COMPSCI 689
Lecture 16: Mixture Models and EM

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Mixture Models

- A mixture model is a simple structured model in the Bayesian network family.
- The model consists of one discrete variable $Z \in \{1, ..., K\}$ called the mixture indicator, and a vector of feature random variables $\mathbf{X} = [X_1, ..., X_D]$ where $X_d \in \mathcal{X}_d$.
- The joint distribution for a mixture model is composed from a product of two factors:

$$P(\mathbf{X} = \mathbf{x}, Z = z|\theta) = P(\mathbf{X} = \mathbf{x}|Z = z, \theta_z)P(Z = z|\pi)$$

Mixture Models: Generative process

- The distribution over Z is a multinoulli distribution $P(Z = z|\pi) = \pi_z$. The parameters π_z are referred to as the *mixture proportions*.

- The distribution over $\mathbf{X}$ given $Z = z$ is referred to as the mixture component distribution: $P(\mathbf{X}|Z = z, \theta_z)$.
- Thus, each setting of the mixture indicator variable z induces its own distribution over $\mathbf{X}$ via the parameters θ_z.

Mixture Models: Observed Z

- If we observe the value of Z in the training data, then this model is what is known as a generative classifier. Z is the class label.

- This model makes predictions at test time using Bayes rule:

$$P(Z=z|\mathbf{X}=\mathbf{x},\theta)=\frac{P(\mathbf{X}=\mathbf{x}|Z=z,\theta_z)P(Z=z|\pi)}{\sum_{z'=1}^{K}P(\mathbf{X}=\mathbf{x}|Z=z',\theta_z)P(Z=z'|\pi)}$$

Mixture Models: Observed Z

- If all the X_d's are single Bernoulli random variables, this model is called *Bernoulli Naive Bayes*.
- If all the X_d's are single Multinomial random variables, this model is called *Multinomial Naive Bayes*.
- If all the X_d's are single Normal random variables, this model is called *Gaussian Naive Bayes*.

Mixture Models: Observed Z

- If $\mathbf{X}$ is jointly Gaussian, this model is called *Quadratic Discriminant Analysis*.
- If $\mathbf{X}$ is jointly Gaussian and all mixture components have the same covariance matrix, this model is called *Linear Discriminant Analysis*.
- Naive Bayes, QDA and LDA have some interesting properties as generative classifiers, but they are nearly always out-performed by discriminative classifiers (logistic regression, SVMs, neural networks).

Mixture Models: Unobserved Z

- In the case where Z is not observed in the data, a mixture model can be interpreted as a clustering model.

- The model asserts that observed data vectors $\mathbf{x}$ belong to groups or clusters as specified by their unobserved (or latent) mixture indicator variable values.

- Learning algorithms for the case of unobserved Z attempt to explain a complex data distribution in terms of a finite collection of simple mixture components.

- **Question:** What optimization criteria should we use to learn the parameters of a mixture model with unobserved Z's?

MLE for Mixtures

- When we don't know what the right value of the mixture indicator variable is, we can still apply the maximum likelihood framework.

- Specifically, we need to select the parameters to maximize the likelihood of the *observed* data. For a mixture model with unobserved $Z's$, this is just the $\mathbf{x}'s$.

- **Question:** What is the likelihood of just the $\mathbf{x}'s$?

The Marginal Likelihood

- The function we need to optimize is the marginal log likelihood.
- The marginal likelihood of **x** is given by:

$$P(\mathbf{X} = \mathbf{x}|\theta) = \sum_{z=1}^{K} P(\mathbf{X} = \mathbf{x}|Z = z, \theta_z)P(Z = z|\pi)$$

- The marginal log likelihood of a data set $\mathcal{D} = \{\mathbf{x}_n\}_{n=1:N}$ is:

$$\mathcal{L}(\mathcal{D}, \theta) = \sum_{n=1}^{N} \log\left(\sum_{z=1}^{K} P(\mathbf{X}_n = \mathbf{x}_n|Z = z, \theta_z)P(Z = z|\pi)\right)$$

Learning

- We can learn the model parameters $\theta = [\pi, \theta_1, ..., \theta_K]$ directly by numerically optimizing the marginal likelihood $\mathcal{L}(\mathcal{D}, \theta)$.
- However, there is an interesting alternative approach based on lower bound optimization that results in a coordinate ascent algorithm that has exact updates and is guaranteed to monotonically improve $\mathcal{L}(\mathcal{D}, \theta)$ without line search.
- This algorithm was developed in the statistics literature and is known as the *Expectation Maximization* or EM algorithm.

