QCB408 Week 6 Scribed Notes

Mayisha Mahdiya Sultana

3/19/2020

Summary

The following is a summary of the topics covered in class over Week 6.

- Bayesian inference
  - Summary of Bayesian inference, thus far
  - Estimation
  - Priors
  - Empirical Bayes
- Numerical Methods for Likelihood
  - Expectation-Maximization (EM) Algorithm
  - Markov Chain Monte Carlo (MCMC)
    - Metropolis-Hastings algorithm
    - Gibbs sampling

Bayesian inference

A summary of Bayes theorem

In Week 5, we talked about the framework underlying Bayesian inference, which involves having a prior belief about a parameter, observing data, and then updating the prior belief to achieve a posterior belief. More specifically, we said that a **prior probability distribution** is introduced for an unknown parameter, which is a probability distribution on the unknown parameter that captures one's subjective belief about its possible values. We also said that a **posterior probability distributuon** of the parameter is calculated using Bayes theorem after data is observed. We can also use this posterior distribution to make point estimates and gain analogs of confidence intervals.

Suppose $X_1, X_2, ..., X_n \sim F_\theta$, or $\mathbf{X} \sim F_\theta$, with a prior distribution where $\theta \sim F_\tau$. This means that we are treating the parameter θ to be random and depending on τ. In this case, the posterior distribution is given by

$$f(\theta|\mathbf{X}) = \frac{f(\mathbf{X}|\theta)f(\theta)}{f(\mathbf{X})} = \frac{f(\mathbf{X}|\theta)f(\theta)}{\int f(\mathbf{X}|\theta^*)f(\theta^*)d\theta^*}$$

The term $f(\mathbf{X}|\theta)$ is the likelihood. We can think of Bayes theorem as essentially getting the posterior belief by weighing our likelihood function based on our prior belief.

The denominator in Bayes theorem is the marginal distribution of the data, which is achieved by integrating over the values of θ. Since the denominator does not depend on theta, and because calculating integrals is often computationally difficult, the following is commonly used in inference:

$$f(\theta|\mathbf{X}) \propto f(\mathbf{X}|\theta)f(\theta)$$

In addition, when the denominator is important to calculate but too difficult, techniques like Markov chain Monte Carlo are useful.

Estimation

While we achieve a posterior distribution using Bayes theorem, obtaining point estimates is a useful way of summarizing the distribution. These point estimates can be any value in the distribution – it could be the mean, the median, the maximum, etc.

One common point estimate is the **posterior expectation**, which is the expected value of the posterior distribution:

$$E[\theta|\mathbf{x}] = \int \theta f(\theta|\mathbf{x})d\theta$$

The **posterior interval** is the Bayesian version of a confidence interval – it is the probability that the point estimate of θ lies between two points. The $1-\alpha$ posterior interval is given by:

$$1 - \alpha = \Pr(C_l \leq \theta \leq C_u|\mathbf{x})$$

Another common point estimate is the **maximum a posteriori**, or the **MAP**; this tells us the value of θ at which the posterior probability is the highest.

$$\hat{\theta}_{MAP} = \text{argmax}_\theta \Pr(\theta|\mathbf{x})$$

The goodness of a Bayesian estimator can be measured by a **loss function**. For example, $\mathcal{L}(\theta, \tilde{\theta})$ is a loss function for an estimator $\tilde{\theta}$ if $\mathcal{L}(\theta, \tilde{\theta}) = (\theta - \tilde{\theta})^2$ (sum of squared error) or $\mathcal{L}(\theta, \tilde{\theta}) = |\theta - \tilde{\theta}|$ (absolute error).

The **Bayes risk** is defined as the expected value of the loss function with respect to the posterior.

$$E[\mathcal{L}(\theta, \tilde{\theta})|\mathbf{x}] = \int \mathcal{L}(\theta, \tilde{\theta}) f(\theta|\mathbf{x})d\theta$$

An estimator $\hat{\theta}$ is said to be a **Bayes estimator** if it minimizes the Bayes risk among all estimators.

Classification

The Bayesian analog of hypothesis testing is called classification. To understand this, let's consider $(X_1, X_2, ..., X_n)|\theta \overset{\text{iid}}{\sim} F_\theta$ where $\theta \in \Theta$ and $\theta \sim F_\tau$. Let $\Theta_0, \Theta_1 \subseteq \Theta$ so that $\Theta_0 \cap \Theta_1 = \varnothing$ and $\Theta_0 \cup \Theta_1 = \Theta$. Having observed data $\mathbf{x}$, we want to figure out whether θ belongs to Θ_0 or Θ_1.

Let H be a random variable such that $H = 0$ when $\theta \in \Theta_0$ and $H = 1$ when $\theta \in \Theta_1$. From the prior distribution, we can calculate the **prior probability on** H as $\Pr(H = 0) = \int_{\theta \in \Theta_0} f(\theta)d\theta$ and $\Pr(H = 1) = 1 - \Pr(H = 0)$.

Next, using Bayes theorem, we calculate the **posterior probability on** H.

$$\Pr(H = 0|\mathbf{x}) = \frac{f(\mathbf{x}|H_0)\Pr(H = 0)}{f(\mathbf{x})}$$

$$\Pr(H = 0|\mathbf{x}) = \frac{\int_{\theta\in\Theta_0} f(\mathbf{x}|\theta)f(\theta)d\theta}{\int_{\theta\in\Theta} f(\mathbf{x}|\theta)f(\theta)d\theta}$$

Therefore, $\Pr(H = 1|\mathbf{x}) = 1 - \Pr(H = 0|\mathbf{x})$

After getting a posterior distribution on the two values of H, we need to summarize them. Hence, we calculate the **loss function** $\mathcal{L}(H, \tilde{H})$ as $\mathcal{L}(\tilde{H} = 1, H = 0) = c_I$ and $\mathcal{L}(\tilde{H} = 0, H = 1) = c_{II}$ for some $c_I, c_{II} > 0$.

