Clustering and Gaussian Mixture Models

Piyush Rai
IIT Kanpur

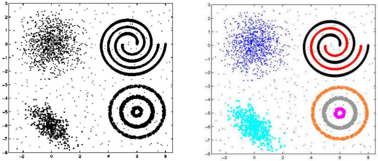
Probabilistic Machine Learning (CS772A)

Jan 25, 2016

Recap of last lecture..

Clustering

- Usually an unsupervised learning problem
- Given: N unlabeled examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; the number of partitions K
- Goal: Group the examples into K partitions



(a) Input data (b) Desired clustering

- Clustering groups examples based of their mutual similarities
- A good clustering is one that achieves:
  - High within-cluster similarity
  - Low inter-cluster similarity
- Examples: K-means, Spectral Clustering, Gaussian Mixture Model, etc.

Picture courtesy: "Data Clustering: 50 Years Beyond K-Means", A.K. Jain (2008)

Refresher: K-means Clustering

- **Input:** N examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; $\boldsymbol{x}_n \in \mathbb{R}^D$; the number of partitions K
- **Initialize:** K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, $\boldsymbol{\mu}_k \in \mathbb{R}^D$; many ways to initialize:
  - Usually initialized randomly, but good initialization is crucial; many smarter initialization heuristics exist (e.g., K-means++, Arthur & Vassilvitskii, 2007)
- **Iterate:**
  - (Re)-Assign each example $\boldsymbol{x}_n$ to its closest cluster center

$$\boxed{\mathcal{C}_k = \{n : \quad k = \arg\min_k ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2\}}$$

  ($\mathcal{C}_k$ is the set of examples assigned to cluster k with center $\boldsymbol{\mu}_k$)

  - Update the cluster means

$$\boxed{\mu_k = \text{mean}(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k|} \sum_{n \in \mathcal{C}_k} \boldsymbol{x}_n}$$

  - Repeat while not converged
- A possible convergence criteria: cluster means do not change anymore

The K-means Objective Function

- Notation: Size K one-hot vector to denote membership of $\boldsymbol{x}_n$ to cluster k

$$\boldsymbol{z}_n = \underbrace{[0 \;\; 0 \;\; \ldots \;\; 1 \;\; 0 \;\; 0]}_{\text{all zeros except the k-th bit}}$$

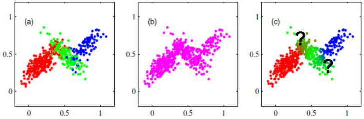
- Also equivalent to just saying $\boldsymbol{z}_n = k$
- K-means objective can be written in terms of the total distortion

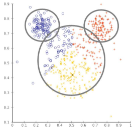
$$J(\boldsymbol{\mu}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- Distortion: Loss suffered on assigning points $\{\boldsymbol{x}_n\}_{n=1}^{N}$ to clusters $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$
- Goal: To minimize the objective w.r.t. $\boldsymbol{\mu}$ and $\mathbf{Z}$
- Note: Non-convex objective. Also, exact optimization is NP-hard
- The K-means algorithm is a heuristic; alternates b/w minimizing J w.r.t. $\boldsymbol{\mu}$ and $\mathbf{Z}$; converges to a local minima

K-means: Some Limitations

- Makes hard assignments of points to clusters
  - A point either totally belongs to a cluster or not at all
  - No notion of a soft/fractional assignment (i.e., probability of being assigned to each cluster: say $K = 3$ and for some point $\boldsymbol{x}_n$, $p_1 = 0.7, p_2 = 0.2, p_3 = 0.1$)
- K-means often doesn't work when clusters are not round shaped, and/or may overlap, and/or are unequal





- **Gaussian Mixture Model:** A probabilistic approach to clustering (and density estimation) addressing many of these problems

Mixture Models

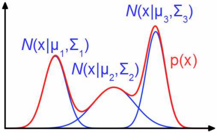
- Data distribution $p(x)$ assumed to be a weighted sum of K distributions

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\boldsymbol{\theta}_k)$$

where π_k's are the mixing weights: $\sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0$ (intuitively, π_k is the proportion of data generated by the k-th distribution)