Problems with the Marginal Likelihood

- Recall the marginal log likelihood of a data set $\mathcal{D}$ is:

$$\mathcal{L}(\mathcal{D},\theta) = \sum_{n=1}^{N} \log\left(\sum_{z=1}^{K} P(\mathbf{X}_n = \mathbf{x}_n | Z = z, \theta_z) P(Z = z|\pi)\right)$$

- The main barrier to analytically maximizing $\mathcal{L}(\mathcal{D},\theta)$ is the log of the sum over z.

- Learning would decompose nicely over the mixture model's Bayesian network if we could only take the sum over z outside the log...

Key Idea 1: Learning by Bound Optimization

- Suppose we have a function $f(\mathbf{x})$ that we would like to maximize, but directly maximizing $f(\mathbf{x})$ is hard for some reason.

- Suppose we have a function $g(\mathbf{x})$ that provides a lower bound $g(\mathbf{x}) \leq f(\mathbf{x})$ on $f(\mathbf{x})$ for all $\mathbf{x}$.

- In that case, we can maximize $g(\mathbf{x})$ instead of $f(\mathbf{x})$.

- If we find a local maximum $\mathbf{x}_*$ of $g(\mathbf{x})$, we know that the maximum of $f(\mathbf{x})$ is at least $f(\mathbf{x}_*)$.

- Importantly, we can also add more optimization variables to g if this helps in some way. For example, if $g(\mathbf{x},\mathbf{y}) \leq f(\mathbf{x})$ for all $\mathbf{x}$ and $\mathbf{y}$, we can optimize $g(\mathbf{x},\mathbf{y})$ over $\mathbf{x}$ and $\mathbf{y}$.

Key Idea 2: Jensen's Inequality

- Suppose $f()$ is a concave function.

- Consider the application of $f()$ to a convex combination of inputs $x_1, \ldots, x_K$ where the combination weights are $\alpha = [\alpha_1, \ldots, \alpha_K]$, $\alpha_k > 0$ and $\sum_{k=1}^K \alpha_k = 1$:

$$f\Big(\sum_{k=1}^{K} \alpha_k x_k\Big)$$

- Jensen's Inequality states that for any valid choice of α_k's, the following lower bound holds:

$$\sum_{k=1}^{K} \alpha_k f(x_k) \leq f\Big(\sum_{k=1}^{K} \alpha_k x_k\Big)$$

Lower-Bounding Marginal Likelihood

- Suppose we introduce an auxiliary set of multinoulli probability distributions $q_n(Z = z) = \phi_{zn}$ into the marginal likelihood in the following way:

$$\mathcal{L}(\mathcal{D}, \theta) = \sum_{n=1}^{N} \log \left(\sum_{z=1}^{K} \frac{\phi_{zn}}{\phi_{zn}} P(\mathbf{X}_n = \mathbf{x}_n | Z = z, \theta_z) P(Z = z | \pi) \right)$$

- Note that this doesn't change the marginal likelihood, but since log is concave, we can now apply Jensen's inequality using $q_n(Z = z)$ as the convex combination weights for each n:

$$\mathcal{L}(\mathcal{D}, \theta) \geq \sum_{n=1}^{N} \sum_{z=1}^{K} \phi_{zn} \log \left(\frac{P(\mathbf{X}_n = \mathbf{x}_n | Z = z, \theta_z) P(Z = z | \pi)}{\phi_{zn}} \right)$$

