

8주차 - Clustering

- unsupervised learning

비지도 학습은 라벨이 없는 데이터에 대해서 구조를 분석하는 학습 알고리즘이다.

- clustering

군집화는 비지도 학습의 일종으로 데이터의 구조를 밝힐 때 데이터 안에서 비슷하게 묶을 수 있는 덩어리를 찾아낸다.



K-Means Algorithm

군집화 알고리즘 중 가장 유명한 방법으로 2가지 작업을 반복함으로써 수렴하는 K개의 군집을 찾아낸다.

input은 데이터 set과 나누고자 하는 군집 수이다.

K-means algorithm

Input:

- K (number of clusters) 
- Training set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ 

$$\underline{x^{(i)} \in \mathbb{R}^n} \text{ (drop } \underline{x_0 = 1} \text{ convention)}$$

2가지 작업은 아래 사진과 같다.

1. cluster assignment step

initial cluster centroid(초기 군집 중심)에 대하여 각 데이터가 어느 군집에 가까운지 체크한다.

1. 각 군집에 가까운 점들의 무게중심을 구하여 새로운 해당 군집의 중심이 되게 한다.

K-means algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

Repeat {

Cluster assignment step: for $i = 1$ to m
 $c^{(i)} := \text{index (from 1 to } K \text{) of cluster centroid closest to } x^{(i)}$
 $\min_k \|x^{(i)} - \mu_k\|^2$

Move centroid: for $k = 1$ to K
 $\mu_k := \text{average (mean) of points assigned to cluster } k$
 $\rightarrow \mu_2 = \frac{1}{4} [x^{(1)} + x^{(5)} + x^{(6)} + x^{(10)}] \in \mathbb{R}^n$
 $\rightarrow c^{(1)}=2, c^{(5)}=2, c^{(6)}=2, c^{(10)}=2$

한편, 어떤 군집은 가까운 데이터가 하나도 없을 수 있는데 이런 경우 그 군집을 없애거나 그 군집 중심의 위치를 다시 randomize하는 방법이 있다.

• Optimization Algorithm

k-means 군집화 알고리즘도 cost function을 정의하고 그것을 minimize하는 파라미터를 구하는데 이를 distortion cost function, distortion of the k-means algorithm이라고도 부른다.

비용함수는 아래 그림과 같이 군집의 중심과 각점이 배정된 군집의 index를 파라미터로 가진다. 그리고 모든 점에 대해서 그 점이 속한 군집 중심과의 거리의 제곱의 평균으로 정의된다.

K-means optimization objective

$\rightarrow c^{(i)}$ = index of cluster $\{1, 2, \dots, K\}$ to which example $x^{(i)}$ is currently assigned

$\rightarrow \mu_k$ = cluster centroid k ($\mu_k \in \mathbb{R}^n$)

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned
 $x^{(1)} \rightarrow 5, c^{(1)}=5, \mu_{c^{(1)}} = \mu_5$

Optimization objective:

$$\rightarrow J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

$\min_{c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K} J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$

$\rightarrow \mu_1, \dots, \mu_K$ Distortion

한편, k-means 군집화 알고리즘은 2가지 작업을 반복적으로 한다고 했다.

2가지 작업은 각각 2개의 파라미터 그룹을 갱신하는 작업인데, 비용함수를 최소화 하는 과정도 2가지 과정으로 나누어져 있다. 먼저 군집 중심은 고정시킨채 데이터에 가까운 군집을 비용함수를 최소화하는 방향으로 구하고, 이후에는 각 데이터에 속한 군집을 고정시킨채 군집의 중심을 비용함수를 최소화하는 방향으로 구한다.

K-means algorithm

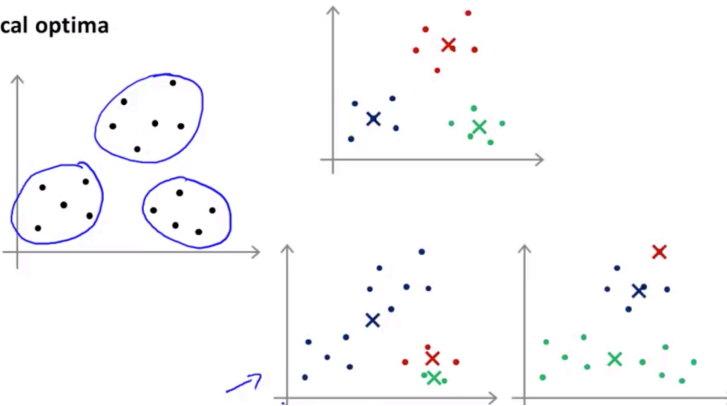
Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

Repeat {
Cluster assignment step
Minimize $J(\dots)$ wrt $c^{(1)}, c^{(2)}, \dots, c^{(n)}$
(hold μ_1, \dots, μ_K fixed)
for $i = 1$ to m
 $c^{(i)} := \text{index (from 1 to } K \text{) of cluster centroid}$
 $\text{closest to } x^{(i)}$
for $k = 1$ to K
 $\mu_k := \text{average (mean) of points assigned to cluster } k$
}
move centroid
Minimize $J(\dots)$ wrt μ_1, \dots, μ_K
}

• Random initialization

초기 군집 중심을 random하게 설정하는 이유는 초기 군집 중심에 따라 비용함수를 최소화 시키는 군집 중심의 수렴 값이 제각각이기 때문이다. 이런 Local optima 문제를 어느정도 해소하기 위한 개념이다.

Local optima



군집은 상식적으로 data set size보다 작아야한다. ($K < m$)

여기서 1~m개의 데이터 중 K개를 무작위로 뽑은 뒤에 각 군집의 초기 중심으로 잡는다.

그리고 local optima를 피하기 위해서 위 과정을 100~1000회 반복한 뒤,

그중에서 최종 비용함수의 값이 가장 작은 케이스를 선택한다.

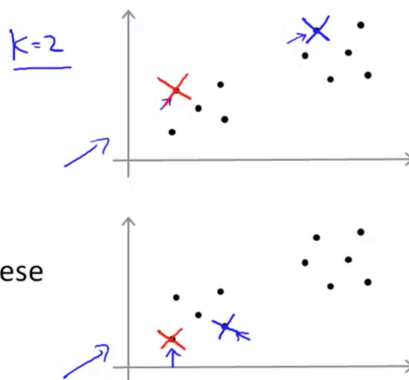
Random initialization

Should have $K < m$

Randomly pick K training examples.

Set μ_1, \dots, μ_K equal to these K examples.

$$\begin{aligned} \mu_1 &= x^{(i)} \\ \mu_2 &= x^{(j)} \\ &\vdots \end{aligned}$$



이런 무작위 반복법은 K가 작을 때(2~10) 가장 유용하고 K가 클때는 그렇게 큰 차이를 만들지는 못한다.

하지만 어느 경우나 local optima를 피하기 위한 유용한 방법이다.

- Choosing the Number of Clusters

cluster의 개수는 일반적으로 사람의 직관을 통해서 정한다.

물론 K 를 자동으로 구하는 방법도 있는데,

Elbow method는 automatic하게 cluster의 개수를 구하는 방법이다.

K 에 대한 비용함수를 그래프로 그려서 기울기가 급격하게 변하는 곳(Elbow)의 k 값을 개수로 삼으면 된다.

하지만 우측하단 그래프처럼 Elbow가 잘 나타나지 않는 경우가 대부분이다.

따라서 목적에 따라서 적당한 cluster 개수를 정하는 것이 가장 일반적이다.

Choosing the value of K

Elbow method:

