Chapter 11

Model selection

11.1 Our modelled world

Let us have a set of k models that we will consider in our analyses:

$$\underline{\mathcal{M}} = \{\mathcal{M}_1, \ldots, \mathcal{M}_k\}.$$

To be clear, a model in this context is a joint probability distribution over data and parameters of interest. If we were to specify a different prior for a parameter and keep the same likelihood structure, we would be dealing with a different model. The purpose of our modelling may be to predict what might occur if we were to have more data. For simplicity, let us assume that these future data are denoted $\mathbf{x}$.

Given $\underline{\mathcal{M}}$, we will be operating in one of the following regimes:

(1) M-closed

We believe the true data generating process is in an element of $\underline{\mathcal{M}}$ with the prior distributions set up so as not to rule out the true parameterisation. In this case, we have a simple equation for our predictions of $\mathbf{x}$:

$$\pi(\mathbf{x}) = \sum_{i=1}^{k} \pi\left(\mathbf{x}|\mathcal{M}_i\right) \pi\left(\mathcal{M}_i\right)$$

by the law of total probability. As we collect more data, we will get $\pi\left(\mathcal{M}_j\right)$ getting closer to one if $\mathcal{M}_j$ is the true model.

(2) M-complete

We believe that a true data generating process exists, and, despite us being able to conceptualise it, we cannot put it in a from that makes the model accessible to us for computational purposes. However, in this case, we assume $\underline{\mathcal{M}}$ include the best models that we could currently utilise.

Here, we cannot really use the same formula as in the M-closed scenario because assigning prior probabilities for the models does not really make sense because we know that none of them is

the true model. Therefore, we are restricted to reporting $\pi(\mathbf{x}|\mathcal{M}_i)$ for each of our models and focus shifts to evaluating individual model performance.

(3) M-open

The reality is that we are almost always in a situation where we know the true model is not in $\underline{\mathcal{M}}$. Again, assigning prior probabilities for the models does not make sense, and we are left trying to evaluate predictive performance.

The solutions to handling (2) and (3) are not entirely satisfactory. In the remainder of this chapter, we will be focussing on the M-closed situation.

11.2 Nested models

The ideal situation is for us to have our alternative models contained within an overarching model. Nested models are usually used to describe regression models where simpler models (with fewer explanatory variables and interactions) are within an overarching model containing all variables and possible interactions. In our context, we want all the models in $\underline{\mathcal{M}}$ to be a special case of a all-encompassing model.

Example 11.2.1

Imagine that we have two models:

$\mathcal{M}_1$:

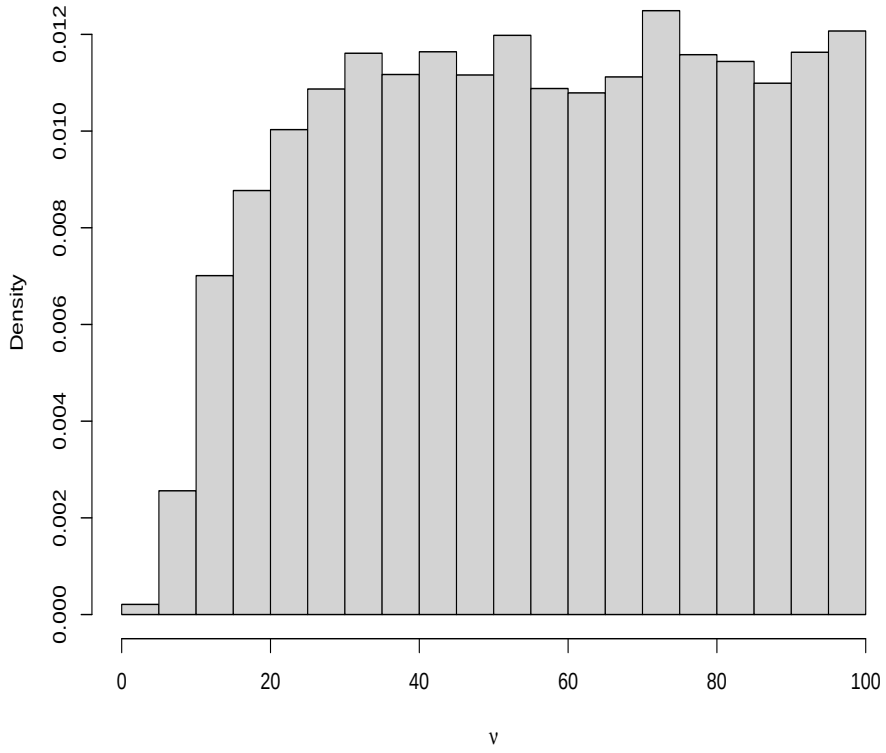
$$\begin{aligned} x_i|\mu,\sigma &\sim \mathsf{N}(\mu,\sigma^2), \quad i=1,\ldots,n, \\ \pi(\mu,\sigma); & \end{aligned}$$