The **Bayes risk** is therefore

$$E[\mathcal{L}(\theta, \tilde{\theta})|\boldsymbol{x}] = c_I\Pr(\tilde{H} = 1, H = 0) + c_{II}\Pr(\tilde{H} = 0, H = 1)$$

$$\therefore E[\mathcal{L}(\theta, \tilde{\theta})|\boldsymbol{x}] = c_I\Pr(\tilde{H} = 1|H = 0)\Pr(H = 0) + c_{II}\Pr(\tilde{H} = 0|H = 1)\Pr(H = 1)$$

The estimate $\tilde{H}$ minimizes the Bayes risk. This will be

$$\tilde{H} = 1 \text{ when } \Pr(H = 1|\boldsymbol{x}) \geq \frac{c_I}{c_I + c_{II}}$$

and $\tilde{H} = 0$ otherwise.

Priors

The selection of a prior is an important part of Bayesian inference. The fewer data points we have, the more the prior distribution matters. Ideally, we also want a prior distribution that is mathematically convenient to use in our model. Conjugate priors, improper priors and Jeffrey's prior are all commonly-used families of prior distributions.

A **conjugate priors** is prior distribution for a data generating distribution so that the posterior distribution is of the same type (or family) as the prior. Using them makes calculating posterior distributions much easier. Two examples we have talked about in class include the Beta-Bernoulli and the Normal-Normal.

e.g. Beta-Bernoulli

Suppose $\boldsymbol{X}|\mu \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$ and suppose that $p \sim \text{Beta}(\alpha, \beta)$. Then, $f(p) = p^{\alpha-1}(1-p)^{\beta-1}$ is the prior distribution, and we calculate the posterior distribution $f(p|\mathbf{x})$ as follows:

$$\begin{aligned} f(p|\boldsymbol{x}) &\propto L(p; \boldsymbol{x})f(p) \\ &= p^{\sum x_i}(1-p)^{\sum(1-x_i)}p^{\alpha-1}(1-p)^{\beta-1} \\ &= p^{\alpha-1+\sum x_i}(1-p)^{\beta-1+\sum(1-x_i)} \\ &\propto \text{Beta}(\alpha + \sum x_i, \beta + \sum(1-x_i)) \end{aligned}$$

Note that the posterior distribution is a Beta distribution where the new parameters incorporate information from the observations.

e.g. Normal-Normal

Suppose $\boldsymbol{X}|\mu \overset{\text{iid}}{\sim} \text{Normal}(\mu, \sigma^2)$, where σ^2 is known, and suppose that $\mu \sim \text{Normal}(a, b^2)$.

Then it can be shown that $\mu|\boldsymbol{x} \sim \text{Normal}(\text{E}[\mu|\boldsymbol{x}], \text{Var}(\mu|\boldsymbol{x}))$ where

$$\text{E}[\mu|\boldsymbol{x}] = \frac{b^2}{\frac{\sigma^2}{n} + b^2}\overline{x} + \frac{\frac{\sigma^2}{n}}{\frac{\sigma^2}{n} + b^2}a$$

$$\text{Var}(\mu|\boldsymbol{x}) = \frac{b^2\frac{\sigma^2}{n}}{\frac{\sigma^2}{n} + b^2}$$

We can also see a weighted average of the prior and sampling distribution variances.

We use conjugate priors depending on our prior knowledge and mathematical convenience. But what happens when we don't have any prior information? In that case, we can try to insert a **non-informative prior** into our model. The simplest form of a non-informative prior would take the form $f(p) \propto 1$ or $f(p) \propto$ constant. This is a **flat prior**.

If we use a flat prior $f(\theta) \propto c$ with $c > 0$, then $\int f(\theta)d\theta = \infty$. Such priors are known as **improper priors**, and they don't represent probability densities. Nevertheless, sometimes the posterior $f(\mathbf{x}|\theta)f(\theta)$ still yields a probability distribution.

For example, when $\boldsymbol{X}|\mu \overset{\text{iid}}{\sim} \text{Normal}(\mu, \sigma^2)$, where σ^2 is known, and $f(\mu) \propto 1$, then $\int f(\theta)d\theta = \infty$, but

$$f(\theta|\boldsymbol{x}) \propto L(\theta; \boldsymbol{x})f(\theta) \sim \text{Normal}\left(\overline{x}, \sigma^2/n\right)$$

which is a proper probability distribution. In general, improper priors are not a problem as long as the resulting posterior is a well-defined probability distribution.

Although flat priors seem to be successful in being un-subjective, they suffer from the problem of not being **transformation invariant**. This means that a flat prior on a parameter does not imply a flat prior on a transformed version of the parameter. Hence, we introduced another useful prior in class that takes care of these problems by being transformation invariant.

Jeffrey's prior is a prior proportional to the square-root of the Fisher information. Intuitively, Jeffrey's prior allocates more priority to the areas of the likelihood where there is more information.

$$f(\theta) \propto \sqrt{I(\theta)}$$

If $\eta = g(\theta)$, then $f(\eta) \propto \sqrt{I(\eta)}$.

As an example, consider the Bernoulli(p) model. Recall that

$$I(p) = \frac{1}{p(1-p)}$$

Jeffrey's prior would be $f(p) \propto \sqrt{I(p)}$, $\therefore f(p) \propto \frac{1}{\sqrt{p(1-p)}} = p^{-1/2}(1-p)^{-1/2}$ This is equivalent to Beta($\frac{1}{2}, \frac{1}{2}$), which is close to a uniform density.

