CSC 2516: Introduction to Machine Learning

Week 11 Tutorial - EM Algorithm

Fall 2024

University of Toronto, Fall 2024

Bernoulli Distribution

- Bernoulli distribution: $\mathbf{X}$ is a random variable with two outcomes. We say that $\mathbf{X}$ follows $\mathcal{B}(x;\mu)$ if:

$$P(\mathbf{X}=x)=\mu^x(1-\mu)^{1-x}, x\in\{0,1\}=\begin{cases}\mu & x=1\\ 1-\mu & x=0\end{cases}$$

- Mean: $\mathbb{E}[\mathcal{B}(x;\mu)]=1\times\Pr(x=1)+0\times\Pr(x=0)=\mu$
- Variance:

$$\begin{aligned}\mathrm{Var}[\mathcal{B}(x;\mu)]&=\mathbb{E}[\mathcal{B}(x^2;\mu)]-\mathbb{E}[\mathcal{B}(x;\mu)^2]\\&=1^2\times\Pr(x=1)+0^2\times\Pr(x=0)-\mu^2\\&=\mu-\mu^2\\&=\mu(1-\mu)\end{aligned}$$

- Example: A coin follows Bernoulli distribution.

Mixture of Bernoulli

- Mixture of Gaussians is defined over continuous variables. Mixture of Bernoulli can be seen as its counterpart for binary variables.

Mixture of Bernoulli

- Mixture of Gaussians is defined over continuous variables. Mixture of Bernoulli can be seen as its counterpart for binary variables.
- Like a GMM, a plausible data generation "story" for x is:

Mixture of Bernoulli

- Mixture of Gaussians is defined over continuous variables. Mixture of Bernoulli can be seen as its counterpart for binary variables.
- Like a GMM, a plausible data generation "story" for x is:
  - Given K clusters, $\{1, \ldots, K\}$

Mixture of Bernoulli

- Mixture of Gaussians is defined over continuous variables. Mixture of Bernoulli can be seen as its counterpart for binary variables.
- Like a GMM, a plausible data generation "story" for x is:
  - Given K clusters, $\{1, \ldots, K\}$
  - Sample a cluster $z \sim \Pr(z)$
    - We parameterize $\Pr(z)$ with π, i.e., $\Pr(z = k) = \pi_k$

Mixture of Bernoulli

- Mixture of Gaussians is defined over continuous variables. Mixture of Bernoulli can be seen as its counterpart for binary variables.
- Like a GMM, a plausible data generation "story" for x is:
  - Given K clusters, $\{1, \ldots, K\}$
  - Sample a cluster $z \sim \Pr(z)$
    - We parameterize $\Pr(z)$ with π, i.e., $\Pr(z = k) = \pi_k$
  - Given z, sample $x \sim \mathcal{B}(x; \mu_k)$.

Mixture of Bernoulli

- We say x follows mixtures of Bernoulli distributions.
- Recall:
  - $\Pr(z = k) = \pi_k$
  - $\Pr(x|z = k) = \mu_k^x(1 - \mu_k)^{1-x}$
  - $\Pr(x, z = k) = \Pr(x|z = k)\Pr(z = k) = \mu_k^x(1 - \mu_k)^{1-x} \times \pi_k$
  - $\Pr(x) = \sum_{k=1}^{K} \Pr(x, z = k)$, marginalization
- Thus, the pmf of $\mathbf{X}$ can be expressed as:

$$\Pr(x) = \sum_{k=1}^{K} \mu_k^x(1 - \mu_k)^{1-x} \times \pi_k$$

- Note that we can tell that $\mu_k^x(1 - \mu_k)^{1-x}$ is conditioned on $z_n = k$ as we are using μ_k

Maximum Likelihood

- We want to learn the parameters $\{\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k\}$ from the observations $\{x_i\}_{i=1}^N$.

Maximum Likelihood

- We want to learn the parameters $\{\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k\}$ from the observations $\{x_i\}_{i=1}^N$.
- We want to select parameters that maximize the log likelihood of the observed data,
  $\log \Pr(x_1, \ldots, x_n) = \log \prod_{n=1}^N \Pr(x_n) = \sum_{n=1}^N \log \Pr(x_n)$.
- Training Objective:

$$\max_{\pi,\mu} \sum_{n=1}^{N} \log \Pr(X_n)$$

$$\max_{\pi,\mu} \sum_{n=1}^{N} \log \sum_{k=1}^{K} \mu_k^{x_n} (1-\mu_k)^{1-x_n} \times \pi_k$$

- Sum inside log $\rightarrow$ EM algorithm

Fully Supervised Case

- If our data included z_n values, MLE would be much easier.

$$\log \Pr(X, Z; \mu, \pi) = \sum_{n=1}^{N} \log \Pr(x_n, z_n; \mu, \pi)$$
$$= \sum_{n=1}^{N} \log \Pr(x_n | z_n; \mu, \pi) \Pr(z_n; \mu, \pi)$$

Above, we know what the value of z_n is, so we can just write it down in terms of z_n, but we could also express it as a sum. The indicator function ensures only a single term contributes, so the value is the same. This will allow easier manipulations later.

