A prelude to the EM algorithm

Matthew Stephens
University of Chicago
January 9, 2026

See here for a PDF version of this vignette.

Pre-requisites

This document assumes basic familiarity with mixture models, likelihoods and Bayesian computations for the two-class problem.

Overview

The **expectation-maximization** algorithm, usually known as the EM algorithm, is a widely-used computational method for performing maximum likelihood method in certain models. Its attractions include that it is often simple to code and monotonic (every iteration increases the likelihood). It is also often quick to find a "reasonably good" solution, although if a very precise solution is required then it can be slow. In this document we describe the EM algorithm for a particular problem – maximum likelihood estimation of the mixture proportions – where it has a particularly simple and intuitive form. We do not give any formal derivation or explain why it works, which will require further study. The idea is to motivate you to perfom this further study.

The problem

We consider again the medical screening example from this vignette. This example involves a test for a disease that is based on measuring the concentration (X) of a protein in the blood. In normal individuals X has a Gamma distribution with mean 1 and shape 2 (so scale parameter is 0.5 as scale = mean/shape). In diseased individuals the protein becomes elevated, and X has a Gamma distribution with mean 2 and shape 2 (so scale =1).

Now consider the following question. Suppose we observe data $x_1, \dots, x_n$ on n individuals randomly sampled from a population. How can we estimate the proportion of individuals in the population that are diseased?

Formulating via maximum likelihood

A natural approach to parameter estimation to use maximum likelihood. The first step in this approach is to write down the likelihood. To do this note that the samples are independent from the following mixture model:

$$p(x|\pi) = \pi_1 \Gamma(x; 0.5, 2) + \pi_2 \Gamma(x; 1, 2)$$

where $\pi := (\pi_1, \pi_2)$, with π_1 the proportion of normal individuals, and π_2 the proportion of diseased individuals. Note that $\pi_1 + \pi_2 = 1$ as these are the only possibilities, so there is only really one parameter to be estimated here.

The likelihood for π is:

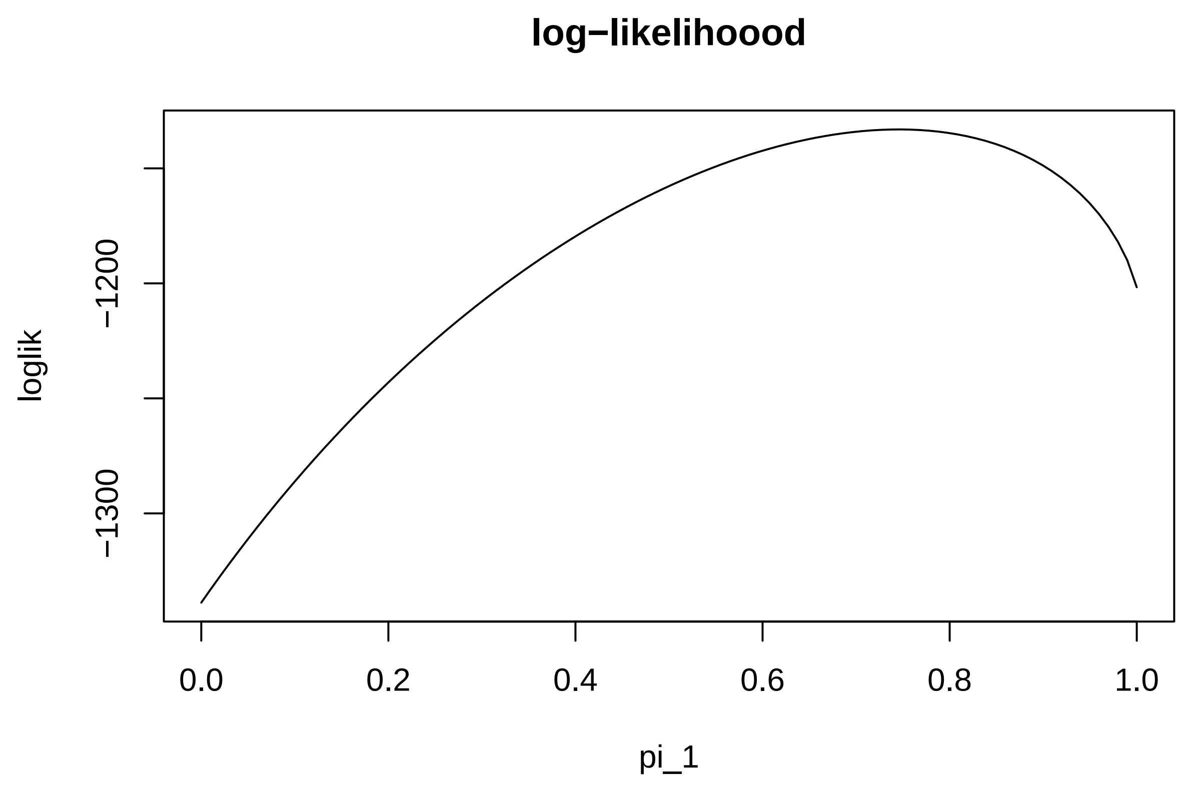
$$L(\pi) = \prod_j [\pi_1 \Gamma(x_j; 0.5, 2) + \pi_2 \Gamma(x_j; 1, 2)]$$

