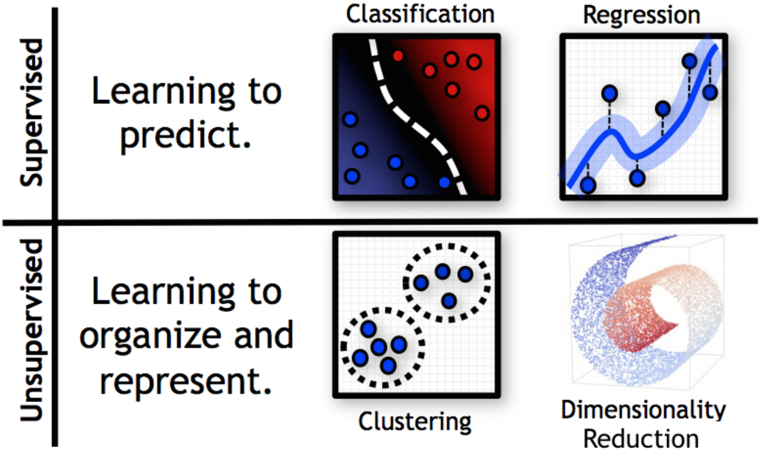
COMPSCI 689
Lecture 17: Mixture Models

Brendan O'Connor

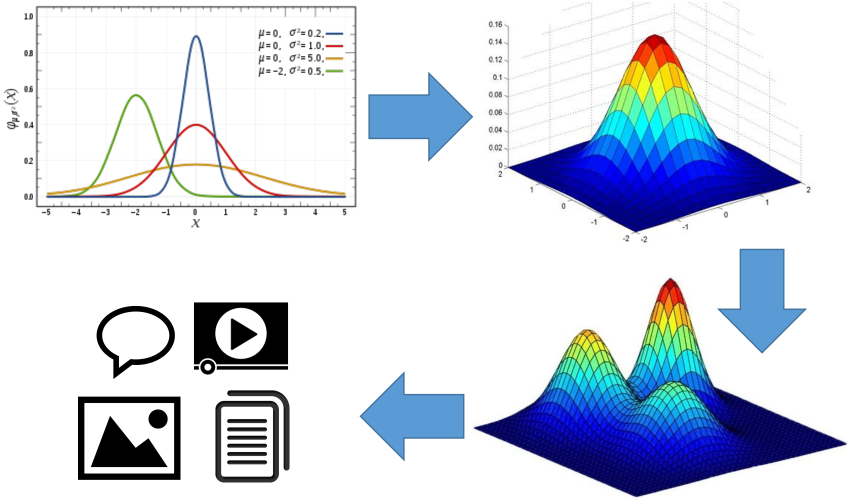
College of Information and Computer Sciences
University of Massachusetts Amherst

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

Machine Learning Tasks



Probabilistic Unsupervised Learning



Probabilistic Unsupervised Learning

- In probabilistic unsupervised learning, our goal is to model multivariate data $\mathbf{x} = [x_1, ..., x_D]$ generated by an unknown probabilistic process using a probabilistic model learned from a data set $\mathcal{D} = \{\mathbf{x}_n | 1 \leq n \leq N\}$.

- Since the data are vectors, we use vector-valued random variables to model them $\mathbf{X} = [X_1, ..., X_D]$.

- Each data dimension d takes values from a potentially different set $\mathcal{X}_d$. We have $\mathbf{x} \in \mathcal{X}$. $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_D$.

Joint Distributions

- A probability distribution over the joint settings of multiple random variables is referred to as a *joint distribution*.
- When all dimensions of $\mathbf{x}$ are discrete, the joint distribution is represented by a *joint probability mass* function $P(\mathbf{X} = \mathbf{x}) = P(X_1 = x_1, ..., .X_D = x_d)$.
- When all dimensions of $\mathbf{x}$ are continuous, the joint distribution is represented by a *joint probability density* function $p(\mathbf{X} = \mathbf{x}) = p(X_1 = x_1, ..., .X_D = x_d)$.
- When the data are of mixed-type, we can still model them via a probability distribution consisting of both mass and density function components.

Probabilistic Inference

- Joint probability distributions allow us to compute the joint probability (mass or density) of a fully specified vector of values $\mathbf{x} = [x_1, ..., x_D]$ of a vector valued random variable $\mathbf{X}$.

- Often, we are instead interested in computing the probability of an assignment to a subset of the dimensions of a vector-valued random variable, an operation referred to as *marginalization*.

- We can also use joint distributions to make probabilistic predictions about the distribution of one subset of dimensions in the given values for another subset. This operation is called *conditioning*.

- Marginalization and conditioning are the two fundamental probabilistic inference operations.

