Lecture 12: Clustering and EM

wbg231

December 2022

1 unsupervised learning

- the goal is to discover unknown structure in the data
- we try to estimate densities with some latent variable θ

$$P(x|\theta)$$

k means

- dataset $\mathcal{D} = (x_1 \cdots x_n) \subset X$ where $X \in \mathbb{R}^d$
- the goal is to partition D into k disjoint subsets $C_1 \cdots C_k$
- let $c_i \in \{1 \cdots k\}$ be the cluster assignment of data point x_i
- the centroid of the data C_i is defined as

$$\mu_i = argmin_{\mu \in X} |||x - \mu||^2$$

so that is each centroid is the mean of it's cluster

- the objective is

$$j(c, \mu) = \sum_{i=1}^{n} ||x_i \mu_c||^2$$

- the k means algorithm is described here
  1. Initialize: Randomly choose initial centroids $\mu_1, \dots, \mu_k \in \mathrm{R}^d$.
  2. Repeat until convergence (i.e. c_i doesn't change anymore):
     1. For all i, set

        $$c_i \leftarrow \arg\min_j \|x_i - \mu_j\|^2. \qquad \text{Minimize } J \text{ w.r.t. } c \text{ while fixing } \mu$$

     2. For all j, set

        $$\mu_j \leftarrow \frac{1}{|C_j|} \sum_{x \in C_j} x. \qquad \text{Minimze } J \text{ w.r.t. } \mu \text{ while fixing } c.$$

  - Recall the objective: $J(c, \mu) = \sum_{i=1}^{n} \|x_i - \mu_{c_i}\|^2$.

- so there is an alternative behavior of picking the best cluster for each data point, and picking the best centroid for each cluster
- the objective of k means is non convex, so it can get stuck in bad local minima pretty easily
- can re run it multiple times to try to avoid this

gaussian mixture models

- a generative model for X, done with MLE
- assume there are k sets up and we have the probability distribution of each
- the generative story of a GMM is as follows
  1. chose a cluster $z \sim catagorical(\pi_1 \cdots \pi_k)$
  2. chose a conditional distribution for that cluster $x|z \sim \mathcal{N}(\mu_z, \Sigma_z)$
- then we can get the marginal likelihood of our dataset by marginalizing over the latent variable z

$$P(x) = \sum_z P(x, y) = \sum_z p(x|z)P(z) = \sum_k \pi_k \mathcal{N}(\mu_k, \Sigma_k)$$

- note that in GMMs the label of the cluster is not important
- how do we learn the parameters μ_k, π_k, Σ_K
- we can do mle

$$L(\theta) = \sum_{i=1}^{n} logP(x_i|\theta) = \sum_{i=1}^{n} log(\sum_z P(x, z|\theta))$$

  note that our class label and data points are connected so we can not just push log into the sum
- there is no closed form solution for GMM
- so gradient descent is kind of involved
- if we had cluster assignments mle would be easy
- we observe x and want to know z.

$$P(z = j|x_i) = \frac{P(x, z = j)}{p(x)} = \frac{P(x|z = j)P(z = j)}{\sum_k P(x|z = k)P(z = k)} = \frac{\pi_i \mathcal{N}(x_i|\mu_j, \Sigma_j)}{\sum_k \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}$$

- think of $P(z|x)$ as a soft class assignment
- if we knew μ, Σ, π that would be easy to compute

expectation max for GMM

- Let's compute the cluster assignments and the parameters iteratively.

  The expectation-minimization (EM) algorithm:

  1. Initialize parameters μ, Σ, π randomly.
  2. Run until convergence:
     1. E-step: fill in latent variables by inference.
        - compute soft assignments $p(z \mid x_i)$ for all i.
     2. M-step: standard MLE for μ, Σ, π given "observed" variables.
        - Equivalent to MLE in the observable case on data weighted by $p(z \mid x_i)$.

- so we estimate using expectation maximization in this method we first intilize the parameters μ, Σ, π randomly
- then alternate between teh E and M step until convergence
- where the E step i gill in latent variables by inference (compute the soft class assignments $P(z|x_i)\forall i$)
- M step: standard MLE for μ, Σ, π given our soft assignments. this is equivalent to mle in observable case on data weighed by $P(z|x_i)$

M step

- let $P(Z|x)$ be the soft assigned

ft assignments:

$$\gamma_i^j = \frac{\pi_j^{\text{old}} \mathcal{N}\left(x_i \mid \mu_j^{\text{old}}, \Sigma_j^{\text{old}}\right)}{\sum_{c=1}^k \pi_c^{\text{old}} \mathcal{N}\left(x_i \mid \mu_c^{\text{old}}, \Sigma_c^{\text{old}}\right)}.$$

- $$\begin{aligned} n_z &= \sum_{i=1}^n \gamma_i^z \\ \mu_z^{\text{new}} &= \frac{1}{n_z} \sum_{i=1}^n \gamma_i^z x_i \\ \Sigma_z^{\text{new}} &= \frac{1}{n_z} \sum_{i=1}^n \gamma_i^z \left(x_i - \mu_z^{\text{new}}\right)\left(x_i - \mu_z^{\text{new}}\right)^T \\ \pi_z^{\text{new}} &= \frac{n_z}{n}. \end{aligned}$$

em for GMM summary

- em is a genearl algorithm for leanring latent vairble mdoels
- key dea is that if the data was fully observable MLE would be easy
- E step fill in latent vaibles by comuting $P(z|x, \theta)$
- M step standard MLE given fully observable data
- this si simpler and more efficent than graidnt methods
- k means is a special case of EM for GMM wil hard assignments

latent variable models

generative latent vairble models

- two sets of random variables Z, X
- z is hidden unobserved variables
- x is observed variables

- joint probability model is parametrized by $\theta \in \Theta$

$$P(x, z|\theta)$$

- a latent variable model is a probability model for which certain varibles are never observed
- x alone is incomplete data
- (x,z) is complete data

objectives

- learning probelm given incomplete data find the mle

$$\hat{\theta} = argmax_{\theta} P(x|\theta)$$

- the inference problem is

$$P(z|x, \theta)$$

- there are cases where learning and inference are both hard

EM algorithm

- at slide 88