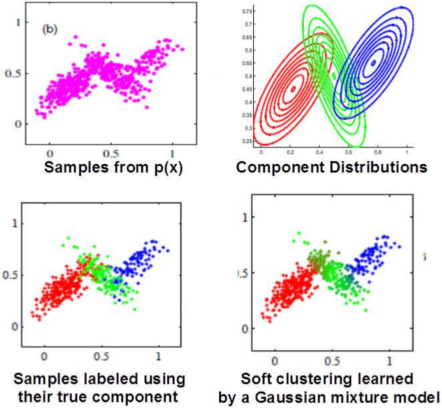
- Each component distribution $p(\boldsymbol{x}|\boldsymbol{\theta}_k)$ represents a "cluster" in the data
- **Gaussian Mixture Model (GMM):** component distributions are Gaussians

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



- Mixture models used in many data modeling problems, e.g.,
  - Unsupervised Learning: Clustering (+density estimation)
  - Supervised Learning: Mixture of Experts models

GMM Clustering: Pictorially



Notice the “mixed” colored points in the overlapping regions

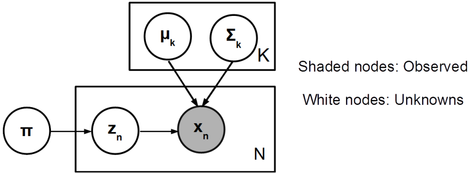
GMM as a Generative Model of Data

- Can think of the data $\{\boldsymbol{x}_1, \boldsymbol{x}_n, \ldots, \boldsymbol{x}_N\}$ using a "generative story"
    - For each example $\boldsymbol{x}_n$, first choose its cluster assignment $\boldsymbol{z}_n \in \{1, 2, \ldots, K\}$ as

$$\boldsymbol{z}_n \sim \text{Multinoulli}(\pi_1, \pi_2, \ldots, \pi_K)$$

    - Now generate $\boldsymbol{x}$ from the Gaussian with id $\boldsymbol{z}_n$

$$\boldsymbol{x}_n | \boldsymbol{z}_n \sim \mathcal{N}(\boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma}_{z_n})$$



- Note: $p(z_{nk} = 1) = \pi_k$ is the prior probability of $\boldsymbol{x}_n$ going to cluster k and

$$p(\boldsymbol{z}_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$$

GMM as a Generative Model of Data

- Joint distribution of data and cluster assignments

$$p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$$

- Marginal distribution of data

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(z_k = 1)p(\boldsymbol{x}|z_k = 1) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

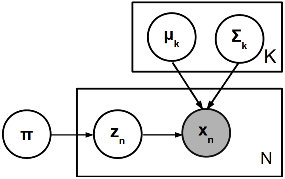
- Thus the generative model leads to exactly the same $p(\boldsymbol{x})$ that we defined

Learning GMM

- Given N observations $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ drawn from mixture distribution $p(\boldsymbol{x})$

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Learning the GMM involves the following:
  - Learning the cluster assignments $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$
  - Estimating the mixing weights $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ and the parameters $\theta = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ of each of the K Gaussians



- GMM, being probabilistic, allows learning probabilities of cluster assignments

GMM: Learning Cluster Assignment Probabilities

- For now, assume $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ and $\theta = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ are known
- Given θ, the posterior probabilities of cluster assignments, using Bayes rule

$$\gamma_{nk} = p(z_{nk} = 1|\boldsymbol{x}_n) = \frac{p(z_{nk} = 1)p(\boldsymbol{x}_n|z_{nk} = 1)}{\sum_{j=1}^K p(z_{nj} = 1)p(\boldsymbol{x}_n|z_{nj} = 1)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- Here γ_{nk} denotes the posterior probability that $\boldsymbol{x}_n$ belongs to cluster k
- Posterior prob. $\gamma_{nk} \propto$ prior probability π_k **times** likelihood $\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- Note that unlike K-means, there is a non-zero posterior probability of $\boldsymbol{x}_n$ belonging to each of the K clusters (i.e., probabilistic/soft clustering)
- Therefore for each example $\boldsymbol{x}_n$, we have a vector $\boldsymbol{\gamma}_n$ of cluster probabilities

$$\boldsymbol{\gamma}_n = [\gamma_{n1} \quad \gamma_{n2} \quad \ldots \quad \gamma_{nK}], \quad \sum_{k=1}^K \gamma_{nk} = 1, \gamma_{nk} > 0$$