and so the log-likelihood is:

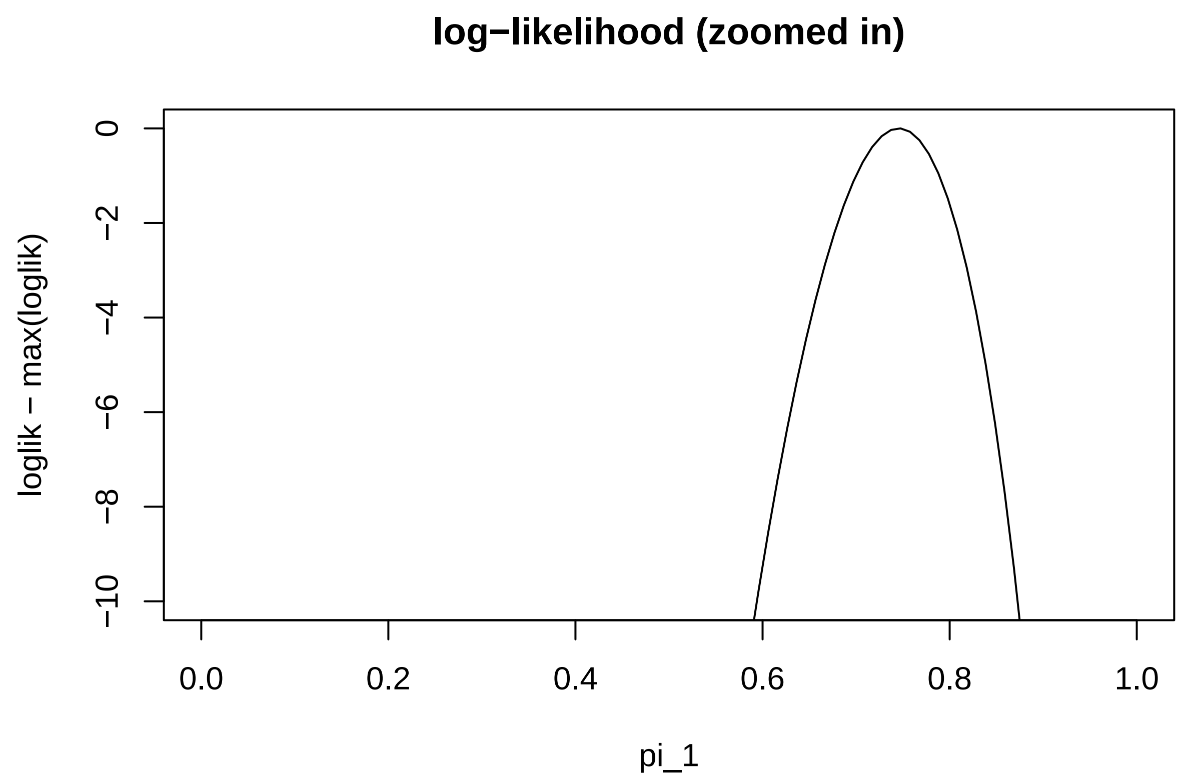
$$l(\pi) = \sum_j \log[\pi_1 \Gamma(x_j; 0.5, 2) + \pi_2 \Gamma(x_j; 1, 2)].$$

