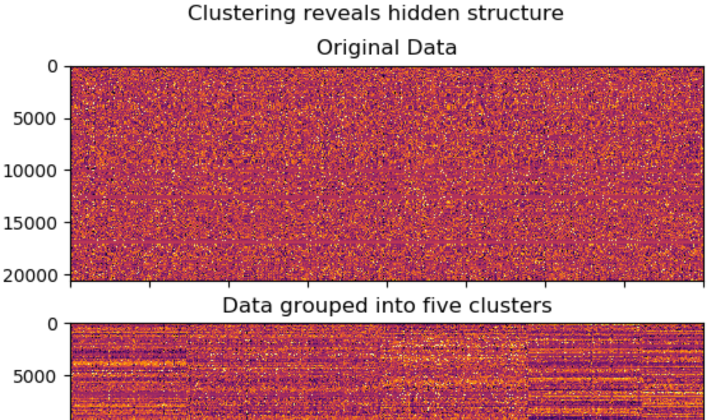
k-means

In general, pick *k* random points in the space. Take the points in the dataset closest to each point and compute their center of mass. Replace the *k* points by these centers of mass and repeat the process.



Hierarchical clustering

Define a distance measure between sets of points in the space. Start with every cluster being a single point. Choose the closest clusters and combine them. Repeat until there is only one cluster. The sequence of joinings gives a tree that describes the hierarchical clustering.



Graph communities

Louvain clustering: An iterative method that tries to maximize a clustering measure called ‘modularity.’ If the nodes of a graph are partitioned into clusters, then the modularity is essentially the ratio of the edges that lie entirely within the clusters over the expected number of such edges if the edges of the graph were rearranged at random in a certain sense. Louvain clustering is a type of hierarchical clustering that iteratively increases this measure.

Data can be converted to a graph by connected points to their nearest neighbors, according to some metric.

Mixtures

One assumes that the data arose from a "mixture distribution" which is a sum of probability distributions. For example, one could assume that points in the plane arose from a sum of n gaussian distributions. A maximum likelihood computation will give you the means and variances of these distributions; and then each point can be assigned to the distribution that was most likely to have produced it.

Dimensionality Reduction

Clustering algorithms typically begin with a dimensionality reduction.

- *Principal component analysis* Projects the data into lower dimensional subspace spanned by the directions where the variation is maximal.
- *Spectral embedding* For a graph, projects the nodes into coordinates given by most significant eigenvalues of the laplacian
- *non-linear, or manifold methods such as tSNE* tSNE converts the data into a graph, and the graph into a probability distribution and then tries to model that distribution in low-dimensional space with the least distortion.

Example

As a demonstration of the 10x genomics single cell sequencing platform, 1.3M brain cells were sequenced from two mice. The output of the experiment is a matrix with 1 row for each cell and 1 column for each gene; the entries count the number of RNA molecules transcribed from that gene in that cell. There are about 30K genes. So we have a matrix of integers that is 1.3M by 30k entries.

The RNA expression data characterizes the operational state of the cell.

Clustering result

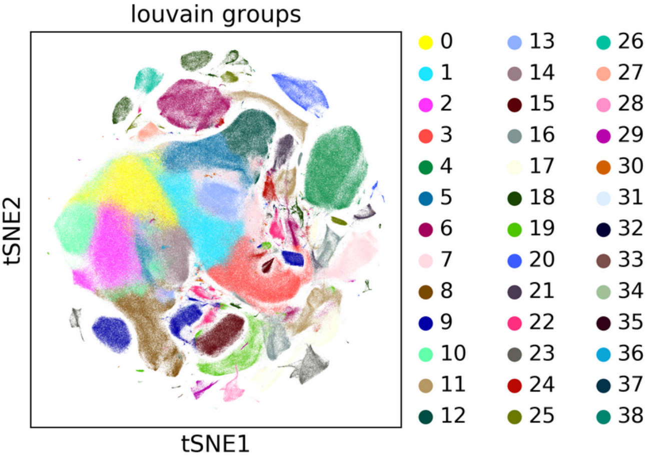


Figure 3: tSNE dimension reduction applied to RNA seq data

Another example

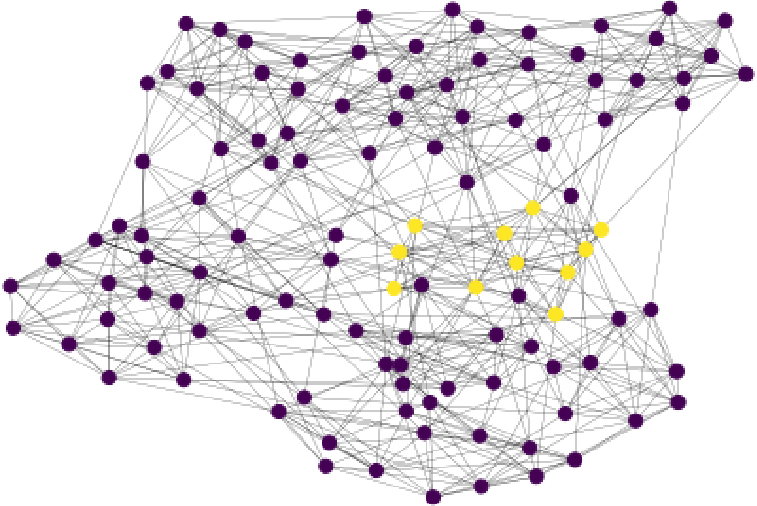


Figure 4: Nodes are football teams, edges are games, yellow='Big Ten'

Finite metric spaces

There are theoretical results on clustering obtained by viewing the problem as one of finite metric spaces.

- Kleinberg's theorem: There is no clustering algorithm that is symmetric, scale invariant, and consistent.

  In particular, Take a finite set X. A similarity function is a function $d : X \times X \to \mathbf{R}$ that is symmetric and zero only on the diagonal.

  A clustering method F is a function from the set of such D to the partitions of X. We say that F is scale invariant if $F(\alpha d)$ gives the same cluster as F for any $\alpha > 0$; we say that F is rich if it is surjective onto the set of partitions; and we say that F is consistent if it agrees on two distance functions d and d' whenever d' is smaller than d for two points in the same cluster, and d' is bigger than d for two points in different clusters. There is no clustering function that is scale invariant, rich, and consistent.

Some thoughts on theoretical questions

What kind of definitive theoretical results about these various approaches exist?

In the broadest terms, suppose one of these approaches says a cluster exists. Can one say something definitive about the original data?

For an algorithm like tSNE, how specifically can one reconstruct information about the original data from the fact that points are placed close together?

Can one compare the “clusters” that arise from different methods in some way?