GMM: Estimating Parameters

- Now assume the cluster probabilities $\gamma_1, \ldots, \gamma_N$ are known
- Let us write down the log-likelihood of the model

$$\mathcal{L} = \log p(\mathbf{X}) = \log \prod_{n=1}^{N} p(\boldsymbol{x}_n) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

- Taking derivative w.r.t. $\boldsymbol{\mu}_k$ (done on black board) and setting to zero

$$\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma_{nk}} \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

- Plugging and chugging, we get

$$\boxed{\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma_{nk}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n}$$

- Thus mean of k-th Gaussian is the weighted empirical mean of all examples
- $N_k = \sum_{n=1}^{N} \gamma_{nk}$: "effective" num. of examples assigned to k-th Gaussian (note that each example belongs to each Gaussian, but "partially")

GMM: Estimating Parameters

- Doing the same, this time w.r.t. the covariance matrix $\mathbf{\Sigma}_k$ of k-th Gaussian:

$$\mathbf{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma_{nk}(\boldsymbol{x}_n - \mu_k)(\boldsymbol{x}_n - \mu_k)^\top$$

.. using similar computations as MLE of the covariance matrix of a single Gaussian (shown on board)

- Thus $\mathbf{\Sigma}_k$ is the weighted empirical covariance of all examples
- Finally, the MLE objective for estimating $\pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$

$$\sum_{n=1}^{N}\log\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k) + \lambda(\sum_{k=1}^{K}\pi_k - 1) \qquad (\lambda \text{ is the Lagrange multiplier for } \sum_{k=1}^{K}\pi_k = 1)$$

- Taking derivative w.r.t. π_k and setting it to zero gives Lagrange multiplier $\lambda = -N$. Plugging it back and chugging, we get

$$\pi_k = \frac{N_k}{N}$$

which makes intuitive sense (fraction of examples assigned to cluster k)

Summary of GMM Estimation

- Initialize parameters $\theta = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ and mixing weights $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$, and alternate between the following steps until convergence:
  - Given current estimates of $\theta = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ and $\boldsymbol{\pi}$
    - Estimate the posterior probabilities of cluster assignments

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \qquad \forall n, k$$

  - Given the current estimates of cluster assignment probabilities $\{\gamma_{nk}\}$
    - Estimate the mean of each Gaussian

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma_{nk} \boldsymbol{x}_n \qquad \forall k, \text{ where } N_k = \sum_{n=1}^N \gamma_{nk}$$

    - Estimate the covariance matrix of each Gaussian

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma_{nk} (\boldsymbol{x}_n - \mu_k)(\boldsymbol{x}_n - \mu_k)^\top \qquad \forall k$$

    - Estimate the mixing proportion of each Gaussian

$$\pi_k = \frac{N_k}{N} \qquad \forall k$$

K-means: A Special Case of GMM

- Assume the covariance matrix of each Gaussian to be spherical

$$\boldsymbol{\Sigma}_k = \sigma^2 \mathbf{I}$$

- Consider the posterior probabilities of cluster assignments

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_k \exp\{-\frac{1}{2\sigma^2}||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2\}}{\sum_{j=1}^{K} \pi_j \exp\{-\frac{1}{2\sigma^2}||\boldsymbol{x}_n - \boldsymbol{\mu}_j||^2\}}$$

- As $\sigma^2 \to 0$, the summation of denominator will be dominated by the term with the smallest $||\boldsymbol{x}_n - \boldsymbol{\mu}_j||^2$. For that j,

$$\gamma_{nj} \approx \frac{\pi_j \exp\{-\frac{1}{2\sigma^2}||\boldsymbol{x}_n - \boldsymbol{\mu}_j||^2\}}{\pi_j \exp\{-\frac{1}{2\sigma^2}||\boldsymbol{x}_n - \boldsymbol{\mu}_j||^2\}} = 1$$

- For $\ell \neq j$, $\gamma_{n\ell} \approx 0 \Rightarrow$ hard assignment with $\gamma_{nj} \approx 1$ for a single cluster j
- Thus, for $\boldsymbol{\Sigma}_k = \sigma^2 \mathbf{I}$ (spherical) and $\sigma^2 \to 0$, GMM reduces to K-means

Next class: The Expectation Maximization Algorithm