Probabilistic Inference

Product of Marginals

- The most basic way to construct a joint probability model over a vector-valued random variable $\mathbf{X}$ is to model the marginal distribution of each random variable and define the joint distribution as the product of marginals.

$$P(\mathbf{X} = \mathbf{x}|\theta) = \prod_{d=1}^{D} P(X_d = x_d|\theta_d)$$

$$p(\mathbf{X} = \mathbf{x}|\theta) = \prod_{d=1}^{D} p(X_d = x_d|\theta_d)$$

- However, this is highly restrictive as we can't model relationships between different data dimensions. The model asserts that add dimensions are probabilistically independent of each other.

Mixture Models

- Mixture models are basic universal probability distribution models that only require a small change to the product of marginals.
- They are constructed by introducing a finite discrete *latent* random variable Z to the observed variables $\mathbf{X}$.
- Instead of using a product of marginals to model $\mathbf{X}$, we use a product of distributions conditioned on Z.
- The joint distribution of $\mathbf{X}$ and Z is given by:

$$P(\mathbf{X} = \mathbf{x}, Z = z|\theta) = P(Z = z|\pi) \prod_{d=1}^{D} P(X_d = x_d|Z = z, \phi_{dz})$$

Mixture Models

- The distribution of $\mathbf{X}$ is given by marginalization of the joint over the values of $z \in [1, ..., K]$:

$$P(\mathbf{X} = \mathbf{x}|\theta) = \sum_{z=1}^{K} P(Z = z|\pi) \prod_{d=1}^{D} P(X_d = x_d|Z = z, \phi_{dz})$$

- The construction above is shown for an $\mathbf{X}$ that is all discrete, but the same construction works for continuous $\mathbf{X}$ as well as for mixed-types.

Example: Binary Mixture

Suppose the data are binary. We have $x_d \in \{0, 1\}$ for $1 \leq d \leq D$. We construct a mixture distribution for these data as follows:

$$
\begin{aligned}
P(Z = z|\pi) &= \pi_z \\
P(X_d = x_d|Z = z, \phi_{dz}) &= \phi_{dz}^{x_d}(1 - \phi_{dz})^{1-x_d} \\
P(\mathbf{X} = \mathbf{x}|Z = z, \phi_z) &= \prod_{d=1}^{D} P(X_d = x_d|Z = z, \phi_{dz}) \\
P(\mathbf{X} = \mathbf{x}, Z = z|\pi, \phi_z) &= P(Z = z|\pi)P(\mathbf{X} = \mathbf{x}|Z = z, \phi_z) \\
&= \pi_z \cdot \prod_{d=1}^{D} \phi_{dz}^{x_d}(1 - \phi_{dz})^{1-x_d}
\end{aligned}
$$

Example: Binary Mixture

The mixture distribution over **X** is obtained by marginalizing the mixture indicator variable Z out of the model:

$$P(\mathbf{X} = \mathbf{x}|\pi, \phi) = \sum_{z=1}^{K} P(Z = z|\pi)P(\mathbf{X} = \mathbf{x}|Z = z, \phi_z)$$
$$= \sum_{z=1}^{K} \pi_z \cdot \prod_{d=1}^{D} \phi_{dz}^{x_d}(1 - \phi_{dz})^{1-x_d}$$

Example: Binary Mixture

Given a vector **x**, we can use probabilistic inference to infer the probability distribution over Z:

$$P(Z = z|\mathbf{X} = \mathbf{x}, \pi, \phi) = \frac{P(Z = z, \mathbf{X} = \mathbf{x}|\pi, \phi)}{P(\mathbf{X} = \mathbf{x}|\pi, \phi)}$$
$$= \frac{\pi_z \cdot \prod_{d=1}^{D} \phi_{dz}^{x_d}(1 - \phi_{dz})^{1-x_d}}{\sum_{z'=1}^{K} \pi_{z'} \cdot \prod_{d'=1}^{D} \phi_{d'z'}^{x_{d'}}(1 - \phi_{d'z'})^{1-x_{d'}}}$$

Note that when we have many dimensions, we need to be careful with this computation as both the numerator and denominator have the potential to underflow.

Example: Normal Mixture

Suppose the data x_d are real-valued for $1 \leq d \leq D$. We can model the data using a mixture where the component densities are conditional univariate normal distributions:

$$
\begin{aligned}
P(Z = z|\pi) &= \pi_z \\
p(X_d = x_d|Z = z, \mu_{dz}, \sigma_{dz}) &= \mathcal{N}(x_d; \mu_{dz}, \sigma^2_{dz}) \\
p(\mathbf{x} = \mathbf{x}|Z = z, \mu_z, \sigma_z) &= \prod_{d=1}^{D} \mathcal{N}(x_d; \mu_{dz}, \sigma^2_{dz}) \\
P(\mathbf{X} = \mathbf{x}, Z = z|\pi, \mu, \sigma) &= P(Z = z|\pi)p(\mathbf{x} = \mathbf{x}|Z = z, \mu_z, \sigma_z) \\
&= \pi_z \cdot \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma^2_{dz}}} \exp\left(-\frac{1}{2\sigma^2_{dz}}(x_d - \mu_{dz})^2\right)
\end{aligned}
$$

