Lecture 10 Mar 25 2021

Clustering (Unsupervised) //

K-means Algorithm //

- Given N data points, we want to assign K clusters.

- We want to define binary matrix $\in \mathbb{R}^{N \times k} = L$
  that indicates which cluster the ith data point belongs to
  - The sum of each row is 1.

- We want to find vector C_j for each cluster, $j \in 1, \dots, k$

K-means //

In K-means, define C_j as mean of datapoints assigned to cluster j. (Center of cluster)

1) Initialize K C_j (centers)
2) Assign each data point to the closest center
3) Update C_js to be the mean of the data points to that cluster
4) Repeat step 2 until C_j doesn't change (convergence)

This algorithm is gaurenteed to converge but we can only gaurentee a local optimum.
Bad initialization = Bad result.

The algorithm tries to minimize objective:

$$E(\{c_j\}_{j=1}^{N}, L) = \sum_{i,j} L_{ij} \|y_i - c_j\|^2$$

which is the squared euclidem distance between data point y_i and its assigned cluster c_j.

Problems with optimization //
We need combinatorial optimization since L is a discrete variable.

- We can't use gradient descent since L's not continuous
- There are K^n configurations to optimize

Problems with initialization//

1) Random label = cluster centers are close to the mean of the data set

2) Random centers = bad centers will result in centers having no data points

3) Data point as center = not the best, but okay

4) Mutiple restarts, run multiple times and pick assignment with best score = expensive

5) K-means ++ : Choose centers that are farthest away from each other.

   1) Choose random data point as 1st center

   2) Compute distance between each data point and the closest center

   3) Choose next center from the remaing data points with greatest distance

   4) Repeat 2) until there are k-centers.

K-Means bad with overlapping data or non-circular data

Gaussian Mixture Models. (GMM or MoG)
- Good for overlapping data / ellipsoidal data.

- Generalization of K-means

K-means: sphrical data and hard assignment. $(0,1)$s

GMM: ellipsoidal data and soft assignment. $[0,1]$.

- GMM is a generative model that you can use to model multimodal data.

- GMM is a linear combination of K-Gaussians each with it's own mean and variance. Each component has it's prior m_j such that $\sum_{j}^{K} m_j = 1$

Mean: μ_j
Covariance Matrix: C_j $\Big\}$ θ_j

Formally we can write the likelihood of y_i as $p(y \mid \theta)$

$$p(y|\theta) = \sum_{j}^{K} p(y,j|\theta)$$

$$= \sum_{j}^{K} p(y|\theta,j)\, p(j|\theta)$$

$$= \sum_{j}^{K} m_j\, p(y|\theta,j)$$

Here j is the index of the gaussian

that generates y.

$$= \sum_{j}^{K} m_j \, G(y; \theta_j)$$

We want to learn $\{\theta_j, \}_{j=1}^{K}$ to maximize $p(y|\theta)$

How to generate data given θ?

1) Sample jth Gaussian using $p(j|\theta) = [m_1, \ldots, m_K]$

2) Take jth Gassian with highest m_j and sample data point from it.

How to learn parameters (θ)?

We want to find mle of $p(y|\theta)$:

$$\begin{aligned} L(\theta) &= -\ln p(y_{1:N}|\theta) \\ &= -\sum_{i}^{N} \ln(p(y_i|\theta)) \\ &= -\sum_{i}^{N} \ln \sum_{j}^{K} p(y_i|\theta_{ij}) p(j|\theta)) \\ &= -\sum_{i}^{N} \ln\left[\sum_{j}^{K} (G(y_i; \theta_j) m_j)\right] \end{aligned}$$

Because of the log of sums, a closed form solution DNE.

1) Use gradient descent, but

1) $\sum_j \pi_j = 1$

2) C_j needs to be PD.

We must use constrained optimization.

2) Use Expectation Maximization (EM) algorithm

EM lets you find missing data by maximizing likelihood.

1) E-step

- Compute the probalistic labeling variable γ_{ij} (the owner probabilities) corresponding to the probability of y_i coming from cluster j

- compute $p(j \mid y_i, \theta) = \frac{p(y_i \mid j, \theta)\, p(j \mid \theta)}{\sum_k p(y_i \mid k, \theta)\, p(k \mid \theta)}$

$p(j \mid y_i, \theta) = \gamma_{ij}$

Bayes Rule + Law of total P.

2) M-step

- Compute π_j and θ_j.

$$\pi_j = \sum_i \frac{\gamma_{ij}}{N}$$

$$\mu_j = \left(\sum_i \gamma_{ij} y_i\right) / \sum_i (\gamma_{ij})$$

$$C_j = \left(\sum_i \gamma_{ij} (y_i - \mu_j)(y_i - \mu_j)^T\right) / \sum_i (\gamma_{ij})$$

Pseudocode:

Initialize γ and θ

Loop

For each data point i

$\gamma_{ij} = p(j | y_i | \theta)$

For each cluster j

update θ

Terminate based on 1) datapoint assignments
or 2) likelihood

K-means vs GMM.

GMM & K-means when

- $\pi_j = \frac{1}{k}$
- $C_j = \sigma^2 I$ (sphere)
- σ^2 small (Hard labeling)