As we see from this section, Bayesian inference heavily relies on the selection of a robust, non-subjective prior. However, this is extremely difficult when we don't have reliable prior information on the data. A newer technique, empirical Bayes, attempts to tackle this problem.

Empirical Bayes

In the empirical Bayes approach, we use observed data to estimate the prior parameter. For example, in the scenario that $\boldsymbol{X}|\theta \overset{\text{iid}}{\sim} F_\theta$ with prior distribution $\theta \sim F_\tau$, we have to determine values for τ. With empirical Bayes, we will use observed data to estimate the prior on τ.

This is especially useful for high-dimensional data when many parameters are simultaneously drawn from a prior with multiple observations drawn per parameter realization. One downside of the empirical Bayes approach is that overfitting may occur with the parameter. However, this problem is minimized in many modern-day datasets because there is a large amount of data.

First, we integrate out the parameter to obtain

$$f(\boldsymbol{x};\tau) = \int f(\boldsymbol{x}|\theta) f(\theta;\tau) d\theta.$$

An estimation method such as MLE is then applied to estimate τ as $\hat{\tau}$. Then inference is done as usual under the assumption that $\theta \sim f(\theta;\hat{\tau})$.

e.g. Empirical Bayes in a Normal distribution
Suppose that $X_i|\mu_i \sim \text{Normal}(\mu_i, 1)$ for $i = 1, 2, \ldots, n$ where these random variables are independent. In addition, $\mu_i \overset{\text{iid}}{\sim} \text{Normal}(a, b^2)$. Then:

$$f(x_i;a,b) = \int f(x_i|\mu_i) f(\mu_i;a,b) d\mu_i \sim \text{Normal}(a, 1+b^2).$$

$$\hat{a} = \overline{x},\ 1 + \hat{b}^2 = \frac{\sum_{k=1}^{n} (x_k - \overline{x})^2}{n}$$

We then use $\hat{a}$ and $\hat{b}$ in $\text{E}[\mu_i|x_i]$ to find $\hat{\text{E}}[\mu_i|x_i]$.

Numerical Methods for Likelihood

For many statistical models, closed form expressions for the maximum likelihood estimators can not be derived. The maximum likelihood estimates must therefore instead be computed using numerical methods.

Latent variable models

Latent variables are random variables that are present in the model but are not observed. We denote them by Z. we assume that

$$(X_1, Z_1), (X_2, Z_2), \ldots, (X_n, Z_n) \overset{\text{iid}}{\sim} F_{\boldsymbol{\theta}}$$

We are introducing the idea of latent variables here because techniques like the EM algorithm and variational inference involve latent variables.

In Bayesian models, we perform a special case of latent variable models where the unobserved random parameters are latent variables. For example, in the empirical Bayes example, we had observations for n random variables $X_i|\mu_i \sim \text{Normal}(\mu_i, 1)$ for $i = 1, 2, \ldots, n$, and $\mu_i \overset{\text{iid}}{\sim} \text{Normal}(a, b^2)$. In this case, we could view the unobserved parameters $\mu_1, \mu_2, \ldots, \mu_n$ to be latent variables. So, $\boldsymbol{\theta}$ in $(X_1, Z_1), (X_2, Z_2), \ldots, (X_n, Z_n) \overset{\text{iid}}{\sim} F_{\boldsymbol{\theta}}$ would be $\boldsymbol{\theta} = (a, b^2)$.

Mixture Models

The definition and example in this section are inspired by a blog called fiveMinuteStats by Matthew Stephens and by lecture notes from a class taught at Carnegie-Mellon University by Cosma Shalizi (links at the end of this document)

The next idea we need to introduce to motivate the EM algorithm is that of **mixture models**. Mixture models are a subset of latent variable models. In statistics, we often assume that each sample comes from the same unimodal distribution; however, this type of modeling can be too restrictive and may not make intuitive sense in many real-world applications, since our observed data can be more complex. For example, the distribution of observed data may be multimodal – with multiple regions having high probability. We use mixture models to model such complex data. A mixture model is essentially a mixture of k distributions. Following is an example, and then we will formally define a mixture model.

First, an example of a mixture model (Gaussian Mixture Model)

Suppose we are interested in simulating the price of hand sanitizers at suburban mega-stores (like Walmart, Costco, etc) during the COVID-19 pandemic of 2020. Say there are two types of hand sanitizers available on the market: Type A is a 90% alcohol 10% water, "kills all germs"-type, and Type B is a 5% alcohol and 95% glitter, "cute Christmas present"-type. During the pandemic, Type A hand sanitizers were (are) more expensive than Type B, so it makes sense to model the price of Type A sanitizers separately from Type B sanitizers. Hence, we can use a mixture model to model the two price distributions. We will have two **mixture components** in our model – one for Type A sanitizers, and one for Type B.

Let's say that if we choose a hand sanitizer at random, there is a 50% chance of choosing a Type A and 50% of choosing Type B. These are the **mixture proportions**. Assume the price of a Type A sanitizer is normally distributed with mean \$25 and standard deviation \$3 (due to higher demand) and the price of a Type B is normally distributed with a mean \$5 and a standard deviation of \$2. We could simulate the prices P_i as follows:

1. Sample $Z_i \sim \text{Bernoulli}(0.5)$

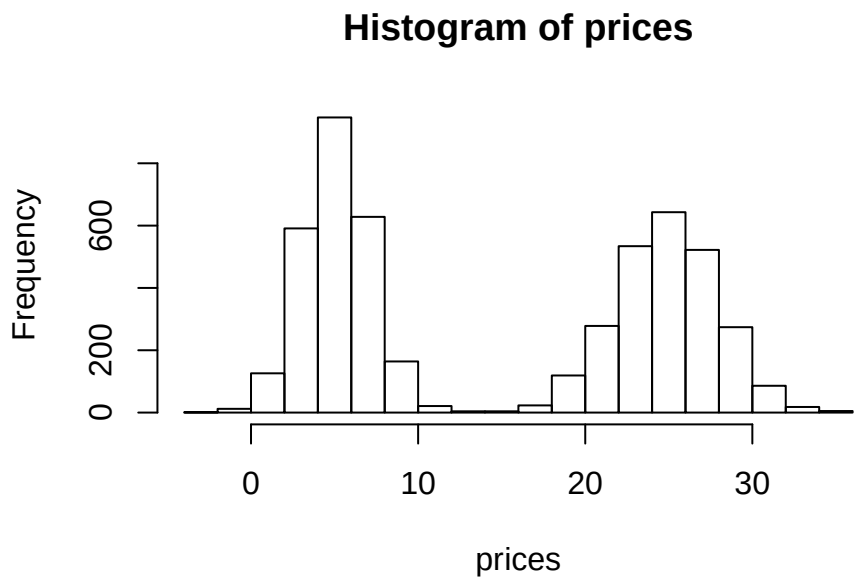
2. If $Z_i = 0$, draw P_i from the Type B distribution $N(5, 2)$. If $Z_i = 1$, draw P_i from the Type A distribution $N(25, 3)$.

The following is a simulation of this process:

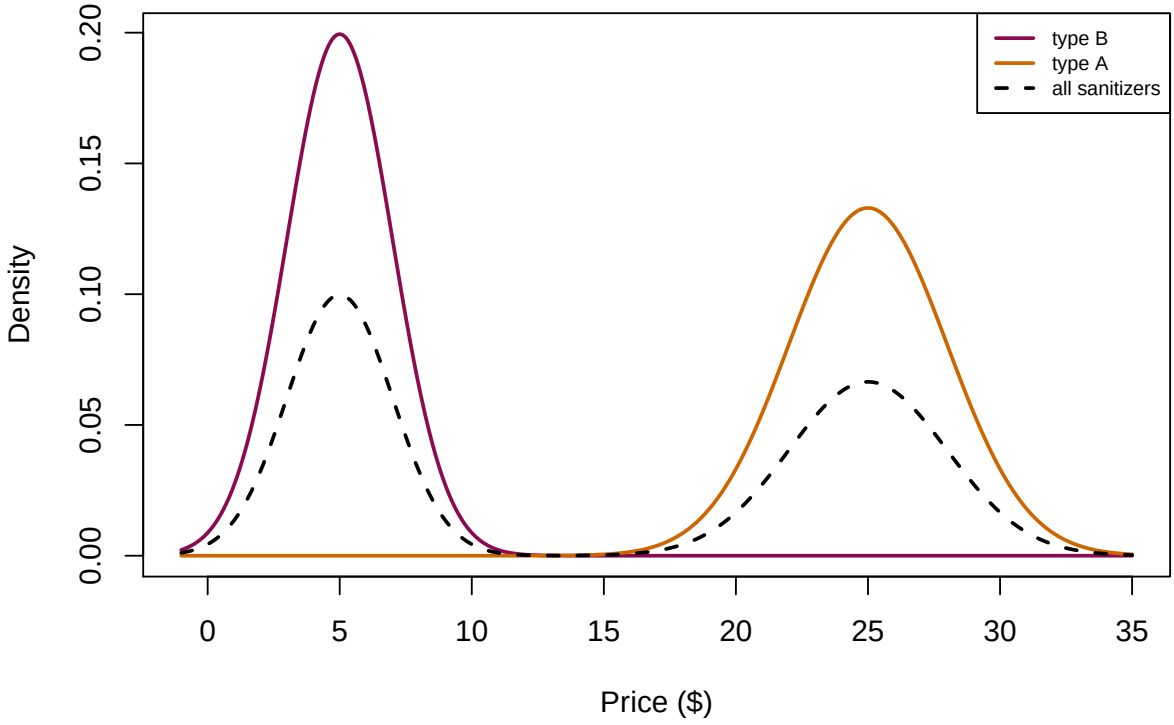
```
# sample n prices of hand sanitizers
n <- 5000
prices <- numeric(n)

# sampling from each of the two distributions
# with proportion 0.5
for(i in seq_len(n)) {
  z.i <- rbinom(1,1,0.5)
  if(z.i == 0) prices[i] <- rnorm(1, mean = 5, sd = 2)
  else prices[i] <- rnorm(1, mean = 25, sd = 3)
}

# histogram of prices sampled
hist(prices)
```



We can see see that our histogram does not look like a Normal distribution – it is bimodal. Even though the mixture components (Type A and Type B prices) are each normal distributions, the distribution of price of a randomly-chosen hand sanitizer is not. The figure below illustrates the true densities of Type A and Type B, and the density of our observed prices.



From the figure, we observe the true distributions of each type of hand sanitizer, and notice that the distribution of randomly-selected hand sanitizers is bimodal and not Normal. This mixture model, where each component is a Gaussian distribution, is called a Gaussian mixture model(GMM).

Formally, we say that a distribution f is a mixture of K component distributions $f_1, f_2, \ldots, f_K$ if

$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x; \theta_k)$$

with π_k being the mixture proportions, $\pi_k > 0$, and $f_k(x; \theta_k)$ is the k^{th} probability distribution.

A formal definition of mixture models

Assume we observe $X_1, \ldots, X_n$ and that each X_i is sampled from one of K mixture components. Z is a variable such that $Z_i \in \{1, \ldots, K\}$, and it indicates which component X_i came from. In our example of hand sanitizers, Z_i would be either 1 or 2 depending on whether X_i was a type A or type B hand sanitizer. When we don't observe Z_i's (in most cases), the Z_i's are the latent variables.