$\mathcal{M}_2$:

$$\begin{aligned} x_i|m,\nu,s &\sim \mathsf{t}_\nu(m,s), \quad i=1,\ldots,n, \\ \pi(m,\nu,s). & \end{aligned}$$

As $\nu \to \infty$, $\mathcal{M}_2 \to \mathcal{M}_1$ (provided that the priors on the parameters are consistent). In fact, by the time $\nu = 30$, we are getting close to parity.

Now, let's imagine that we have collected 100 observations of x that are taken from a normal distribution and we sample from our posterior for ν under $\mathcal{M}_2$. We get the following histogram:



This would seem to give support to $\mathcal{M}_1$ because the ν are mainly favoured over values that would mean that the t-distribution and the normal would be almost indistinguishable.

A direct way to include seemingly non-compatible data-generating models within a single overarching model is to use a mixture distribution:

$$\begin{aligned} z_i|\boldsymbol{\theta} &\sim \text{Categorical}(\boldsymbol{\theta}), \quad i = 1, .., n, \\ x_i|z_1, \boldsymbol{\alpha}_1 &\sim G_1(\boldsymbol{\alpha}_1), \quad i = 1, .., n, \\ &\vdots \\ x_i|z_k, \boldsymbol{\alpha}_k &\sim G_k(\boldsymbol{\alpha}_k), \quad i = 1, .., n, \\ \pi(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k, \boldsymbol{\theta}). & \end{aligned}$$

In this formulation, the z_n are latent variables that indicate which model the data have been drawn from. In practice, we will be uncertain about which model gave rise to the data and we will need to integrate out those variables, and we need to consider the interplay between the different α_j.

Example 11.2.2

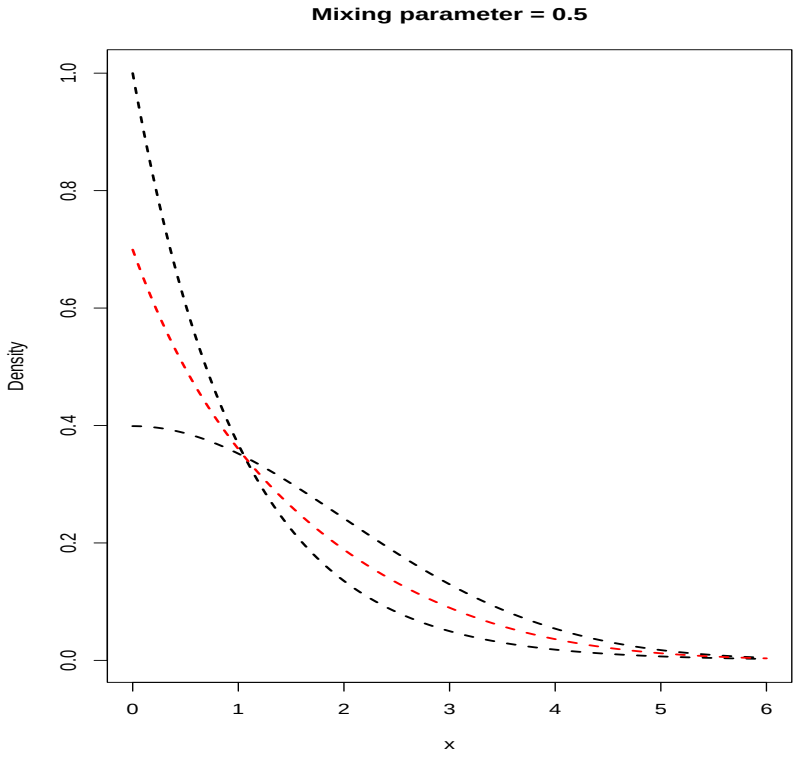
We have two competing models for data that are known to be positive:

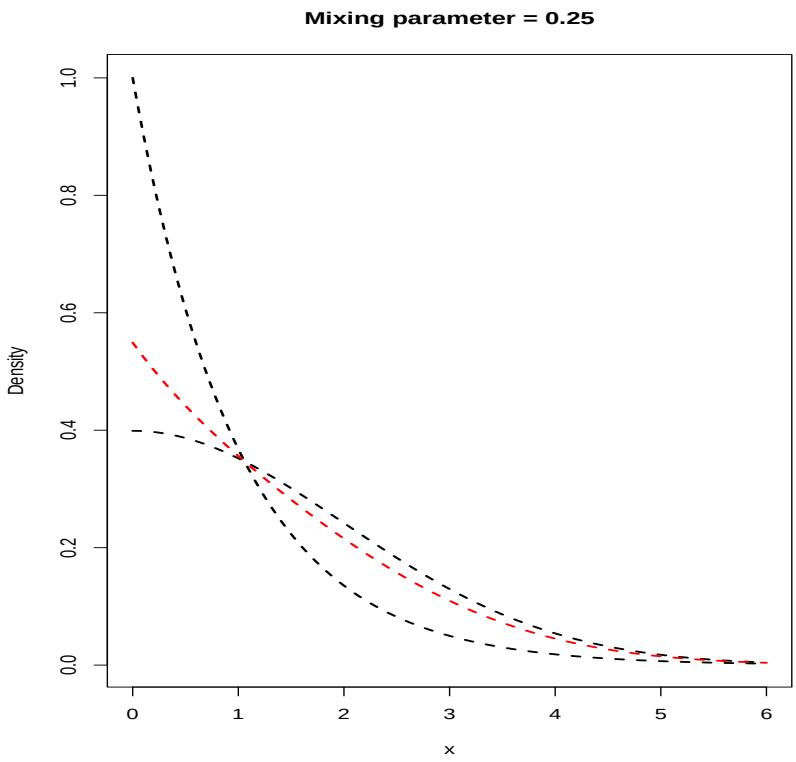
$$\begin{aligned} z_i|\theta &\sim \text{Bernoulli}(\theta), \\ x_i|z_1,\lambda &\sim \text{Exp}(\lambda), \\ x_i|z_2,\sigma &\sim \text{HalfNormal}(\sigma), \quad i=1,..,n. \end{aligned}$$

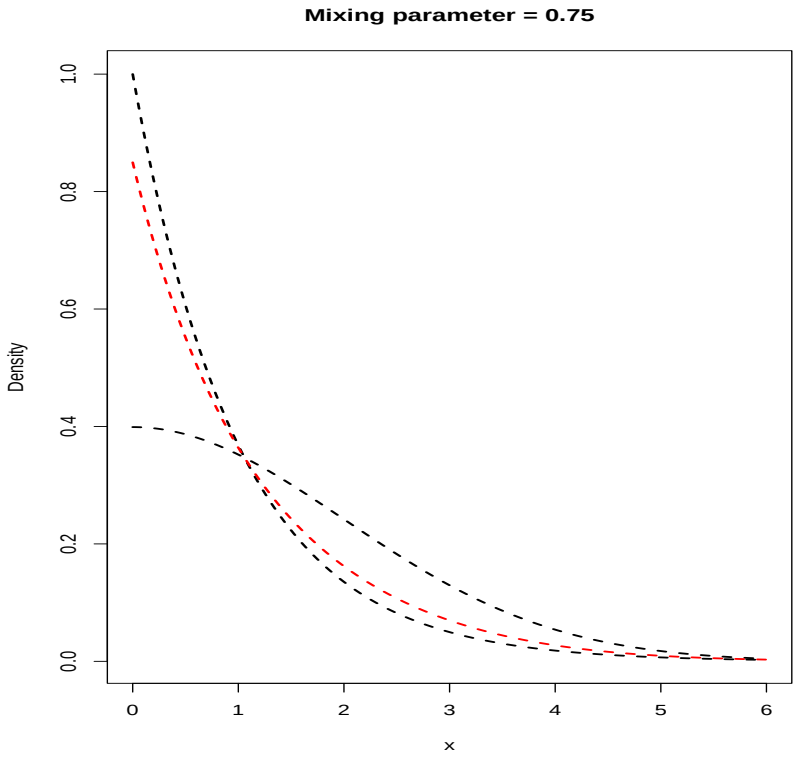
If we let the pdf associated with mixture element j be $\pi_j(x|\text{parameters})$, the likelihood here (dropping the index on x) is

$$\begin{aligned} l(\lambda,\sigma,\theta;x) &\propto \pi(x|\lambda,\sigma,\theta) \\ &= \sum_{j=1}^{2} \pi(x|\lambda,\sigma,z_j)\pi(z_j|\theta) \\ &= \theta\pi_1(x|\sigma,z_j) + (1-\theta)\pi_2(x|\sigma,z_j). \end{aligned}$$

We must be careful about the interpretation of θ in this model. If we find that θ is zero or one, then we can be confident that one of the likelihood forms is favoured over the other. However, it is more likely that we get a distribution for θ that supports values inbetween the two extremes.
Imagine we are mixing a $\text{Exp}(1)$ distribution with a $\text{HalfNormal}(2)$ distribution. Different θ give rise to different shapes, and a model that behaves at odds to the two original distributions.