Example: Normal Mixture

The mixture distribution over **X** is obtained by marginalizing the mixture indicator variable Z out of the model:

$$p(\mathbf{X} = \mathbf{x} | \pi, \mu, \sigma) = \sum_{z=1}^{K} P(Z = z | \pi) p(\mathbf{x} = \mathbf{x} | Z = z, \mu_z, \sigma_z)$$

$$= \sum_{z=1}^{K} \pi_z \cdot \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{dz}^2}} \exp\left(-\frac{1}{2\sigma_{dz}^2} (x_d - \mu_{dz})^2 \right)$$

Example: Normal Mixture

Given a vector **x**, we can use probabilistic inference to infer the probability distribution over Z:

$$P(Z = z|\mathbf{X} = \mathbf{x}, \pi, \mu, \sigma) = \frac{P(Z = z, \mathbf{X} = \mathbf{x}|\pi, \mu, \sigma)}{p(\mathbf{X} = \mathbf{x}|\pi, \mu, \sigma)}$$

$$= \frac{\pi_z \cdot \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{dz}^2}} \exp\left(-\frac{1}{2\sigma_{dz}^2}(x_d - \mu_{dz})^2\right)}{\sum_{z'=1}^{K} \pi_{z'} \cdot \prod_{d'=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{d'z'}^2}} \exp\left(-\frac{1}{2\sigma_{d'z'}^2}(x_d - \mu_{d'z'})^2\right)}$$

Note that when we have many dimensions, we need to be careful with this computation as both the numerator and denominator have the potential to underflow or overflow.

Losses for Distributions

Unlike in supervised learning, there are few commonly used losses between distributions:

- Absolute Loss: $L_1(P_*\|P_\theta) = \mathbb{E}_{P_*(\mathbf{X})}\left[|P_*(\mathbf{x}) - P(\mathbf{x}|\theta)|\right]$
- Squared Loss: $L_2(P_*\|P_\theta) = \mathbb{E}_{P_*(\mathbf{X})}\left[(P_*(\mathbf{x}) - P(\mathbf{x}|\theta))^2\right]$
- KL Divergence: $KL(P_*\|P_\theta) = \mathbb{E}_{P_*(\mathbf{X})}\left[\log\left(\frac{P_*(\mathbf{x})}{P(\mathbf{x}|\theta)}\right)\right]$

Question: Which of these losses can we approximate using a sample of data $\mathcal{D} = \{\mathbf{x}_n\}_{1:N}$?

Optimizing KL Divergence

$$\begin{aligned}
\min_{\theta} KL(P_* \| P_\theta) &= \min_{\theta} \sum_{\mathbf{x} \in \mathcal{X}} P_*(\mathbf{x}) \big(\log P_*(\mathbf{x}) - \log P(\mathbf{x}|\theta) \big) \\
&= \min_{\theta} \sum_{\mathbf{x} \in \mathcal{X}} P_*(\mathbf{x}) \log P_*(\mathbf{x}) - \sum_{\mathbf{x} \in \mathcal{X}} P_*(\mathbf{x}) \log P(\mathbf{x}|\theta) \\
&= \min_{\theta} - \sum_{\mathbf{x} \in \mathcal{X}} P_*(\mathbf{x}) \log P(\mathbf{x}|\theta) \\
&\approx \min_{\theta} - \frac{1}{N} \sum_{n=1}^{N} \log P(\mathbf{x}_n|\theta)
\end{aligned}$$

Optimization-Based Unsupervised Learning

- As we can see, selecting the value of θ that minimizes the NLL both makes the data the most likely and is a Monte Carlo approximation to selecting the value of θ that minimizes $KL(P_*\|P_\theta)$.
- The dominant approaches to optimization-based unsupervised learning of probabilistic models are thus maximum likelihood estimation and its penalized/regularized variants.

Learning for Mixture Models

- In a mixture model, the full joint distribution includes the data variables $\mathbf{X}$ and the latent mixture indicator variable Z.
- Since the mixture indicator variables Z are not observed, we need to marginalize them out of the model and then minimize the negative log likelihood of the marginalized distribution.
- We refer to the resulting optimization criterion as the *negative log marginal likelihood* (NLML) function.

Learning for Mixture Models

The negative log marginal likelihood for a generic mixture model is given below:

$$
\begin{aligned}
nlml(\mathcal{D}, \theta) &= -\sum_{n=1}^{N} \log P(\mathbf{x}_n|\theta) \\
&= -\sum_{n=1}^{N} \log \left(\sum_{z=1}^{K} P(Z = z|\pi) \prod_{d=1}^{D} P(X_d = x_{nd}|Z = z) \right)
\end{aligned}
$$

We can learn mixture models using direct negative log marginal likelihood minimization. Note that parameter transformations must be used to deal with constrained parameter spaces.
(Alternative: EM algorithm)