From the law of total probability, we know that the marginal probability of X_i is:

$$P(X_i = x) = \sum_{k=1}^{K} P(X_i = x | Z_i = k) \underbrace{P(Z_i = k)}_{\pi_k} = \sum_{k=1}^{K} P(X_i = x | Z_i = k) \pi_k$$

Here, the π_k are the mixture proportions, i.e. the probability that X_i belongs to the k-th mixture component. In addition, $\sum_{k=1}^{K} \pi_k = 1$. $P(X_i | Z_i = k)$ is mixture component, representing the distribution of X_i assuming it came from component k. The mixture components in our example above were Normal distributions.

Therefore, the pdf of the overall distribution is

$$f_x(x) = \sum_{k=1}^{K} \pi_k f_{x|Z_k}(x \mid Z_k)$$

Remember that the likelihood is the probability (or probability density) of observing our data, as a function of the parameters. If we observe independent samples $X_1, \ldots, X_n$ from this mixture, with mixture proportion vector $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$, then the likelihood function is:

$$L(\pi) = \prod_{i=1}^{n} P(X_i|\pi) = \prod_{i=1}^{n} \sum_{k=1}^{K} P(X_i|Z_i = k)\pi_k$$

If we have k Gaussian models where each component is $N(\mu_k, \sigma_k)$, and the mixture proportions are π_k, then the joint pdf in such a case would be

$$f(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left[\pi_k \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left\{ -\frac{(x_i - \mu_k)^2}{2\sigma_k^2} \right\} \right]^{z_{ik}}$$

Our goal is to estimate the parameters $\{\pi_k, \mu_k, \sigma_k\}$ in order to determine which of our observations $X_1, X_2, \ldots, X_n$ came from which distribution. However, some likelihood functions (like the one above) are extremely difficult to deal with using methods such as maximum likelihood estimation. Hence we need to numerically estimate $\{\pi_k, \mu_k, \sigma_k\}$, and the Expectation-Maximization algorithm is one of the methods that tackles this problem.

Expectation-maximization (EM) algorithm

Parts of this section are taken from All of Statistics by Larry Wasserman and a blog post in fiveMinuteStats by Matt Bonakdarpour

The idea in the EM algorithm is to iterate between taking an expectation and maximizing a likelihood. Suppose we have data X whose density $f(x;\theta)$ leads to a log likelihood that is hard to maximize. If we can find another random variable Z such that $f(x;\theta) = \int f(x,z;\theta)dz$ and the likelihood based on $f(x,z;\theta)$ is easier to maximize, then the model of interest is the marginal of a model with a simpler likelihood. In this case, Z is the missing data, and if we could just "fill in" the z's, the problem would be simplified. The EM algorithm essentially iterates between "filling in" the missing values and maximizing the likelihood function until the underlying parameter stops changing too much.

Let's try to make sense of the algorithm in terms of the Gaussian mixture model example, and then we will define it formally.

First, an example of EM in a Gaussian mixture model

Suppose we have n observations $X_1, \ldots, X_n$ from K Gaussian distributions, hence our unknowns are $\theta = \{\mu_1, \ldots, \mu_K, \sigma_1, \ldots, \sigma_K, \pi_1, \ldots, \pi_K\}$. The likelihood function is

$$L(\theta|X_1, \ldots, X_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k N(x_i; \mu_k, \sigma_k^2)$$

The log-likelihood is:

$$\ell(\theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \mu_k, \sigma_k^2) \right)$$

Note that summation over the K components "blocks" our log function from being applied to the k Normal densities. Taking a derivation and setting it equal to zero, we have

$$\sum_{i=1}^{n} \frac{1}{\sum_{k=1}^{K} \pi_k N(x_i; \mu_k, \sigma_k)} \pi_k N(x_i; \mu_k, \sigma_k) \frac{(x_i - \mu_k)}{\sigma_k^2} = 0 \tag{1}$$

It is clear that we can't isolate the term μ_k (or σ_k or π_k), hence we can't do straightforward maximum likelihood estimation.

However, if we knew the latent variables Z_i, then we could use it to find estimates of μ_k. Given the observations, we first attempt to compute the posterior distribution of Z_i. Essentially we are asking: having seen these observations, what is the probability that X_i comes from $Z_i = k$?

$$P(Z_i = k|X_i) = \frac{P(X_i|Z_i = k)P(Z_i = k)}{P(X_i)} = \frac{\pi_k N(\mu_k, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k N(\mu_k, \sigma_k)} = \gamma_{Z_i}(k) \tag{1}$$

Now we can rewrite the derivative of the log-likelihood with respect to μ_k, as follows:

$$\sum_{i=1}^{n} \gamma_{Z_i}(k) \frac{(x_i - \mu_k)}{\sigma_k^2} = 0$$

Assuming for a second that $\gamma_{z_i}(k)$ is independent of μ_k in this equation, we can solve for μ_k in this equation to get:

$$\hat{\mu_k} = \frac{\sum_{i=1}^{n} \gamma_{z_i}(k) x_i}{\sum_{i=1}^{n} \gamma_{z_i}(k)} = \frac{1}{N_k} \sum_{i=1}^{n} \gamma_{z_i}(k) x_i \tag{2}$$

.

Where we set $N_k = \sum_{i=1}^{n} \gamma_{z_i}(k)$. We can think of N_k as the effective number of points assigned to distribution k. We see that $\hat{\mu_k}$ is therefore a weighted average of the data with weights $\gamma_{z_i}(k)$. Similarly, if we apply a similar method to finding $\hat{\sigma_k^2}$ and $\hat{\pi_k}$, we find that:

$$\hat{\sigma_k^2} = \frac{1}{N_k} \sum_{i=1}^{n} \gamma_{z_i}(k)(x_i - \mu_k)^2 \tag{3}$$

$$\hat{\pi_k} = \frac{N_k}{n} \tag{4}$$

Remember, though, that $\gamma_{Z_i}(k)$ depends on the unknown parameters, so these equations are not closed-form expressions. If we knew the parameters, we could compute the posterior probabilities $\gamma_{Z_i}(k)$. But to compute the parameters, we need the posteriors $\gamma_{Z_i}(k)$! However, this is where the EM algorithm comes in.

