Introduction to Machine Learning

Mixture Models

Varun Chandola

Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY, USA
chandola@buffalo.edu

Outline

Latent Variable Models

- Consider a probability distribution parameterized by $\boldsymbol{\theta}$
- Generates samples ($\mathbf{x}$) with probability $p(\mathbf{x}|\boldsymbol{\theta})$

2-step generative process

Latent Variable Models

- Consider a probability distribution parameterized by $\boldsymbol{\theta}$
- Generates samples ($\mathbf{x}$) with probability $p(\mathbf{x}|\boldsymbol{\theta})$

2-step generative process

1. Distribution generates the hidden variable

Latent Variable Models

- Consider a probability distribution parameterized by $\boldsymbol{\theta}$
- Generates samples ($\mathbf{x}$) with probability $p(\mathbf{x}|\boldsymbol{\theta})$

2-step generative process

1. Distribution generates the hidden variable
2. Distribution generates the observation, given the hidden variable

Magazine Example - Sampling an Article

- Assume that the editor has access to $p(\mathbf{x})$
- $\mathbf{x}$ - a random variable that denotes an article

Direct Model

- Sample from $p(\mathbf{x})$ for an article

Magazine Example - Sampling an Article

- Assume that the editor has access to $p(\mathbf{x})$
- $\mathbf{x}$ - a random variable that denotes an article

Direct Model

- Sample from $p(\mathbf{x})$ for an article

Latent Variable Model

1. First sample a topic z from a topic distribution $p(z)$
2. Pick an article from the topic-wise distribution $p(\mathbf{x}|z)$

Latent Variable Models - Introduction

- The observed random variable $\mathbf{x}$ depends on a hidden random variable $\mathbf{z}$
- $\mathbf{z}$ is generated using a *prior* distribution - $p(\mathbf{z})$
- $\mathbf{x}$ is generated using $p(\mathbf{x}|\mathbf{z})$
- Different combinations of $p(\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})$ give different latent variable models
  1. Mixture Models
  2. Factor analysis
  3. Probabilistic Principal Component Analysis (PCA)
  4. Latent Dirichlet Allocation (LDA)

Mixture Models

- A latent discrete state

$$z \in \{1, 2, \ldots, K\}$$

- $p(z) \sim Multinomial(\boldsymbol{\pi})$
- For every state k, we have a probability distribution for $\mathbf{x}$

$$p(\mathbf{x}|z = k) = p_k(\mathbf{x})$$

- Overall, probability for $\mathbf{x}$

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

- A **convex combination** of p_k's
- π_k is the probability of k^{th} mixture component to be true
  - Or, contribution of the k^{th} component
  - Or, the mixing weight

Using Mixture Models

1. Black-box Density Model

- Use $p(\mathbf{x}|\boldsymbol{\theta})$ for many things
- Example: *class conditional density*

2. Clustering

- *Soft clustering*
  1. First learn the parameters of the mixture model
     - Each mixture component corresponds to a cluster k
  2. Compute $p(z = k|\mathbf{x}, \boldsymbol{\theta})$ for every input point $\mathbf{x}$ (*Bayes Rule*)

$$p(z = k|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(z = k|\boldsymbol{\theta})p(\mathbf{x}|z = k, \boldsymbol{\theta})}{\sum_{k'=1}^{K} p(z = k'|\boldsymbol{\theta})p(\mathbf{x}|z = k', \boldsymbol{\theta})}$$

Simple Parameter Estimation

- **Given**: A set of scalar observations

$$x_1, x_2, \ldots, x_n$$

- **Task**: Find the generative model (form and parameters)

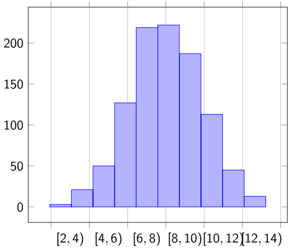
Simple Parameter Estimation

- **Given**: A set of scalar observations

$$x_1, x_2, \ldots, x_n$$

- **Task**: Find the generative model (form and parameters)

1. Observe empirical distribution of x



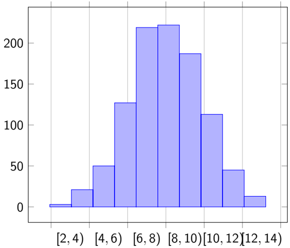
Simple Parameter Estimation

- **Given**: A set of scalar observations

$$x_1, x_2, \ldots, x_n$$

- **Task**: Find the generative model (form and parameters)

