Lecture 13: Expectation Maximization

Mark Hasegawa-Johnson

ECE 417: Multimedia Signal Processing, Fall 2021

Outline

Bayesian Classifiers

A Bayesian classifier chooses a label, $y \in \{0 \ldots N_Y - 1\}$, that has the minimum probability of error given an observation, $\vec{x} \in \Re^D$:

$$\begin{aligned}
\hat{y} &= \operatorname*{argmin}_y \Pr\left\{ Y \neq y | \vec{X} = \vec{x} \right\} \\
&= \operatorname*{argmax}_y \Pr\left\{ Y = y | \vec{X} = \vec{x} \right\} \\
&= \operatorname*{argmax}_y p_{Y|\vec{X}}(y|\vec{x}) \\
&= \operatorname*{argmax}_y p_Y(\hat{y}) p_{\vec{X}|Y}(\vec{x}|y)
\end{aligned}$$

The four Bayesian probabilities

- The **posterior** and **evidence**, $p_{Y|\vec{X}}(y|\vec{x})$ and $p_{\vec{X}}(\vec{x})$, can only be learned if you have lots and lots of training data.
- The **prior**, $p_Y(y)$, is very easy to learn.
- The **likelihood**, $p_{\vec{X}|Y}(\vec{x}|y)$, is easier to learn than the posterior, but still somewhat challenging. This lecture is about learning the likelihood.

Outline

Training Data

A **training dataset** is a set of examples, $\mathcal{D} = \{(\vec{x}_0, y_0), \ldots, (\vec{x}_{n-1}, y_{n-1})\}$, from which you want to learn $p_{\vec{X}|Y}(\vec{x}|y)$.

Parametric Estimation

Parametric estimation means we assume that $p_{\vec{X}|Y}(\vec{x}|y)$ has some parametric functional form, with some learnable parameters, Θ. For example, in a Gaussian classifier,

$$\Theta = \{\vec{\mu}_y, \Sigma_y : y \in \{0 \ldots N_Y - 1\}\}$$

and the parametric form is

$$p_{\vec{X}|Y}(\vec{x}|y) = \frac{1}{(2\pi)^{D/2}|\Sigma_y|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_y)^T\Sigma_y^{-1}(\vec{x}-\vec{\mu}_y)}$$

Maximum Likelihood Estimation

Maximum likelihood estimation finds the parameters that maximize the likelihood of the data.

$$\hat{\Theta}_{ML} = \operatorname{argmax} p\left(\mathcal{D}|\Theta\right)$$

Usually we assume that the data are sampled independently and identically distributed, so that

$$\begin{aligned}\hat{\Theta}_{ML} &= \operatorname{argmax} \prod_{i=0}^{n-1} p_{\vec{X}|Y}(\vec{x}_i|y_i) \\ &= \operatorname{argmax} \sum_{i=0}^{n-1} \ln p_{\vec{X}|Y}(\vec{x}_i|y_i)\end{aligned}$$

Example: Gaussians

For example, let's assume Gaussian likelihoods:

$$
\begin{aligned}
\hat{\Theta}_{ML} &= \operatorname{argmax} \prod_{i=0}^{n-1} p_{\vec{X}|Y}(\vec{x}_i|y_i) \\
&= \operatorname{argmax} \sum_{i=0}^{n-1} \ln p_{\vec{X}|Y}(\vec{x}_i|y_i) \\
&= \operatorname{argmin} \sum_{i=0}^{n-1} \left(\ln|\Sigma_{y_i}| + (\vec{x}_i - \vec{\mu}_{y_i})^T \Sigma_{y_i}^{-1} (\vec{x}_i - \vec{\mu}_{y_i}) \right)
\end{aligned}
$$

Example: Gaussians

$$\hat{\Theta}_{ML} = \text{argmin} \sum_{i=0}^{n-1} \left(\ln |\Sigma_{y_i}| + (\vec{x}_i - \vec{\mu}_{y_i})^T \Sigma_{y_i}^{-1} (\vec{x}_i - \vec{\mu}_{y_i}) \right)$$

If we differentiate, and set the derivative to zero, we get

$$\hat{\mu}_{y,ML} = \frac{1}{n_y} \sum_{i:y_i=y} \vec{x}_i$$

$$\hat{\Sigma}_{y,ML} = \frac{1}{n_y} \sum_{i:y_i=y} (\vec{x}_i - \vec{\mu}_y)(\vec{x}_i - \vec{\mu}_y)^T$$

where n_y is the number of tokens from class $y_i = y$.

Outline

Hidden or Unobserved Variables

Many real-world problems have **hidden** or **unobserved** random variables.