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}[z_n = k] \log \Pr(x_n | z_n; \mu, \pi) \Pr(z_n; \mu, \pi)$$

E-step: Why "Expectation"

- We don't know that actual values of z_n, i.e., when $\mathbb{I}[z_n = k] = 1$, they are latent

E-step: Why "Expectation"

- We don't know that actual values of z_n, i.e., when $\mathbb{I}[z_n = k] = 1$, they are latent
- We do know how to ask "how likely is the value $z_n = k$ according to our model?"
  - Using the posterior probability, $\Pr(z_n = k | x_n; \mu, \pi)$.

E-step: Why "Expectation"

- We don't know that actual values of z_n, i.e., when $\mathbb{I}[z_n = k] = 1$, they are latent
- We do know how to ask "how likely is the value $z_n = k$ according to our model?"
  - Using the posterior probability, $\Pr(z_n = k | x_n; \mu, \pi)$.
- Why can't we just use the most likely z_n value and call it a day?

E-step: Why "Expectation"

- We don't know that actual values of z_n, i.e., when $\mathbb{I}[z_n = k] = 1$, they are latent
- We do know how to ask "how likely is the value $z_n = k$ according to our model?"
  - Using the posterior probability, $\Pr(z_n = k | x_n; \mu, \pi)$.
- Why can't we just use the most likely z_n value and call it a day?
- Instead we want to consider all the possible assignments to z_n

E-step: Why "Expectation"

- We don't know that actual values of z_n, i.e., when $\mathbb{I}[z_n = k] = 1$, they are latent
- We do know how to ask "how likely is the value $z_n = k$ according to our model?"
  - Using the posterior probability, $\Pr(z_n = k | x_n; \mu, \pi)$.
- Why can't we just use the most likely z_n value and call it a day?
- Instead we want to consider all the possible assignments to z_n
- Asked another way, what are the expected values of z_n?
  - $\mathbb{E}[\mathbb{I}[z_n = k | x; \mu, \pi]] = \sum_{k=1}^{K} \mathbb{I}[z_n = k | x_n; \mu, \pi] \Pr(z_n = k | x_n; \mu, \pi) = \Pr(z_n = k | x_n; \mu, \pi)$

E-step: Compute the Posterior

- Compute the posterior probability $z_{nk} = \Pr(z_n = k|x_n)$ using Bayes' theorem:

E-step: Compute the Posterior

- Compute the posterior probability $z_{nk} = \Pr(z_n = k|x_n)$ using Bayes' theorem:

$$\Pr(z_n = k|x_n) = \frac{\Pr(z_n = k, x_n)}{\Pr(x_n)} = \frac{\Pr(x_n|z_n = k)\Pr(z_n = k)}{\sum_{k'=1}^{K} \Pr(x_n|z_n = k')\Pr(z_n = k')}$$
$$= \frac{\mu_k^{x_n}(1-\mu_k)^{1-x_n} \times \pi_k}{\sum_{k'=1}^{K} \mu_i^{x_n}(1-\mu_i)^{1-x_n} \times \pi_{k'}}$$

- z_{nk} can be interpreted as how much we think a cluster k is responsible for generating a datapoint x_n.

M-step - optimize the joint log likelihood

- The joint likelihood can be expressed as:

$$\log p(\mathbf{X}, \mathbf{Z}; \mu, \pi)$$
$$= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} (\log \pi_k + x_n \log \mu_k + (1 - x_n) \log(1 - \mu_k))$$

$$\log p(\mathbf{X}, \mathbf{Z}; \mu, \pi)$$
$$= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} (\log \pi_k + x_n \log \mu_k + (1 - x_n) \log(1 - \mu_k))$$

M-step: Optimize the Joint Log Likelihood

- Lets look back at our joint likelihood:

$$\begin{aligned}
&\log \Pr(X, Z; \mu, \pi) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}[z_n = k] \Pr(x_n | z_n; \mu, \pi) \Pr(z_n; \mu, \pi)
\end{aligned}$$

Now lets replace the known $\mathbb{I}[z_n = k]$ with our expectation given the current parameters.

$$\begin{aligned}
&= \sum_{n=1}^{N} \mathbb{E}\left[\sum_{k=1}^{K} \mathbb{I}[z_n = k] \Pr(x_n | z_n; \mu, \pi) \Pr(z_n; \mu, \pi) \right] \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \Pr(x_n | z_n; \mu, \pi) \Pr(z_n; \mu, \pi) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} (x_n \log \mu_k + (1 - x_n) \log(1 - \mu_k) + \log \pi_k)
\end{aligned}$$