The following code simulates data from the model (with true value of $\pi_1 = 0.7$) and then plots the log-likelihood:

```
n = 1000 # number of samples
x = rep(0,n) # to store the samples
shape = c(2,2) # shapes of the two components
scale = c(0.5,1) # scales of the two components
for(i in 1:n){
  if(runif(1)<0.7)
    z=1 else z=2
  x[i] = rgamma(1,scale=scale[z],shape=shape[z])
}

mix_loglik = function(pi1,x){
  sum(log(pi1 * dgamma(x,scale=0.5,shape=2) + (1-pi1)*dgamma(x,scale=1,shape=2)))
}

pi1_vec = seq(0,1,length=100)
loglik = rep(0,100)
for(i in 1:length(pi1_vec)){
  loglik[i] = mix_loglik(pi1_vec[i],x)
}

plot(pi1_vec, loglik, type="l", main="log-likelihoood", xlab="pi_1")
```



```
plot(pi1_vec,loglik-max(loglik), ylim=c(-10,0),type="l", main="log-likelihood (zoomed in)" , xla
```



We can see from this plot that the maximum likelihood estiamte is near the true value (0.7). However, we cannot actually find the maximum easily analytically (e.g. try differentiating the log-likelihood and setting the derivative to zero). Therefore we need numerical methods. There are many numerical methods, and the EM algorithm is just one of many that could be used for this problem.

The EM algorithm

For this problem the EM algorithm has an intuitive form which we now describe. (Note this is not a derivation - it is just an intuitive heuristic argument).

For convenience introduce latent variables Z_i for each individual to indicate whether individual i is normal ($Z_i = 1$) or diseased ($Z_i = 2$). Of course the Z_i are unobserved.

Now note that *if we knew* π then we could easily compute the posterior probability that each individual is diseased given their test result $X_i = x_i$. Specifically

$$w_{ik} := \Pr(Z_i = k | X_i = x_i) = \pi_k L_{ik} / \sum_{k'=1}^{2} \pi_{k'} L_{ik'}$$

where $L_{ik} := p(x_i | Z_i = k)$. See Bayesian computations for the two-class problem for details.

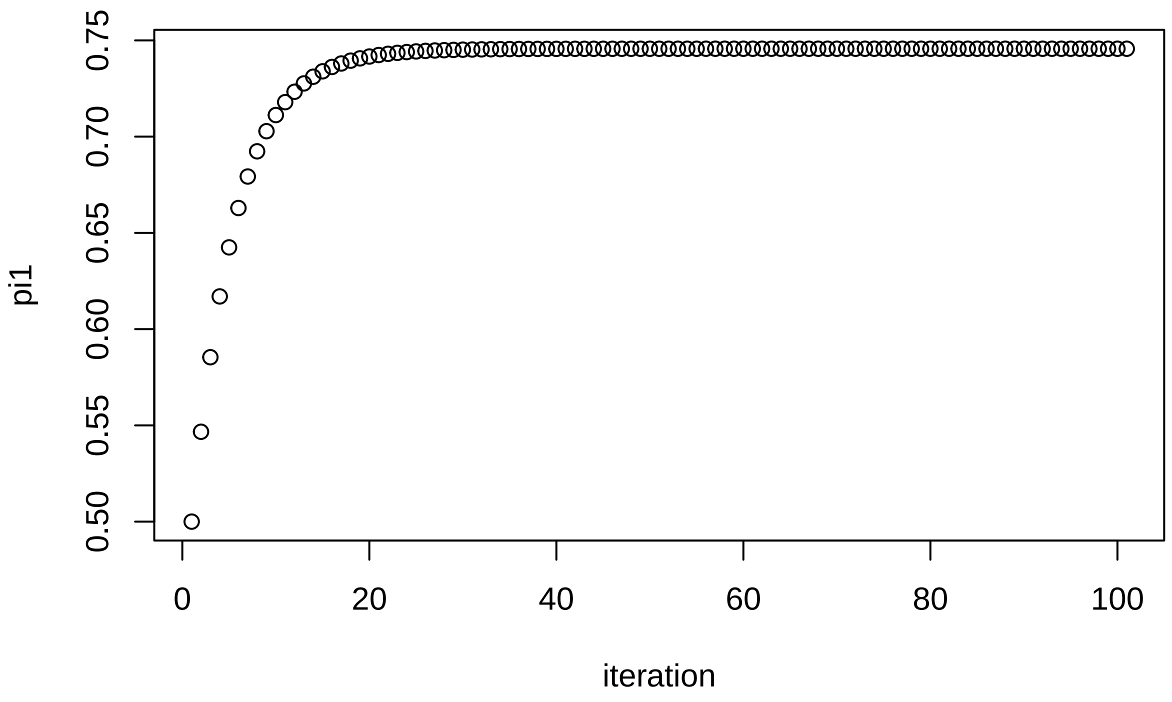
Furthermore, *if we knew these probabilities* it would be natural to estimate the proportion of individuals in class k by the average of the w_{ik}:

$$\pi_k \approx (1/n) \sum_{i=1}^{n} w_{ik}$$

.

So we have the following challenge: if we knew w we could estimate π, but we need to know π to compute w. This suggests, intuitively, an iterative procedure: repeatedly iterate between computing w and estimating π. Here is code implementing this idea:

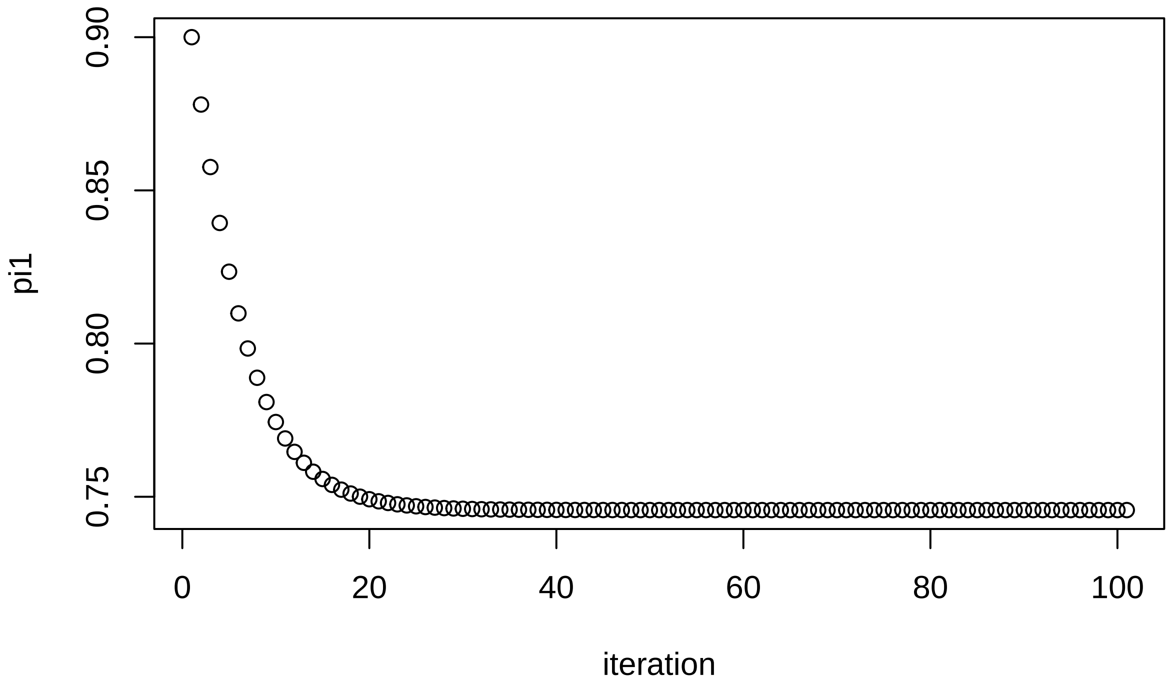
```
simple_em = function(x, pi1.init=0.5, niter = 100){
  n = length(x)
  pi1 = rep(0,niter+1)
  pi1[1] = pi1.init
  L = matrix(nrow=n,ncol=2)
  L[,1] = dgamma(x,scale=0.5,shape=2)
  L[,2] = dgamma(x,scale=1,shape=2)
  w = matrix(nrow=n,ncol=2)
  for(iter in 1:niter){
    w[,1] = pi1[iter]*L[,1]
    w[,2] = (1-pi1[iter])*L[,2]
    for(i in 1:n){ # normalize the rows of w to sum to 1
      w[i,] = w[i,]/(w[i,1]+w[i,2])
    }
    pi1[iter+1] = mean(w[,1])
  }
  return(pi1)
}

pi1.iter = simple_em(x)
plot(pi1.iter, xlab="iteration", ylab="pi1")
```



```
pi1.iter[101]
# [1] 0.7457
```

Here we try another starting point. We see it converges to the same value.