Mixing parameter = 0.25
Density
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
x


Mixing parameter = 0.75
Density
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
x

11.3 Posterior odds

Throughout the module, we have been assuming some data generating model $\mathcal{M}$, but we have not been explicitly conditioning on it:

$$\pi(\theta \mid \mathbf{x}, \mathcal{M}) \propto \pi(\mathbf{x} \mid \theta, \mathcal{M})\pi(\theta|\mathcal{M}).$$

Recall that, finding the proportionality constant, we get

$$\pi(\theta \mid \mathbf{x}, \mathcal{M}) = \frac{\pi(\mathbf{x} \mid \theta, \mathcal{M})\pi(\theta|\mathcal{M})}{\pi(\mathbf{x}|\mathcal{M})},$$

where $\pi(\mathbf{x}|\mathcal{M})$ is the *evidence* for model $\mathcal{M}$.

If we are in the M-closed situation and we have two competing models for some data $\mathbf{x}$ say: $\mathcal{M}_1$ and $\mathcal{M}_2$. Each model will have its own set of parameters that we will need to deal with: θ_1 and θ_2 say. *A priori*, my odds for $\mathcal{M}_1$ against $\mathcal{M}_2$ are

$$\frac{\pi(\mathcal{M}_1)}{\pi(\mathcal{M}_2)} = \frac{\pi(\mathcal{M}_1)}{1-\pi(\mathcal{M}_1)} = O_f(\mathcal{M}_1).$$

After observing data, my posterior odds for $\mathcal{M}_1$ against $\mathcal{M}_2$ are given by

$$\frac{\pi(\mathcal{M}_1|x)}{\pi(\mathcal{M}_2|\mathbf{x})} = \frac{\pi(\mathcal{M}_1)}{\pi(\mathcal{M}_2)}\frac{\pi(\mathbf{x}|\mathcal{M}_1)}{\pi(\mathbf{x}|\mathcal{M}_2)}$$

or

$$\text{POSTERIOR ODDS} = \text{PRIOR ODDS} \times \text{BAYES FACTOR}.$$

A question remains over how we calculate $\pi(\mathbf{x}|\mathcal{M}_i)$. This is an instance of the preposterior distribution under $\mathcal{M}_i$.

Example 11.3.1

Consider two models:

$$\begin{aligned} \mathcal{M}_1 &: X|\theta \sim \text{Bin}(10,\theta), \quad \theta \sim \text{Be}(2,2); \\ \mathcal{M}_2 &: X|\theta \sim \text{Bin}(20,\theta), \quad \theta \sim \text{Be}(2,2). \end{aligned}$$

I strongly believe that the first model is correct: $\pi(\mathcal{M}_1) = 0.9$. So my prior odds for $\mathcal{M}_1$ are 9. I then observe $x = 10$.

$$\begin{aligned} \pi(x=10|\mathcal{M}_1) &= \int_0^1 \pi(x=10|\theta,\mathcal{M}_1)\pi(\theta|\mathcal{M}_1)d\theta \\ &= \binom{10}{10}\frac{1}{B(2,2)}\int_0^1 \theta^{11}(1-\theta)d\theta \\ &= \frac{B(12,2)}{B(2,2)} \text{ (The previous integral was of a Be(12,2) density.)} \end{aligned}$$

$$= \frac{11! \times 1!}{13!} \frac{3!}{1! \times 1!} = \frac{1}{26}.$$

Similarly, $\pi(x = 10|\mathcal{M}_2) = 11/161$. So my posterior odds for $\mathcal{M}_1$ are

$$\frac{\pi(\mathcal{M}_1|x)}{\pi(\mathcal{M}_2|x)} = 9 \times \frac{161}{11 \times 26} = 5.07 \text{ to 2 d.p.}$$

or my posterior probability for the first model is

$$\pi(\mathcal{M}_1|x) = \frac{5.07}{1 + 5.07} = 0.84.$$

If I had believed that both models were equally likely *a priori*, my posterior odds for the first model would equal the Bayes factor, 0.56, and my posterior probability for the first model would be 0.36.