If there are hidden variables, we can imagine that the training dataset is divided into two parts: $\mathcal{D}_v$ is the visible part (the variables whose values we know), and $\mathcal{D}_h$ is the hidden part (the variables we don't know).

ML estimation now needs to find

$$\begin{aligned}\hat{\Theta}_{ML} &= \underset{\Theta}{\operatorname{argmax}}\, p\left(\mathcal{D}_v|\Theta\right)\\ &= \underset{\Theta}{\operatorname{argmax}} \sum_{\mathcal{D}_h} p\left(\mathcal{D}_v, \mathcal{D}_h|\Theta\right)\end{aligned}$$

Example: Missing Data

For example, suppose that the training dataset only has two tokens, $\mathcal{D} = \{\vec{x}_0, \vec{x}_1\}$. Each vector should contain D measurements, $\vec{x}_i = [x_{i,0}, \ldots, x_{i,D-1}]^T$. Unfortunately, due to mechanical equipment failure, we are missing the measurements of $x_{0,16}$ and $x_{1,2}$.
The visible and hidden training datasets are:

$$\mathcal{D}_v = \{x_{0,0}, \ldots, x_{0,15}, x_{0,17}, \ldots, x_{1,1}, x_{1,3}, \ldots, x_{1,D-1}\}$$
$$\mathcal{D}_h = \{x_{0,16}, x_{1,2}\}$$

. . . and the ML parameters are:

$$\hat{\Theta}_{ML} = \underset{\Theta}{\operatorname{argmax}} \int \int \Pr\{\mathcal{D}|\Theta\}\, dx_{0,16} dx_{1,2}$$

Example: Mixture Models

The more relevant case (the reason we really care about the expectation maximization algorithm) is the mixture-density situation, for example, Gaussian mixture models.
Remember the pdf model for a GMM:

$$p_{\vec{X}|Y}(\vec{x}|y) = \sum_{k=0}^{N_K-1} c_{y,k}\mathcal{N}(\vec{x}|\vec{\mu}_{y,k}, \Sigma_{y,k})$$

...where, in order to make sure that $1 = \int p_{\vec{X}|Y}(\vec{x}|y)d\vec{x}$, we have to make sure that

$$c_{y,k} \geq 0 \quad \text{and} \quad \sum_k c_{y,k} = 1$$

Example: Mixture Models

$$p_{\vec{X}|Y}(\vec{x}|y) = \sum_{k=0}^{N_K-1} c_{y,k}\mathcal{N}(\vec{x}|\vec{\mu}_{y,k}, \Sigma_{y,k})$$

Think about what's going on when we generate $\vec{x}_i$ from y_i:

- First, we pick a cluster k_i, according to the probability distribution

$$p_{K|Y}(k|y) = c_{y,k} \quad \text{where} \quad c_{y,k} \geq 0 \quad \text{and} \quad \sum_k c_{y,k} = 1$$

- Second, we generate the observation vector from the chosen cluster:

$$p_{\vec{X}|K,Y}(\vec{x}|k, y) = \mathcal{N}(\vec{x}|\vec{\mu}_{y,k}, \Sigma_{y,k})$$

Example: Mixture Models

We don't have any labels to tell us which cluster corresponds to each training token, so the cluster labels are hidden.
The visible and hidden training datasets are:

$$\mathcal{D}_v = \{(\vec{x}_0, y_0), \ldots, (\vec{x}_{n-1}, y_{n-1})\}$$
$$\mathcal{D}_h = \{k_0, \ldots, k_{n-1}\}$$

Example: Mixture Models

The maximum likelihood parameters are:

$$\hat{\Theta}_{ML} = \underset{\Theta}{\operatorname{argmax}} \sum_{\mathcal{D}_h} \Pr\left\{\mathcal{D}_v, \mathcal{D}_h | \Theta\right\}$$

$$= \underset{}{\operatorname{argmax}} \sum_{i=0}^{n-1} \ln \sum_{k=0}^{N_K-1} c_{y_i,k} \mathcal{N}(\vec{x}_i | \vec{\mu}_{y_i,k}, \Sigma_{y_i,k})$$

The Problem with Missing Data

$$
\begin{aligned}
\hat{\Theta}_{ML} &= \underset{\Theta}{\text{argmax}} \sum_{\mathcal{D}_h} \Pr\left\{\mathcal{D}_v, \mathcal{D}_h | \Theta\right\} \\
&= \text{argmax} \sum_{i=0}^{n-1} \ln \sum_{k=0}^{N_K-1} c_{y_i,k} \mathcal{N}(\vec{x}_i | \vec{\mu}_{y_i,k}, \Sigma_{y_i,k})
\end{aligned}
$$

The problem with mixture models is the same as the problem with any type of missing data:

- The log of a sum cannot be simplified.
- Therefore, differentiating the log of a sum usually results in a complicated equation that has no closed-form solution.
- In fact, the solution is usually not even unique.