1. Initialize the μ_k's, σ_k's and π_k's and evaluate the log-likelihood with these parameters.
2. **E-step:** Evaluate the posterior probabilities $\gamma_{Z_i}(k)$ using the current values of the μ_k's and σ_k's using equation (1).
3. **M-step:** Estimate new parameters $\hat{\mu_k}$, $\hat{\sigma_k^2}$ and $\hat{\pi_k}$ with the current values of $\gamma_{Z_i}(k)$ using equations (2), (3) and (4).
4. Evaluate the log-likelihood with the new parameter estimates. If the log-likelihood has changed by less than a small ϵ, stop. Otherwise, go back to step 2.

This was roughly an idea of how the EM algorithm works. Now, let's define it more formally.

A formal definition of the EM algorithm

If X is the entire set of observed variables and Z the entire set of latent variables, the log-likelihood is given by:

$$\log\left(P(X|\Theta)\right) = \log\left(\sum_{Z} P(X, Z|\Theta)\right)$$

where we've marginalized Z out of the joint distribution.

Now suppose that we observed the **complete dataset**, i.e. we observe both X and Z. As we noted previously, if we knew Z, the maximization would be easy.

The information we have about Z is contained in the posterior $P(Z|X, \Theta)$. Since we don't know the complete log-likelihood, we consider its expectation under the posterior distribution of the latent variables. This corresponds to the *E-step*. In the *M-step*, we maximize this expectation to find a new estimate for the parameters.

In the E-step, we use the current value of the parameters θ^0 to find the posterior distribution of the latent variables given by $P(Z|X, \theta^0)$. This corresponds to the $\gamma_{Z_i}(k)$ in the previous section. We then use this to find the expectation of the complete data log-likelihood, with respect to this posterior, evaluated at an arbitrary θ. This expectation is denoted $Q(\theta, \theta^0)$ and it equals:

$$Q(\theta, \theta^0) = E_{Z|X,\theta^0}\left[\log(P(X, Z|\theta))\right] = \sum_{Z} P(Z|X, \theta^0) \log(P(X, Z|\theta))$$

In the M-step, we determine the new parameter $\hat{\theta}$ by maximizing Q:

$$\hat{\theta} = \text{argmax}_{\theta} Q(\theta, \theta^0)$$

Going back to our Gaussian mixture model example, the complete log-likelihood takes the form:

$$\log f(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \pi_k + z_{ik} \log \phi(x_i; \mu_k, \sigma_k^2)$$

where

$$\phi(x_i; \mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left\{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}\right\}.$$

In calculating

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\boldsymbol{Z}|\boldsymbol{X}=\boldsymbol{x}}\left[\log f(\boldsymbol{x}, \boldsymbol{Z}; \boldsymbol{\theta}); \boldsymbol{\theta}^{(t)}\right]$$

we only need to know $\mathrm{E}_{\boldsymbol{Z}|\boldsymbol{X}=\boldsymbol{x}}[Z_{ik}|\boldsymbol{x}; \boldsymbol{\theta}]$, which turns out to be

$$\mathrm{E}_{\boldsymbol{Z}|\boldsymbol{X}=\boldsymbol{x}}[Z_{ik}|\boldsymbol{x}; \boldsymbol{\theta}] = \frac{\pi_k \phi(x_i; \mu_k, \sigma_k^2)}{\sum_{j=1}^K \pi_j \phi(x_i; \mu_j, \sigma_j^2)}.$$

Note that we take

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\boldsymbol{Z}|\boldsymbol{X}=\boldsymbol{x}}\left[\log f(\boldsymbol{x}, \boldsymbol{Z}; \boldsymbol{\theta}); \boldsymbol{\theta}^{(t)}\right]$$

so the parameter in $\log f(\boldsymbol{x}, \boldsymbol{Z}; \boldsymbol{\theta})$ is a free $\boldsymbol{\theta}$, but the paramaters used to take the conditional expectation of $\boldsymbol{Z}$ are fixed at $\boldsymbol{\theta}^{(t)}$. Now, we calculate $\hat{z}_{ik}^{(t)}$:

$$\hat{z}_{ik}^{(t)} = \mathrm{E}\left[z_{ik}|\boldsymbol{x}; \boldsymbol{\theta}^{(t)}\right] = \frac{\pi_k^{(t)} \phi(x_i; \mu_k^{(t)}, \sigma_k^{2,(t)})}{\sum_{j=1}^K \pi_j^{(t)} \phi(x_i; \mu_j^{(t)}, \sigma_j^{2,(t)})}.$$

Then, we move on to the E-step:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\boldsymbol{Z}|\boldsymbol{X}=\boldsymbol{x}}\left[\log f(\boldsymbol{x}, \boldsymbol{Z}; \boldsymbol{\theta}); \boldsymbol{\theta}^{(t)}\right]$$

$$= \sum_{i=1}^n \sum_{k=1}^K \hat{z}_{ik}^{(t)} \log \pi_k + \hat{z}_{ik}^{(t)} \log \phi(x_i; \mu_k, \sigma_k^2)$$

We can now caculate $\boldsymbol{t}^{(t+1)} = \mathrm{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}$, which gives:

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^n \hat{z}_{ik}^{(t)}}{n}$$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \hat{z}_{ik}^{(t)} x_i}{\sum_{i=1}^n \hat{z}_{ik}^{(t)}}$$

$$\sigma_k^{2,(t+1)} = \frac{\sum_{i=1}^n \hat{z}_{ik}^{(t)} \left(x_i - \mu_k^{(t+1)}\right)^2}{\sum_{i=1}^n \hat{z}_{ik}^{(t)}}$$

We can then use these parameter estimates to compute the log-likelihood and repeat the process until convergence.