1. Observe empirical distribution of x
2. Make choice of the *form* of the probability distribution (Gaussian)



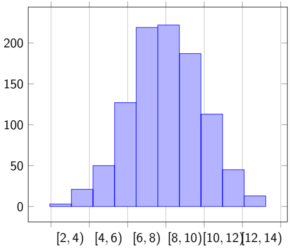
Simple Parameter Estimation

- **Given**: A set of scalar observations

$$x_1, x_2, \ldots, x_n$$

- **Task**: Find the generative model (form and parameters)

1. Observe empirical distribution of x
2. Make choice of the *form* of the probability distribution (Gaussian)
3. Estimate parameters from the data using MLE or MAP (μ and σ)

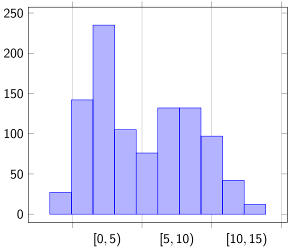


When Data has Multiple Modes

- Single mode is not sufficient
- In reality data is generated from two Gaussians
- How to estimate $\mu_1, \sigma_1, \mu_2, \sigma_2$?

When Data has Multiple Modes

- Single mode is not sufficient
- In reality data is generated from two Gaussians
- How to estimate $\mu_1, \sigma_1, \mu_2, \sigma_2$?
- What if we knew $z_i \in \{1, 2\}$?
  - $z_i = 1$ means that x_i comes from first mixture component
  - $z_i = 2$ means that x_i comes from second mixture component
- **Issue**: z_i's are not known beforehand
- Need to explore 2^N possibilities



Optimizing Likelihood or Posterior is Not Possible

- For direct optimization, we find parameters that maximize (log-)likelihood (or (log-)posterior)
- Easy to optimize if z_i were all known

Optimizing Likelihood or Posterior is Not Possible

- For direct optimization, we find parameters that maximize (log-)likelihood (or (log-)posterior)
- Easy to optimize if z_i were all known
- What happens when z_i's are not known
  - Likelihood and posterior will have multiple modes
  - Non-convex function - harder to optimize

Estimating Parameters of a Mixture Model

- Recall the we want to maximize the log-likelihood of a data set with respect to $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{maximize}}\ \ell(\boldsymbol{\theta})$$

- Log-likelihood for a mixture model can be written as:

$$\begin{aligned} \ell(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \log p(\mathbf{x}_i|\boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} \log \left[\sum_{k=1}^{K} p(z_k) p_k(\mathbf{x}_i|\boldsymbol{\theta}) \right] \end{aligned}$$

- Hard to optimize (a summation inside the log term)

A 2 Step Approach

- Repeat until converged:
  1. Start with some guess for $\boldsymbol{\theta}$ and compute the most likely value for $z_i, \forall i$
  2. Given $z_i, \forall i$, update $\boldsymbol{\theta}$
- Does not explicitly maximize the log-likelihood of mixture model
- Can we come up with a better algorithm?

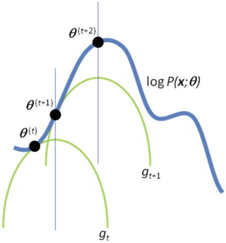
A 2 Step Approach

- Repeat until converged:
  1. Start with some guess for $\boldsymbol{\theta}$ and compute the most likely value for $z_i, \forall i$
  2. Given $z_i, \forall i$, update $\boldsymbol{\theta}$
- Does not explicitly maximize the log-likelihood of mixture model
- Can we come up with a better algorithm?
  - Repeat until converged:
    1. Start with some guess for $\boldsymbol{\theta}$ and compute the probability of $z_i = k, \forall i, k$
    2. Combine probabilities to update $\boldsymbol{\theta}$

Expectation Maximization Algorithm

- A principled approach to maximize a function with latent variables

- At iteration t, for a given value of $\theta^{(t)}$, let Q be a convex function that is a lower bound of $l(\theta)$



Supplementary Figure 1 Convergence of the EM algorithm. Starting from initial parameters $\theta^{(t)}$, the E-step of the EM algorithm constructs a function g_t that lower-bounds the objective function $\log P(x;\theta)$. In the M-step, $\theta^{(t+1)}$ is computed as the maximum of g_t. In the next E-step, a new lower-bound g_{t+1} is constructed; maximization of g_{t+1} in the next M-step gives $\theta^{(t+2)}$, etc.