Outline

The Problem with Missing Data

- Standard ML estimation works really well because we use logarithms to turn the product into a sum:

$$\hat{\Theta}_{ML} = \text{argmax} \prod_{i=0}^{n-1} p_{\vec{X}|Y}(\vec{x}_i|y_i)$$
$$= \text{argmax} \sum_{i=0}^{n-1} \ln p_{\vec{X}|Y}(\vec{x}_i|y_i)$$

- But suppose that you also need to estimate some hidden variable, k. Then you need a sum of logs of sums:

$$\hat{\Theta}_{ML} = \text{argmax} \prod_{i=0}^{n-1} \sum_{k} p_{\vec{X},K|Y}(\vec{x}_i, k|y_i)$$
$$= \text{argmax} \sum_{i=0}^{n-1} \ln \sum_{k} p_{\vec{X},K|Y}(\vec{x}_i, k|y_i)$$

The Problem with Missing Data

Let's write it like this:

$$\hat{\Theta}_{ML} = \operatorname{argmax} \mathcal{L}(\Theta),$$

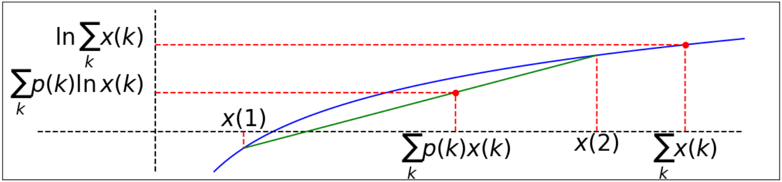
where $\mathcal{L}(\Theta)$ is the log likelihood of the training data:

$$\begin{aligned}\mathcal{L}(\Theta) &= \ln p\left(\mathcal{D}_v|\Theta\right)\\ &= \ln \sum_{\mathcal{D}_h} p\left(\mathcal{D}_v, \mathcal{D}_h|\Theta\right)\end{aligned}$$

Solution: The EM Inequality

Expectation Maximization uses the idea that the **log of a sum** is greater than or equal to the **average of the logs**. For any set of positive numbers $x(k)$, if you can define a pmf such that $\sum_k p(k) = 1$, then

$$\ln \sum_k x(k) \geq \ln \max_k x(k) \geq \sum_k p(k) \ln x(k)$$



Solution: The EM Inequality

Let's make the following definitions:

$$x(\mathcal{D}_h) = p(\mathcal{D}_v, \mathcal{D}_h|\Theta)$$
$$p(\mathcal{D}_h) = p(\mathcal{D}_h|\mathcal{D}_v, \hat{\Theta}),$$

where Θ and $\hat{\Theta}$ can be any two estimates of the parameters. Then the EM inequality says

$$\ln \sum_k x(k) \geq \sum_k p(k) \ln x(k)$$

or

$$\ln \sum_{\mathcal{D}_h} p(\mathcal{D}_v, \mathcal{D}_h|\Theta) \geq \sum_{\mathcal{D}_h} p(\mathcal{D}_h|\mathcal{D}_v, \hat{\Theta}) \ln p(\mathcal{D}_v, \mathcal{D}_h|\Theta)$$

The Q Function

The name "expectation" in "expectation maximization" comes from the lower bound on the previous slide. That lower bound is usually called the "Q function." It looks like this:

$$
\begin{aligned}
Q(\Theta,\hat{\Theta}) &= \sum_{\mathcal{D}_h} p(\mathcal{D}_h|\mathcal{D}_v,\hat{\Theta}) \ln p(\mathcal{D}_v,\mathcal{D}_h|\Theta) \\
&= E\left[\ln p(\mathcal{D}_v,\mathcal{D}_h|\Theta) \middle| \mathcal{D}_v,\hat{\theta}\right]
\end{aligned}
$$

The word "maximization" comes from the following idea: since $\mathcal{L}(\Theta) \ge Q(\Theta,\hat{\Theta})$, how about if we choose

$$
\Theta^* = \underset{\Theta}{\operatorname{argmax}}\, Q(\Theta,\hat{\Theta})
$$

The Expectation Maximization Algorithm

The expectation maximization algorithm has the following steps:

Initialize: Find the best initial guess, Θ^*, that you can.

