Lecture Notes on ML

付星宇. 中山大学数学学院.

2018.1.22 广东·广州

Section 1: Clustering

- K-Means
- Hierarchical Clustering
- DBSCAN

• K-Means

In this ~~section~~ part, we first introduce the Basic-K-Means Algorithm, then we use minimization of objective function to illustrate why K-Means works. Finally, we give a short introduction about Bisect-K_Means to solve the problem of randomness of initial centroids.

▶ Basic_K_Means.

1: Let user specify how many clusters to form, say, k.
2: Select k points randomly as intial centroids.
3: do
4: Form k clusters by ~~giving~~ assigning each point to its closest centroid. (why?)
5: Recompute the centroid of each cluster. (why?)
6: until Centroids do not change.

The remaining problem is: how to define "closest"? how to "recompute" centroid? and why we do step 4 an 5?

▶ Minimization of objective function.

First of all, we need to define distance between two data points, i.e. dis $(\vec{x}, \vec{y})$, where $\vec{x}$ and $\vec{y} \in R^{n}$.

Some typical distance measures are L2 Measure and L1 Mease, where

$dis_{(L_2)}(\vec{x}, \vec{y}) = \sum_{i=1}^{n} (x_i - y_i)^2$, $dis_{(L_1)}(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_i|$. (Now you know how to define "closest")

The goal of k_Means is to minimize some user specified objective function, here we give two examples:

<u>Example 1</u>

Say we have m sample points $\{\vec{x}_i\}_{i=1}^{m} \subseteq R^n$, and we want to find k clusters of $\{\vec{x}_i\}_{i=1}^{m}$. The measure is L2 and the objective function is:

$$SSE = \sum_{j=1}^{k} \left(\sum_{i=1}^{m_j} dis_{(L_2)}(\vec{x}_i, c_j) \right)$$

where m_j denotes the number of sample points in the jth Clusters and c_j is the centroid of the jth cluster, clearly $\sum_{j=1}^{k} m_j = m$.

As for the step 4 of Basic_k_Means Algorithm, the centroids $\{c_j\}_{j=1}^{k}$ are given and we assign x_i to different centroids to minimize SSE. Clearly assign x_i to the closest centroid is an optimal choice.

As for the step 5 of Basic-k-Means, the clusters are given, i.e. each point has been assigned to a cluster and we now try to relocate $\{c_j\}_{j=1}^{k}$ to minimize SSE further, i.e.

$$\frac{\partial}{\partial c_j} SSE = \frac{\partial}{\partial c_j} \sum_{t=1}^{k} \sum_{i=1}^{m_t} (x_i - c_t)^2$$

$$= \sum_{t=1}^{K} \sum_{i=1}^{m_t} \frac{\partial}{\partial c_j} (x_i - c_t)^2 = 2 \sum_{i=1}^{m_j} (x_i - c_j) = 0.$$

Hence we have $\Rightarrow$ $\boxed{c_j = \frac{1}{m_j} \sum_{i=1}^{m_j} x_i}$ ← how to recompute the centroid. $(\forall j)$. when L2 and SSE.

Example 2.

Say L1 Measure and objective function is

$$SAE = \sum_{j=1}^{K} \sum_{i=1}^{m_j} |x_i - c_j|$$

Step 4 is exactly the same as example 1.

As for Step 5:

$$\frac{\partial}{\partial c_j} SAE = \frac{\partial}{\partial c_j} \sum_{t=1}^{k} \sum_{i=1}^{m_t} |x_i - c_t|$$

L1 and SAE

$$= \sum_{i=1}^{m_j} \text{sign}(x_i - c_j) = 0 \quad (\forall j)$$

hence $\boxed{c_j \text{ is the median}}$ of the jth clusters. $(\forall j)$

The above the examples show how the recomputation of c_j differs when measure and objective function changes.

▶ Bisect_K-Means

1: Initialize the list of clusters to only contain one cluster consisting of all point.

2: do

3: Select a cluster from the list of clusters. (how?) Say largest size, say largest SSE.

4: for i = 1 : number of trial

5: Bisect (i.e. k=2) the selected cluster using Basic-K-Means.

6: end for.

7: Select the two clusters from all the bisections with lowest SSE or SAE. and add them into the list of clusters (the original selected one is certainly removed).

8: Until the size of the list is k.

9: Refine the result by using their centroids as the intial centroid and run a Basic_k-Means on them. (Necessary! since bisecting is not optimal).

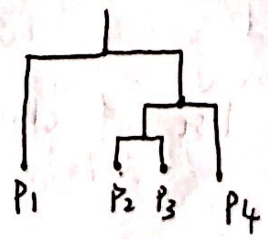
By doing this strategy, we can conquer the weakness of random initial points.

- Hierarchy Clustering

Hierarchy clustering ~~the~~ techniques are a second important Category of clustering method. Although these approaches are relatively old compared to many clustering algorithms, yet they still enjoy widespread use.

A hierarchical clustering is often displayed graphically using a tree-like diagram called a dendrogram (系统树图), which displays both the cluster-subcluster relationships and the order in which the clusters were merged.

Example:



Here is the general descriptions how the algorithm works:

Starting with individual points as singleton clusters, then successiv merge the two closest clusters until only one cluster remains.

The key operation of the Algorithm is the computation of proximit between two clusters and it is the definition of different cluster proximity that differentiates various Hierarchy clustering techniques. ~~thus~~ Now we are going to introduce several different calculation methods:

- Single Link : $D(C_1, C_2) \triangleq \min_{\substack{x_1 \in C_1 \\ x_2 \in C_2}} D(x_1, x_2)$

- Complete link : $D(C_1, C_2) \triangleq \max_{\substack{x_1 \in C_1 \\ x_2 \in C_2}} D(x_1, x_2)$.

- Average Link : $D(C_1, C_2) \triangleq \frac{1}{|C_1|} \cdot \frac{1}{|C_2|} \cdot \sum_{x_1 \in C_1} \sum_{x_2 \in C_2} D(x_1, x_2)$

- Controids : $D(C_1, C_2) \triangleq D\left(\frac{1}{|C_1|} \sum_{\vec{x}_1 \in C_1} \vec{x}_1 , \frac{1}{|C_2|} \sum_{\vec{x}_2 \in C_2} \vec{x}_2 \right)$.

- Ward's Method : $D(C_1, C_2) \triangleq$ Increase in SSE After the Merging.

- DBSCAN

DBSCAN is the abbreviation of Density-Based Spatial Clustering of Applications with Noise, which locates regions of high density that are seperated from each other by regions of Low density.

In the DBSCAN setting, all points are classified into 3 types, i.e.:

- Core point

A point is called core point if the number of points within a given neighborhood around the point exceeds a certain threshold. (The Radius of neighboorhood and the threshold are both user specified).

- Border Point

A point which is not a core point while falls within the neighboorhood of a core point.

- Noise Point

A noise point is any point that is neither a core point nor a border point.

With the above definition, we can now introduce DBSCAN formally:

▶ DBSCAN

1: Label all points as Core, border, and noise points.

2: Eliminate all noise points.

3: Link an edge between all core points that are within Eps (user specified) of each other

4: Make each group of connected core points into a separate cluster

5: Assign each border point to one of the clusters of its associated core points.