11.4 Bayesian model averaging

Bayesian model averaging (BMA) takes us back to the equation at the start of this chapter for the M-closed scenario:

$$\pi(\mathbf{x}^*|\mathbf{x}) = \sum_{i=1}^{k} \pi\left(\mathbf{x}^*|\mathbf{x}, \mathcal{M}_i\right) \pi\left(\mathcal{M}_i|\mathbf{x}\right),$$

where we make the distinction between data we have seen, $\mathbf{x}$, and data we might see in the future $\mathbf{x}^*$.

Example 11.4.1

Returning to the previous example and assuming that both models were thought to be equally likely *a priori*, we have

$$\pi\left(\mathcal{M}_1|\mathbf{x}\right) = 0.36 \text{ and } \pi\left(\mathcal{M}_2|\mathbf{x}\right) = 0.64.$$

Under $\mathcal{M}_1$, the predictive probability mass function is

$$\pi(x^*|x) = \binom{10}{x^*} \frac{\mathrm{B}(x^* + 12, 12 - x^*)}{\mathrm{B}(12, 2)}.$$

Similarly, under $\mathcal{M}_2$, the predictive probability mass function is

$$\pi(x^*|x) = \binom{20}{x^*} \frac{\mathrm{B}(x^* + 12, 32 - x^*)}{\mathrm{B}(12, 12)}.$$

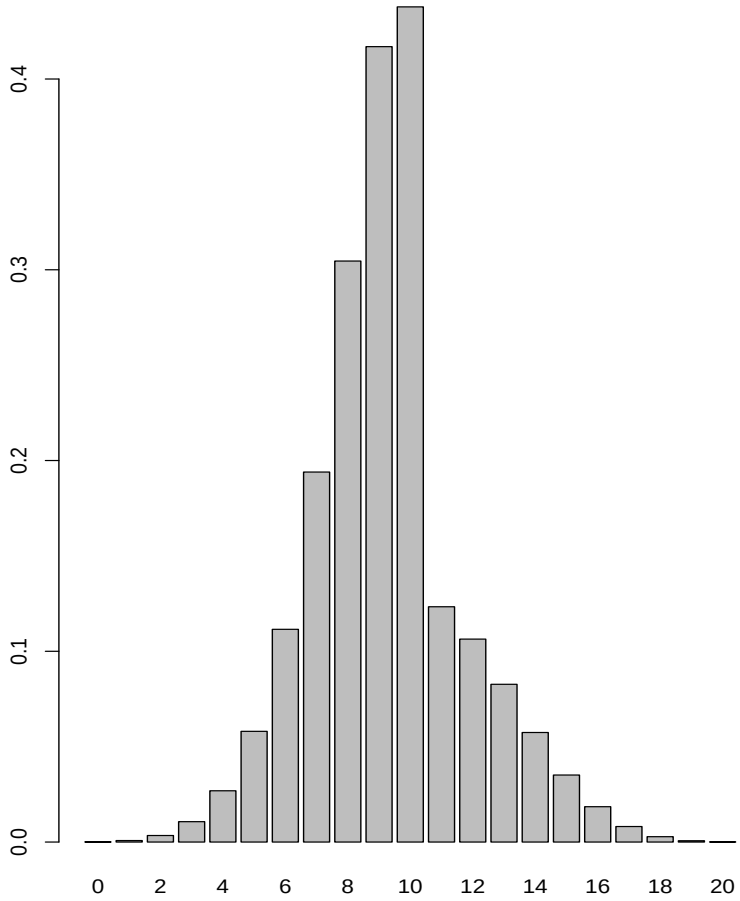
These lead us to combined predictive probability mass function using BMA of

$$\pi(x^*|x) = 0.36\binom{10}{x^*} \frac{\mathrm{B}(x^* + 12, 12 - x^*)}{\mathrm{B}(12, 2)} + 0.54\binom{20}{x^*} \frac{\mathrm{B}(x^* + 12, 32 - x^*)}{\mathrm{B}(12, 12)}$$

for $x^* \leq 10$ and

$$\pi(x^*|x) = 0.54\binom{20}{x^*} \frac{\mathrm{B}(x^* + 12, 32 - x^*)}{\mathrm{B}(12, 12)}$$

for $10 < x^* \leq 20$.



This entire method depends on our ability to derive the model evidence and, subsequently, the posterior probabilities for the models.

