Recitation 13

Kmean, GMM and EM

Colin

Spring 2022

Apr 27

Motivation

- We are now moving away from supervised learning to unsupervised learning (no labels)
- The goal of modeling is no longer prediction/classification but discovering underlying pattern
- To understand how are data generated, what characteristic does the generation process have
- Formally, either $p(x)$ or $p(z|x)$
    - Learning/Inference problems

Outline

- Start by discussing clustering algorithms
  - Hard clustering: K-means
  - Soft clustering: GMM
- Move into EM algorithms and how the clustering problems are related

Clustering Algorithm: K-mean

- Very intuitive to understand
- Start by randomly defining centroids
  - Compute clusters
    - Classify points based on those centroids by distance
  - Update centroids
    - Update centroids based on classified points by average
- Notice how the two steps depend on each other's result to proceed.
- Each update step is independent of the other.
- Notice the update steps are hard classifications
  - One points is either class 1 or class 2.
  - The centroid updates only consider points of its class

Kmean/GMM

Generalization

- To slightly generalize the procedure
- Start by randomly defining centroids
  - Compute soft clusters
    - Assign weights to points to each centroid
  - Update centroids
    - Update centroids based on the weighted points
- This is the 'softer' version of K-means
  - Instead of 0-1 label to points, its a sequence of weights
  - Each centroid update considers all the points

Generalization

- The 'weight' mentioned in GMM is essentially a distribution over each centroid.
- We can generalize it to some distribution $q(z)$, and update its parameters as we train the model.

Generalization

- To further generalize the procedure
- Start by randomly defining centroids, define a distribution, $q(z)$ (with parameter λ)
  - Compute soft clusters
    - Assign weights to points to each centroids, which is equivalent to
    - Update $q_i(z)$ (or update λ_i).
  - Update centroids
    - Update centroids based on the weighted points (or update θ)
- This is essentially the GMM algorithm

Clustering Algorithm: GMM

- Start by randomly defining centroids (μ_k, Σ_k) and $q_i(z)$ to be $Ber(p_1, p_2, ..., p_n)$
  - Compute soft clusters
    - Assign weights to points based on those centroids and $q_i(z)$
    - $q_i(z) = \gamma_i^k = \frac{\pi_k^{old}\mathcal{N}\left(x_i|\mu_k^{old},\Sigma_k^{old}\right)}{\sum_{c=1}^{k}\pi_c^{old}\mathcal{N}\left(x_i|\mu_c^{old},\Sigma_c^{old}\right)}$
  - Update centroids
    - Update centroids based on the weighted points
    - $\mu_k = \frac{1}{\sum_i^n \gamma_i^k}\sum_i^n \gamma_i^k x_i$
    - $\Sigma_k = \frac{1}{\sum_i^n \gamma_i^k}\sum_i^n \gamma_i^k(\mu_k - x_i)^T(\mu_k - x_i)$

Extension to EM

- Now we have the whole setup, we can switch out terms specific to GMM
  - Start by randomly defining parameters θ, define $q(z)$
  - Define loss function
  - $L(\theta, \lambda) = \sum_i -KL(q_i(z|\lambda)||p(z|x_i, \theta)) + logp(x_i|\theta)$
    - Optimize $q_i(z)$ — λ
    - Optimize $p(x_i|z)$ — θ
- This is the EM algorithm
- By some computation, we know that the optimal $q_i^*(z)$ is $p(z|x_i)$
  - Therefore the E-step is sometimes in closed form solution
- But the optimization over θ may not be trivial.

EM

- Therefore, the rationale behind EM algorithm is basically
  - Introduce $q(z)$ to divide the problem
  - Solve the problem by coordinate descent
    - Sequential updates
- The idea is a part of variations inference which is popular in both traditional statistics and deep learning
  - e.g. **variational auto-encoder (VAE)**

Summary

KL Divergence

- It is a "metric" to measure the difference between two distributions
- It is **not symmetric**, hence not a actual metric!
- Originated from information theory, but widely used in deep learning