The Q Function

- This lower bound is denoted by $Q(\mathcal{D}, \theta, \phi) \leq \mathcal{L}(\mathcal{D}, \theta)$
- We can now simplify $Q(\mathcal{D}, \theta, \phi)$ as shown below:

$$
\begin{aligned}
Q(\mathcal{D}, \theta, \phi) &= \sum_{n=1}^{N} \sum_{z=1}^{Z} \phi_{zn} \log \left(\frac{P(\mathbf{X} = \mathbf{x} | Z = z, \theta_z) P(Z = z | \pi)}{\phi_{zn}} \right) \\
&= \sum_{n=1}^{N} \sum_{z=1}^{Z} \phi_{zn} \big(\log P(\mathbf{X} = \mathbf{x} | Z = z, \theta_z) + \log P(Z = z | \pi) - \log \phi_{zn} \big) \\
&= \sum_{n=1}^{N} \Big(\mathbb{E}_{q_n} [\log P(\mathbf{X} = \mathbf{x}, Z = z | \theta)] + \mathbb{H}(q_n) \Big)
\end{aligned}
$$

- The first term is the *expected complete log likelihood* and the second term is the entropy of q_n.

Optimizing the Q Function

- We could again simply optimize $Q(\mathcal{D}, \theta, \phi)$ numerically, but it turns out that we can obtain a coordinate ascent algorithm with exact steps.

- This algorithm starts from random parameters and optimizes the ϕ_n variables for every data case with the model parameters θ fixed.

- It then optimizes the θ variables with the ϕ variables fixed.

- We obtain these two steps by solving the gradient equations $\nabla_\theta Q(\mathcal{D}, \theta, \phi) = 0$ and $\nabla_\phi Q(\mathcal{D}, \theta, \phi) = 0$. This yields the EM algorithm.

The Expectation Maximization Algorithm

E-Step: In E-Step step of the algorithm, we determine the responsibility of each mixture component for each data case:

$$\phi_{zn}^{t} \leftarrow P(Z = z|\mathbf{X} = \mathbf{x}_n, \theta^{t-1})$$

M-Step: In the M-step, we update the parameters using responsibility weighted averages.

$$\pi_z^t \leftarrow \frac{1}{N}\sum_{n=1}^{N}\phi_{zn}^{t}$$

$$\theta_{dz}^{t} \leftarrow \arg\max_{\theta_{dz}} \sum_{n=1}^{N}\phi_{zn}^{t}\log P(\mathbf{X}_d = \mathbf{x}_{dn}|Z = z, \theta_{dz}^{t-1})$$

Example: Gaussian Mixture Models

- In a Gaussian mixture model, there is one block of observed data variables that are Gaussian distributed given the value of the mixture indicator.
- $P(Z = z|\pi) = \pi_z$
- $P(\mathbf{X} = \mathbf{x}|Z = z, \theta_z) = \mathcal{N}(\mathbf{x}, \mu_z, \Sigma_z)$
- $\theta_z = [\mu_z, \Sigma_z]$

Example: EM for Gaussian Mixture Models

E-Step: Compute responsibilities.

$$\phi_{zn}^{t} \leftarrow \frac{\pi_z^{t-1}\mathcal{N}(\mathbf{x}_n;\mu_z^{t-1},\Sigma_z^{t-1})}{\sum_{z'=1}^{K}\pi_{z'}^{t-1}\mathcal{N}(\mathbf{x}_n;\mu_{z'}^{t-1},\Sigma_{z'}^{t-1})}$$

M-Step: Update parameters.

$$\pi_z^t \leftarrow \frac{1}{N}\sum_{n=1}^{N}\phi_{zn}^t \qquad \mu_z^t = \frac{\sum_{n=1}^{N}\phi_{zn}^t\mathbf{x}_n}{\sum_{n'=1}^{N}\phi_{zn'}^t}$$

$$\Sigma_z^t = \frac{\sum_{n=1}^{N}\phi_{zn}^t(\mathbf{x}_n-\mu_z^t)^T(\mathbf{x}_n-\mu_z^t)}{\sum_{n=1}^{N}\phi_{zn}^t}$$