M-step: Optimizing the Joint Log Likelihood

- Assume the z_{nk} is known, setting the derivative respect to μ_k to zero, we get:

$$\mu_k = \frac{\sum_{n=1}^{N} z_{nk} x_n}{\sum_{n=1}^{N} z_{nk}}$$

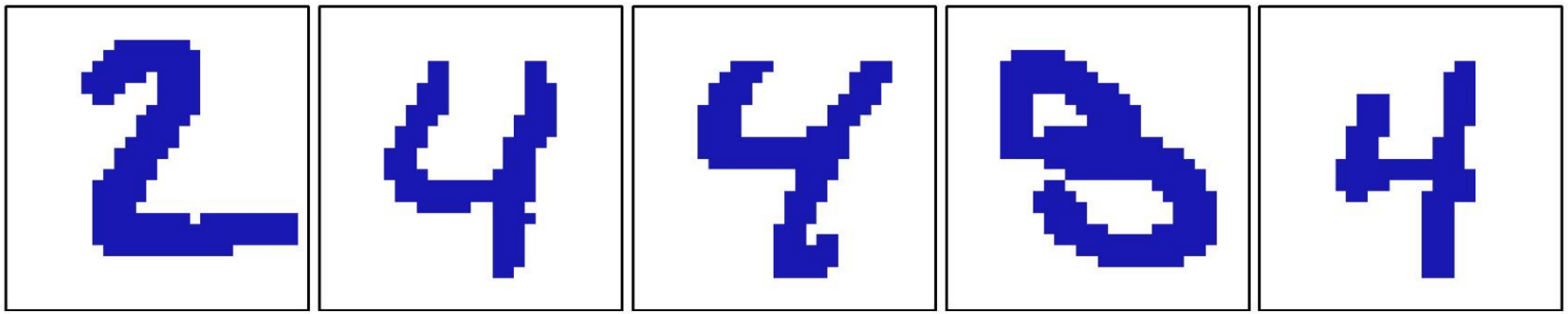
- Similarly for π_k:

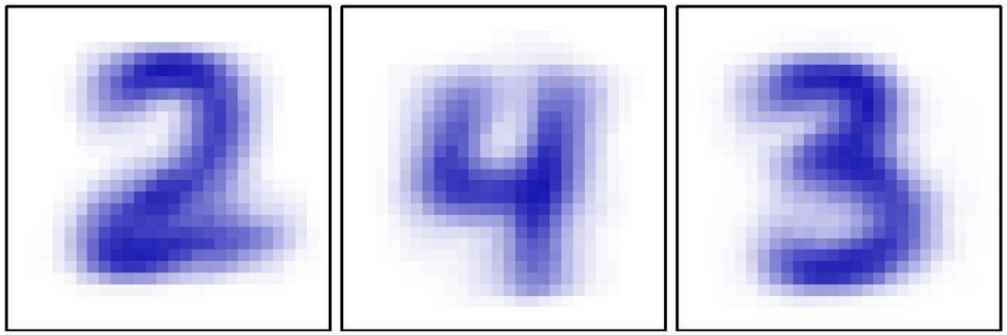
$$\pi_k = \frac{\sum_{n=1}^{N} z_{nk}}{N}$$

- Interpretation: The mean of component k is equal to the weighted mean of the data, with weighted coefficients proportional to the responsibility.

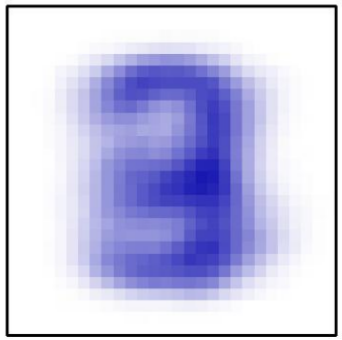
Example

Training data





Learned μ_k for the first three components.



A single multinomial Bernoulli distribution fit to the full data.

True or False

The EM algorithm optimizes a lower bound on its objective function, which is the marginal likelihood $\prod_i P(x_i)$ of the observed data points $x_1, x_2, ...x_N$.

True or False

The EM algorithm optimizes a lower bound on its objective function, which is the marginal likelihood $\prod_i P(x_i)$ of the observed data points $x_1, x_2, ...x_N$.

True.

True or False

The EM algorithm is guaranteed to never decrease the value of its objective function on any iteration.

True or False

The EM algorithm is guaranteed to never decrease the value of its objective function on any iteration.

True

True or False

The objective function optimized by the EM algorithm can also be optimized by a gradient descent algorithm which will find the global optimal solution, whereas EM finds its solution more quickly but may return only a locally optimal solution.

True or False

The objective function optimized by the EM algorithm can also be optimized by a gradient descent algorithm which will find the global optimal solution, whereas EM finds its solution more quickly but may return only a locally optimal solution.

False

True or False

Consider the set of training data below, and two clustering algorithms: K-Means, and a Gaussian Mixture Model (GMM) trained using EM. These two clustering algorithms will produce the same cluster centers (means) for this data set.

True or False

Consider the set of training data below, and two clustering algorithms: K-Means, and a Gaussian Mixture Model (GMM) trained using EM. These two clustering algorithms will produce the same cluster centers (means) for this data set.

False. In k-means, the means of the clusters are determined by an average of the points assigned to that cluster, but in GMM the means of each cluster are (differently) weighted averages of all points.

HMMs

- Hidden Markov Models
- Classic NLP model for things like Part of Speech Tagging
- Used in things like Computational Biology for Genome Tagging
- EM is used for unsupervised training, called Baum-Welch