Note the caveat that if we assign only one data point to mixture component k, meaning $\mu_k^{(t)} = x_i$ and $\hat{z}_{ik}^{(t)} = 1$ for some k and i, then as $\sigma_k^{2,(t)} \to 0$, the likelihood goes to ∞. Therefore, when implementing the EM algorithm for this particular Normal mixture model, we have to be careful to bound all $\sigma_k^{2,(t)}$ away from zero and avoid this scenario.

Markov Chain Monte Carlo

This section is partially adapted from Ramon van Handel's lecture notes for ORF309 at Princeton University, the fiveMinuteStats blog by Matthew Stephens, Ben Shaver's blog on TowardsDataScience, and the lecture notes for this class

The last topic we covered in Week 6 was Markov Chain Monte Carlo. As mentioned above, in Bayesian inference, it is often possible to calculate

$$f(\boldsymbol{\theta}|\boldsymbol{x}) \propto L(\boldsymbol{\theta};\boldsymbol{x})f(\boldsymbol{\theta})$$

but it is typically much more difficult to calculate

$$f(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{L(\boldsymbol{\theta};\boldsymbol{x})f(\boldsymbol{\theta})}{f(\boldsymbol{x})}$$

since $f(\boldsymbol{x})$ is difficult to achieve.

Markov chain Monte Carlo is a method for simulating data approximately from $f(\boldsymbol{\theta}|\boldsymbol{x})$ with knowledge of only $L(\boldsymbol{\theta};\boldsymbol{x})f(\boldsymbol{\theta})$. Before we move on to MCMC, we briefly define the idea of a Markov chain and its stationary distribution. Then, drawing on these definitions, we provide an intuition for MCMC, and look at an MCMC method called the Metropolis-Hastings algorithm.

A brief summary of pre-requisite concepts

A **Markov chain** is a random process whose state tomorrow (at time $n+1$), conditional on the history of the process to date, depends only on the value of the process today (at time n). Hence it is a random process $\{X_n\}_{n>0}$ where each X_n takes values in a finite set D, is called a Markov chain if $P\{X_{n+1} = x_{n+1}|X_n = x_n, \ldots, X_0 = x_0\} = P\{X_{n+1} = x_{n+1}|X_n = x_n\}$ for all $n \geq 0$ and $x_1, \ldots, x_{n+1} \in D$

The **stationary distribution of a Markov chain** describes the distribution of X_n after a sufficiently long time that the distribution of X_n does not change any longer. The stationary state distribution is important because it lets us define the probability for every state of a system at a random time.

Lastly, **Monte Carlo** simulations are a way of estimating a fixed parameter by repeatedly generating random numbers. By taking the random numbers generated and doing some computation on them, Monte Carlo simulations provide an approximation of a parameter where calculating it directly is impossible or prohibitively expensive.

Now, a big picture of MCMC

Essentially, the entire goal of MCMC is to allow one to sample values from a probability distribution, and then compute statistics (like the expected value) on the samples drawn. In Bayesian inference, MCMC methods are used to approximate the posterior distribution of a parameter of interest by random sampling in a probabilistic space.

If we want to draw values from a distribution, we want values that represent high probability areas of the distribution. MCMC methods construct a Markov chain that converges to a stationary distribution that is equivalent to the target probability distribution.

The following is a quote that explains the process very intuitively. It is taken from the blog by Ben Shaver on TowardsDataScience.

> "To begin, MCMC methods pick a random parameter value. The simulation will continue to generate random values (this is the Monte Carlo part), but subject to some rule for determining what makes a good parameter value. The trick is that, for a pair of parameter values, it is possible to compute which is a better parameter value, by computing how likely each value is to explain the data, given our prior beliefs. If a randomly generated parameter value is better than the last one, it is added to the chain of parameter values with a certain probability determined by how much better it is (this is the Markov chain part)."

Therefore, here are the steps we follow:

1. pick a new "proposed" location;
2. figure out how close that location is to an area of high probability compared to your current location;
3. probabilistically stay put or move to that location in a way that respects the overall goal of spending time proportional to the probability of the location.

The Metropolis-Hastings Algorithm

The **Metropolis-Hastings Algorithm** is an implementation of an MCMC method. To implement the MH algorithm, the user must provide a "transition kernel", Q. A transition kernel is a specification of randomly "walking" to a new position in the probability space $\boldsymbol{\theta}$, given a current position ($\boldsymbol{\theta}^{(b)}$). That is, Q is a distribution on $\boldsymbol{\theta}^{(b)}$ given $\boldsymbol{\theta}$, and we will write it as $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(b)})$, for some pdf or pmf. The algorithm simulates a Markov chain whose stationary distribution is π.