Example 11.4.2

$X|\theta \sim \mathsf{N}\left(\theta, \frac{1}{6}\right)$ with $\theta \sim \mathsf{Exp}(3)$, and we observe $x_1 = 1$ and $x_2 = 6$:

$$\pi(\theta|\underline{x}) \quad \propto \quad e^{-3\theta} e^{-3(1-\theta)^2} e^{-3(6-\theta)^2}.$$

We can find the posterior mode $\frac{13}{4}$ (by differentiation), but, to get the evidence, we need to integrate. We have:

$$\begin{aligned} \pi(\underline{x} = \{1, 6\}) &= \int_0^\infty \pi(x = 1|\theta)\pi(x = 6|\theta)\pi(\theta)\, d\theta \\ &\propto \int_0^\infty e^{-3\theta} e^{-3(1-\theta)^2} e^{-3(6-\theta)^2}\, d\theta. \end{aligned}$$

11.5 Predictive performance and cross validation

At the beginning of this chapter, the idea of using predictive performance to rate and select models was mentioned. In an ideal world, we would build our models and then use them to make separate predictions of some unseen data.

Example 11.5.1

Consider two models:

$$\begin{aligned} \mathcal{M}_1 &: \quad X|\mu \sim \mathsf{N}(\mu, 5^2), \quad \mu \sim \mathsf{N}(5, 1); \\ \mathcal{M}_2 &: \quad X|\mu \sim \mathsf{N}(\mu, 10^2), \quad \mu \sim \mathsf{N}(3, 1). \end{aligned}$$

We observe 50 x and get

$$\sum_{i=1}^{50} x_i = 191.34.$$

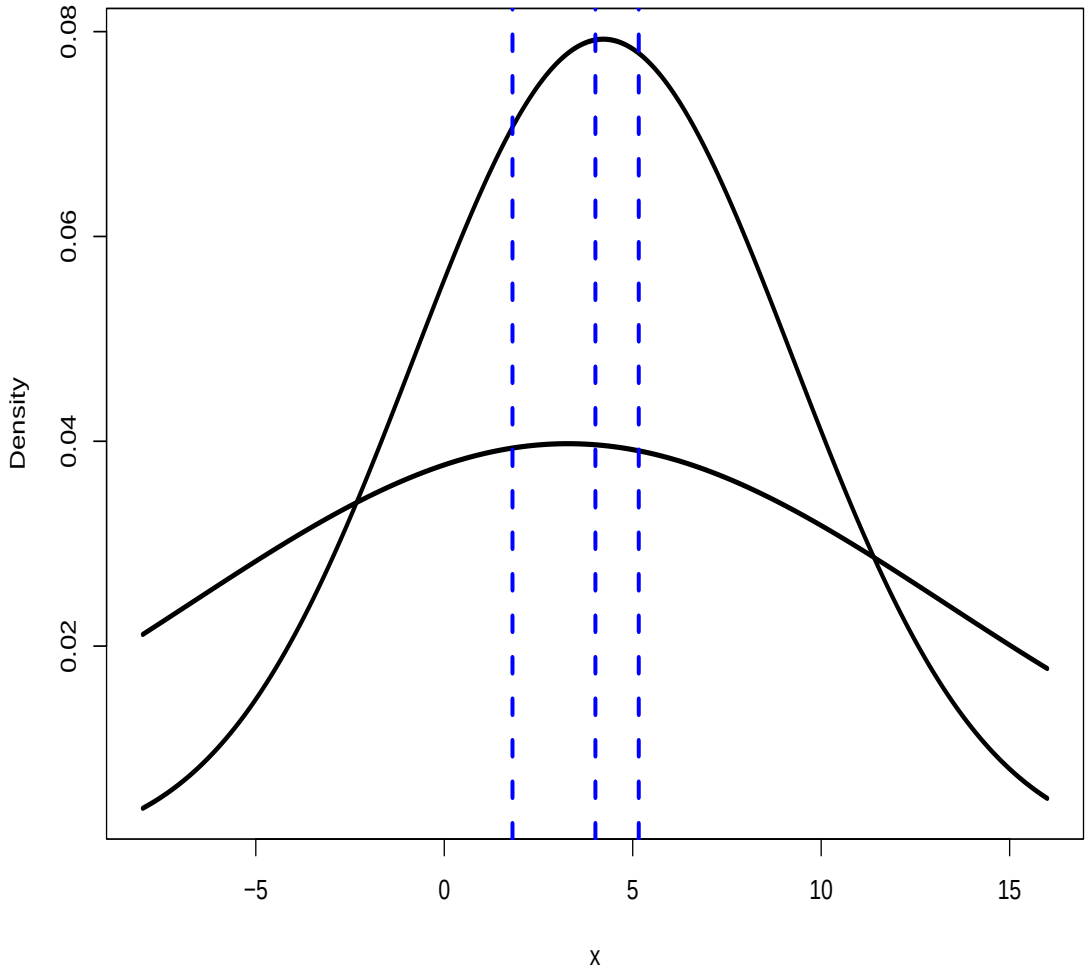
By conjugacy, we get the following posteriors for each model:

$$\begin{aligned} \mathcal{M}_1 &: \quad \mu|\underline{x} \sim \mathsf{N}(4.22, 1/3); \\ \mathcal{M}_2 &: \quad \mu|\underline{x} \sim \mathsf{N}(3.28, 2/3). \end{aligned}$$

We can also derive predictive distributions for both models:

$$\begin{aligned} \mathcal{M}_1 &: \quad X^*|\underline{x} \sim \mathsf{N}(4.22, 1/3 + 5^2); \\ \mathcal{M}_2 &: \quad X^*|\underline{x} \sim \mathsf{N}(3.28, 2/3 + 10^2). \end{aligned}$$

If we now observe 1.81, 4.01 and 5.16, which model is best?



What if we cannot get new data and we want to make a determination based on the data that we have already got? It is wrong to try to predict the data points that have been used to fit the model. *Cross validation* is popular in both statistics and machine learning for evaluating model performance (and for model fitting). Here, we will consider leave-one-out cross validation where the model is fitted on all the data apart from one observation and the model performance is a measure of how well the model using the reduced dataset replicates the single removed data point.

In the context of model selection, we may choose to use the log point-wise predictive density to evaluate model performance (logarithms are used to improve numeric stability and point-wise refers to the leave-one-out scheme):

$$\log\left[\pi(x_i|\underline{x}_{-i},\mathcal{M}_p)\right] = \log\left[\int \pi(\boldsymbol{\theta}|\underline{x}_{-i},\mathcal{M}_p)\pi(x_i|\boldsymbol{\theta},\mathcal{M}_p)d\boldsymbol{\theta}\right],$$

where $\boldsymbol{\theta}$ represents the unknown parameters and $\underline{x}_{-i}$ represents the full data set with the ith point removed. Clearly, we would like our performance metric to accommodate all of the individual data point predictions so we use the so-called expected log point-wise predictive density for model k, denoted here by elpd:

$$\widehat{\mathsf{elpd}}_p = \sum_{i=1}^{n} \log\left[\pi(x_i|\underline{x}_{-i},\mathcal{M}_p)\right].$$

We then convert this to a model weight, called a pseudo-Bayesian-model-averaging weight:

$$\widehat{w}_p = \frac{\exp\left(\widehat{\mathsf{elpd}}_p\right)}{\sum_{j=1}^{k}\exp\left(\widehat{\mathsf{elpd}}_j\right)}.$$

Therefore, the higher a model's elpd, the more influence it will have in the pseudo-Bayesian-model-averaging prediction. Note that these weights operate under the assumption that all models in $\underline{\mathcal{M}}$ are equally likely.

Example 11.5.2

Let's return to an earlier example. Again, consider two models:

$$\begin{aligned}\mathcal{M}_1 &: X|\theta \sim \mathsf{Bin}(10,\theta), \quad \theta \sim \mathsf{Be}(2,2);\\ \mathcal{M}_2 &: X|\theta \sim \mathsf{Bin}(20,\theta), \quad \theta \sim \mathsf{Be}(2,2).\end{aligned}$$

This time we observe $x_1 = 10$, $x_2 = 8$ and $x_3 = 10$. If we condition on two of those observations at a time, we have

$$\begin{aligned}\pi(x_i|\underline{x}_{-i},\mathcal{M}_1) &= \binom{10}{x_i}\frac{\mathrm{B}(x_i+\sum_{-i}x_j+2, 32-x_i-\sum_{-i}x_j)}{\mathrm{B}(2+\sum_{-i}x_j, 22-\sum_{-i}x_j)},\\ \pi(x_i|\underline{x}_{-i},\mathcal{M}_2) &= \binom{20}{x_i}\frac{\mathrm{B}(x_i+\sum_{-i}x_j+2, 62-x_i-\sum_{-i}x_j)}{\mathrm{B}(2+\sum_{-i}x_j, 42-\sum_{-i}x_j)}.\end{aligned}$$