Steps in EM

- EM is an iterative procedure
- Start with some value for $\boldsymbol{\theta}$
- At every iteration t, update $\boldsymbol{\theta}$ such that the log-likelihood of the data goes up
  - Move from $\boldsymbol{\theta}^{t-1}$ to $\boldsymbol{\theta}$ such that:

$$\ell(\boldsymbol{\theta}) - \ell(\boldsymbol{\theta}^{t-1})$$

    is maximized

EM - Continued

- **Complete log-likelihood** for any LVM

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta})$$

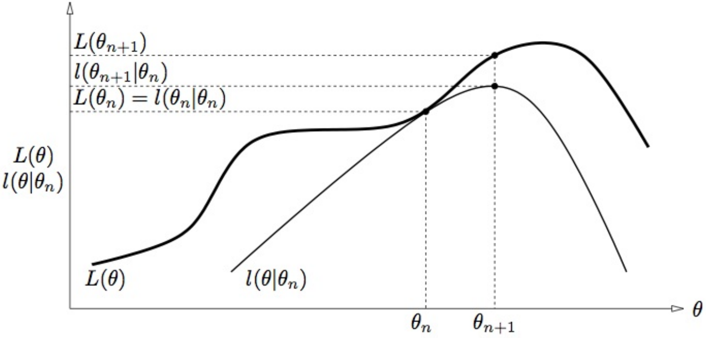
- Cannot be computed as we do not know $\mathbf{z}_i$

Expected complete log-likelihood

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) = \mathbb{E}[\ell(\boldsymbol{\theta}|D, \boldsymbol{\theta}^{t-1})]$$

- Expected value of $\ell(\boldsymbol{\theta}|D, \boldsymbol{\theta}^{t-1})$ for all possibilities of $\mathbf{z}_i$

EM Operation



1. Initialize $\boldsymbol{\theta}$
2. At iteration t, compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$
3. Maximize $Q()$ with respect to $\boldsymbol{\theta}$ to get $\boldsymbol{\theta}^t$
4. Goto step 2

Using EM for MM Parameter Estimation

- EM formulation is generic
- Calculating (E) and maximizing (M) $Q()$ needs to be done for specific instances

Q for MM

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) &= \mathbb{E}\left[\sum_{i=1}^{N} \log p(\mathbf{x}_i, z_i | \boldsymbol{\theta})\right] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log p(\mathbf{x}_i | \boldsymbol{\theta}_k) \\
r_{ik} \triangleq p(z_i &= k | \mathbf{x}_i, \boldsymbol{\theta}^{t-1})
\end{aligned}
$$

E-Step

- Compute $r_{ik}, \forall i, k$

$$\begin{aligned} r_{ik} &= p(z_i = k|\mathbf{x}_i, \boldsymbol{\theta}^{t-1}) \\ &= \frac{\pi_k p(\mathbf{x}_i|\boldsymbol{\theta}_k^{t-1})}{\sum_{k'} \pi'_k p(\mathbf{x}_i|\boldsymbol{\theta}'^{t-1}_k)} \end{aligned}$$

- Compute $Q()$

M-Step

- Maximize $Q()$ w.r.t. $\boldsymbol{\theta}$
- $\boldsymbol{\theta}$ consists of $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_K\}$ and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\}$
- For Gaussian Mixture Model (GMM) ($\boldsymbol{\theta}_k \equiv (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$):

$$\pi_k = \frac{1}{N}\sum_i r_{ik} \quad (1)$$

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_i r_{ik}} \quad (2)$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_i r_{ik}\mathbf{x}_i\mathbf{x}_i^\top}{\sum_i r_{ik}} - \boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top \quad (3)$$

Is K-Means an EM Algorithm?

- Similar to GMM
  1. $\mathbf{\Sigma} = \sigma^2 \mathbf{I}_D$
  2. $\pi_k = \frac{1}{K}$
  3. The most probable cluster for $\mathbf{x}_i$ is computed as the prototype closest to it (hard clustering)

References