The MH algorithm for sampling from a target distribution π, using transition kernel Q, consists of the following steps:

1. Initialize, $\theta^{(0)}$
2. Generate $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(b)})$ for some pdf or pmf $q(\cdot|\cdot)$
3. Compute
$$A(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(b)}) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^*)Q(\boldsymbol{\theta}^{(b)}|\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta}^{(b)})Q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(b)})}\right)$$
$$A(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(b)}) = \min\left(1, \frac{L(\boldsymbol{\theta}^*;\boldsymbol{x})f(\boldsymbol{\theta}^*)Q(\boldsymbol{\theta}^{(b)}|\boldsymbol{\theta}^*)}{L(\boldsymbol{\theta}^{(b)};\boldsymbol{x})f(\boldsymbol{\theta}^{(b)})Q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(b)})}\right)$$
A is often called the "acceptance probabilty". With probability A "accept" the proposed value, and set $\boldsymbol{\theta}^{(b+1)} = \boldsymbol{\theta}^*$. Otherwise, set $\boldsymbol{\theta}^{(b+1)} = \boldsymbol{\theta}^{(b)}$.
4. Continue for $b = 1, 2, \ldots, B$ iterations and carefully select which $\boldsymbol{\theta}^{(b)}$ are utilized to approximate iid observations from $f(\boldsymbol{\theta}|\boldsymbol{x})$

Notice that the example random walk proposal Q given above satisfies $Q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(b)}) = Q(\boldsymbol{\theta}^{(b)}|\boldsymbol{\theta}^*)$ for all $(\boldsymbol{\theta}^{(b)}, \boldsymbol{\theta}^*)$. Any proposal that satisfies this is called "symmetric". When Q is symmetric the formula for A in the MH algorithm simplifies to:
$$A = \min\left(1, \frac{\pi(y)}{\pi(x_t)}\right)$$
or,
$$A(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(b)}) = \min\left(1, \frac{L(\boldsymbol{\theta}^*;\boldsymbol{x})f(\boldsymbol{\theta}^*)}{L(\boldsymbol{\theta}^{(b)};\boldsymbol{x})f(\boldsymbol{\theta}^{(b)})}\right)$$

This special case with Q symmetric, was first presented by Metropolis et al. in 1953, and for this reason it is sometimes called the "Metropolis algorithm".

Two common uses of the output from MCMC are as follows:

1. We take the expected value of draws from the posterior distribution. $\mathrm{E}[f(\boldsymbol{\theta})|\boldsymbol{x}]$ is approximated by

$$\hat{\mathrm{E}}[f(\boldsymbol{\theta})|\boldsymbol{x}] = \frac{1}{B}\sum_{b=1}^{B} f\left(\boldsymbol{\theta}^{(b)}\right).$$

2. We use it as a sampling technique, so some subsequence $\boldsymbol{\theta}^{(b_1)}, \boldsymbol{\theta}^{(b_2)}, \ldots, \boldsymbol{\theta}^{(b_m)}$ from $\left\{\boldsymbol{\theta}^{(b)}\right\}_{b=1}^{B}$ is utilized as an empirical approximation to iid draws from $f(\boldsymbol{\theta}|\boldsymbol{x})$.

The following are a few remarks mentioned in class:

1. The random draw $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(b)})$ perturbs the current value $\boldsymbol{\theta}^{(b)}$ to the next value $\boldsymbol{\theta}^{(b+1)}$. It is often a Normal distribution for continuous $\boldsymbol{\theta}$.
2. Choosing the variance of $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(b)})$ is important as it requires enough variance for the theory to be applicable within a reasonable number of computations, but it cannot be so large that new values of $\boldsymbol{\theta}^{(b+1)}$ are rarely generated.
3. The algorithm must be run for a certain number of iterations ("burn in") before observed $\boldsymbol{\theta}^{(b)}$ can be utilized.
4. The generated $\boldsymbol{\theta}^{(b)}$ are typically "thinned" (only sampled every so often) to reduce Markov dependence.

Lastly, we talked briefly in lecture about **Gibbs sampling**. Gibbs sampling is a special case of the MH algorithm where the algorithm samples one coordinate of θ at a time ($A(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(b)})$). The process is as follows:

1. Initialize $\boldsymbol{\theta}^{(0)}$.
2. Sample:
   $\theta_1^{(b+1)} \sim \Pr(\theta_1|\boldsymbol{\theta}_{2:K}^{(b)}, \boldsymbol{x})$
   $\theta_2^{(b+1)} \sim \Pr(\theta_2|\theta_1^{(b+1)}, \boldsymbol{\theta}_{3:K}^{(b)}, \boldsymbol{x})$
   $\theta_3^{(b+1)} \sim \Pr(\theta_3|\boldsymbol{\theta}_{1:2}^{(b+1)}, \boldsymbol{\theta}_{3:K}^{(b)}, \boldsymbol{x})$
   $\vdots$
   $\theta_K^{(b+1)} \sim \Pr(\theta_K|\boldsymbol{\theta}_{1:K-1}^{(b+1)}, \boldsymbol{x})$
3. Continue for $b = 1, 2, \ldots, B$ iterations.

Sources used

[1] J. D. Storey, Foundations of Applied Statistics, Chapter 46.

[2] R. van Handel, "Probability and Random Processes," p. 202.

[3] C. Shalizi, "36-402, Undergraduate Advanced Data Analysis (2012)." https://www.stat.cmu.edu/~cshalizi/uADA/12/. [Accessed: 23-Mar-2020].

[4] B. Shaver, "A Zero-Math Introduction to Markov Chain Monte Carlo Methods." https://towardsdatascience.com/a-zero-math-introduction-to-markov-chain-monte-carlo-methods-dcba889e0c50. [Accessed: 23-Mar-2020].

[5] M. Stephens, "fiveMinuteStats Blog." https://stephens999.github.io/fiveMinuteStats/index.html. [Accessed: 23-Mar-2020].

Session Information

```
sessionInfo()
```

```
## R version 3.5.3 (2019-03-11)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17134)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_3.5.3  magrittr_1.5    tools_3.5.3     htmltools_0.4.0
##  [5] yaml_2.2.0      Rcpp_1.0.3      stringi_1.4.5   rmarkdown_2.1
##  [9] knitr_1.27      stringr_1.4.0   xfun_0.12       digest_0.6.23
## [13] rlang_0.4.3     evaluate_0.14
```