Iterate: Repeat the following steps. Set $\hat{\Theta} = \Theta^*$, then

- **E-Step:** Compute the posterior probabilities of the hidden variables

$$p(\mathcal{D}_h | \mathcal{D}_v, \hat{\Theta})$$

- **M-Step:** Find new values of Θ that maximize $Q(\Theta, \hat{\Theta})$:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}}\, Q(\Theta, \hat{\Theta})$$

Terminate: If Θ^* does not change from one iteration to the next, it means you have reached a local maximum of both Q and $\mathcal{L}$:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}}\, \mathcal{L}(\Theta) = \underset{\Theta}{\operatorname{argmax}}\, Q(\Theta, \Theta)$$

Outline

EM for Gaussian Mixture Models

For a Gaussian mixture model,

- The observed dataset includes the labels, and the feature vectors:

$$\mathcal{D}_v = \{(\vec{x}_0, y_0), \ldots, (\vec{x}_{n-1}, y_{n-1})\}$$

- The hidden dataset is the cluster identity labels:

$$\mathcal{D}_h = \{k_0, \ldots, k_{n-1}\}$$

E-Step for Gaussian Mixture Models

For a Gaussian mixture model, the E-step probability is

$$p(\mathcal{D}_h|\mathcal{D}_v,\Theta) = p_{K|\vec{X},Y}(k|\vec{x},y)$$
$$= \frac{p_{K|Y}(k|y)p_{\vec{X}|K,Y}(\vec{x}|k,y)}{\sum_\ell p_{K|Y}(\ell|y)p_{\vec{X}|K,Y}(\vec{x}|\ell,y)}$$

In order to solve the last equation, we make these substitutions:

$$p_{K|Y}(k|y) = c_{y,k}$$
$$p_{\vec{X}|Y,K}(\vec{x}|y,k) = \mathcal{N}(\vec{x}|\vec{\mu}_{y,k},\Sigma_{y,k})$$

which gives us something that's often called the "gamma probability:"

$$p_{K|\vec{X},Y}(k|\vec{x}_i,y_i) = \gamma_i(k) = \frac{c_{y_i,k}\mathcal{N}(\vec{x}_i|\vec{\mu}_{y_i,k},\Sigma_{y_i,k})}{\sum_\ell c_{y_i,\ell}\mathcal{N}(\vec{x}_i|\vec{\mu}_{y_i,\ell},\Sigma_{y_i,\ell})}$$

M-Step for Gaussian Mixture Models

For a Gaussian mixture model, the Q function is

$$E_{\mathcal{D}_h}\left[\ln p(\mathcal{D}_h, \mathcal{D}_v, \Theta)\right] = E_k\left[\ln p_{K,\vec{X},Y}(k, \vec{x}, y)\right]$$
$$= E_k\left[\ln p_Y(y) + \ln c_{y,k} + \ln \mathcal{N}(\vec{x}|\vec{\mu}_{y,k}, \Sigma_{y,k})\right]$$

$$= \ln p_Y(y) - \frac{D}{2}\ln(2\pi)$$
$$+ \sum_k \gamma_i(k)\left(\ln c_{y,k} - \frac{1}{2}\left(\ln|\Sigma_{y,k}| + (\vec{x}_i - \vec{\mu}_{y,k})^T \Sigma_y^{-1}(\vec{x}_i - \vec{\mu}_{y,k})\right)\right)$$

M-Step for Gaussian Mixture Models

Maximizing the Q function gives

$$p_Y(y) = \frac{n_y}{n}, \qquad c_{y,k} = \frac{n_{y,k}}{n_y},$$

$$\vec{\mu}_{y,k} = \frac{1}{n_{y,k}} \sum_{i=0}^{n-1} \gamma_i(k) \vec{x}_i,$$

$$\Sigma_{y,k} = \frac{1}{n_{y,k}} \sum_{i=0}^{n-1} \gamma_i(k) (\vec{x}_i - \vec{\mu}_{y,k})(\vec{x}_i - \vec{\mu}_{y,k})^T,$$

where the "soft counts" are the sums of the gamma probabilities, across all tokens

$$n_{y,k} = \sum_{i:y_i=y} \gamma_i(k)$$

Outline

Summary

- Maximum likelihood estimation finds model parameters that maximize the log likelihood:

$$\Theta = \mathrm{argmax}\, \mathcal{L}(\Theta)$$

- Expectation maximization finds model parameters that maximize the expected log likelihood:

$$\Theta = \mathrm{argmax}\, Q(\Theta, \hat{\Theta})$$

- Applying EM to a GMM gives:

$$c_{y,k} = \frac{n_{y,k}}{n_y}$$

$$\vec{\mu}_{y,k} = \frac{1}{n_{y,k}} \sum_{i=0}^{n-1} \gamma_i(k) \vec{x}_i$$

$$\Sigma_{y,k} = \frac{1}{n_{y,k}} \sum_{i=0}^{n-1} \gamma_i(k) (\vec{x}_i - \vec{\mu}_{y,k})(\vec{x}_i - \vec{\mu}_{y,k})^T$$