We can then compute the elpd for each model:

Observation left out	$\log\pi(x_i\|\underline{x}_{-i},\mathcal{M}_1)$	$\log\pi(x_i\|\underline{x}_{-i},\mathcal{M}_2)$
x_1	-1.53	-1.98
x_2	-1.97	-2.18
x_3	-1.53	-1.98

We can then get the unnormalised pseudo-Bayesian-model-averaging weights of

$$\begin{aligned} \exp\left(\widehat{\mathsf{elpd}}_1\right) &= 0.0065, \\ \exp\left(\widehat{\mathsf{elpd}}_2\right) &= 0.0021, \end{aligned}$$

which can be combined to give normalised pseudo-Bayesian-model-averaging weights of 0.75 for $\mathcal{M}_1$ and 0.25 for $\mathcal{M}_2$.

This contradicts what was calculated in previous examples where the data seemed to favour $\mathcal{M}_2$. The reason is that, in full posterior calculations, you are rewarded for getting the prior in the right area. With the predictive-based weights, you are rewarded for stacking up posterior predictive probability near to the data (in other words, the likelihood takes over).

Just like model evidence, the elpd is difficult to calculate. However, we can find the psuedo-Bayesian-model-averaging weights by generating some extra quantities in Stan and utilising functions in the `loo` package. We add a variable called `log_lik` to the Stan code for each model under consideration:

```
generated quantities {
  // we need a log-likelihood calculation for each observation
  vector[N_obs] log_lik;
  // we have a normal likelihood in this example
  for (n in 1:N_obs)
    log_lik[n] = normal_lpdf(x[n] | mu, sigma);
}
```

Here, we are getting evaluations of the posterior predictive distribution of the observed values. These are $\log\left[\pi(x_i|\underline{x}, \mathcal{M}_p)\right]$ rather then the $\log\left[\pi(x_i|\underline{x}_{-i}, \mathcal{M}_p)\right]$ that are needed in the elpd calculation. The `loo` package works directly with the sampled log-likelihood values to approximate the expected log pointwise predictive density (by applying a correction):

```
library(loo)

# LOO for model 1
loo1 <- loo(fit)
# LOO for model 2
loo2 <- loo(fit_2)

# The weights
loo_model_weights(list(loo1, loo2),
                  method = "pseudobma",
                  BB = FALSE)
```

Example 11.5.3

Consider two models:

$$\begin{aligned} \mathcal{M}_1 &: X|\theta \sim \mathsf{N}(\theta, 0.16), \quad \theta \sim \mathsf{Exp}(3); \\ \mathcal{M}_2 &: X|\theta \sim \mathsf{N}(\theta, 0.17), \quad \theta \sim \mathsf{Exp}(3). \end{aligned}$$

We observe $\underline{x} = \{0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3\}$. It is clear already that both models are terrible. But, if we persist, we have the following Stan code:

```
data {
  int<lower=0> N;
  real x[N];
  real<lower=0> sigma2;
}
parameters {
  real<lower=0> theta;
}
model {
  // Prior
  theta ~ exponential(3);

  // Likelihood
  for (i in 1:N)
    x[i] ~ normal(theta, sqrt(sigma2));
}
generated quantities {
  vector[N] log_lik;
  for (i in 1:N)
    log_lik[i] = normal_lpdf(x[i] | theta, sqrt(sigma2));
}
```

The R code to then generate weights for each of the two models is then:

```
library(rstan)
library(loo)

# compile the model
model <- stan_model(file = "Model selection/expnorm.stan")
```

```
# generate a posterior sample for M1
fit1 <- sampling(model,
                 data = list(N = 20,
                             x = rep(0:3,5),
                             sigma2 = 0.16),
                 iter = 10000)
summary(fit1)$summary

# generate a posterior sample for M2
fit2 <- sampling(model,
                 data = list(N = 20,
                             x = rep(0:3,5),
                             sigma2 = 0.17),
                 iter = 10000)
summary(fit2)$summary

# calculate pseudo-BMA weights
loo1 <- loo(fit1)
loo2 <- loo(fit2)
loo_model_weights(list(loo1, loo2),
                  method = "pseudobma",
                  BB = FALSE)
```

This results in the following output:

```
Method: pseudo-BMA
------
       weight
model1 0.016
model2 0.984
```

which is not at all surprising and shows a key feature of modelling in the M-complete or M-open regimes: the weights will converge to one for the model that is closest to the true real-world model even if that model is wrong.