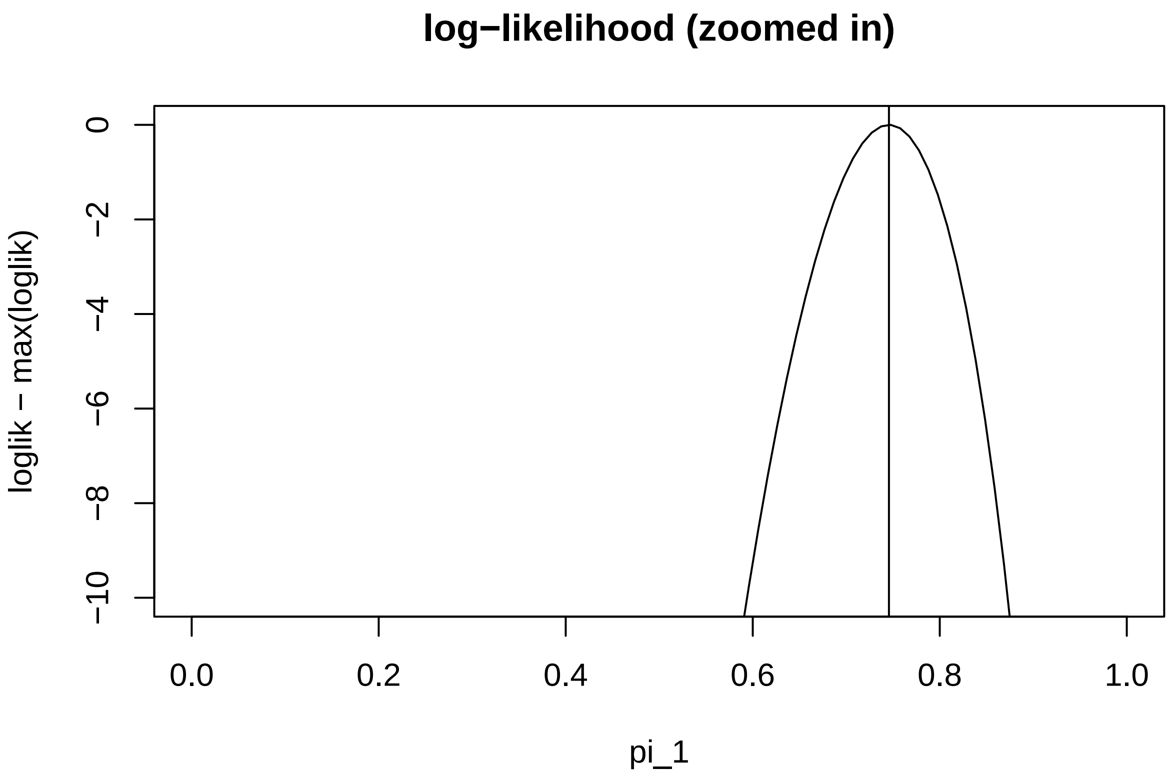
```
pi1.iter = simple_em(x,pi1.init=0.9)
plot(pi1.iter, xlab="iteration", ylab="pi1")
```



```
pi1.iter[101]
# [1] 0.7457
```

And check that this value it converges to is the maximum likelihood estimate:

```
plot(pi1_vec,loglik-max(loglik), ylim=c(-10,0),type="l", main="log-likelihood (zoomed in)" , xla
abline(v=pi1.iter[101])
```


log−likelihood (zoomed in)
loglik − max(loglik)
0
−2
−4
−6
−8
−10
0.0
0.2
0.4
0.6
0.8
1.0
pi_1