MSI Statistical Training Series

Expectation Maximization (EM)

2023-04-01

Table of contents

## Background

The MSI Statistical Training Series offers tutorials on a range of topics that are commonly applied in data analysis. The objective is to i) refresh readers on the derivation and application of methods from a first statistics course, ii) provide the underlying theory that goes beyond an introductory statistics course, iii) introduce readers to new methods that would not be covered in an introductory statistics course, and iv) provide practical, hands-on coding examples to enable users of statistical analysis software to gain proficiency in coding skills not just to run statistical procedures, but also to derive statistical procedures through the underlying theory.

Exploration of each topic should include the underlying theory, a demonstration using toy or fake data, and a real-world application in data that MSI has generated for a client deliverable.

## Introduction to Expectation Maximization

The Expectation Maximization (EM) algorithm is a popular approach to maximum likelihood estimation with missing data. The seminal reference is (Dempster, Laird, and Rubin 1977), which is among the most often cited article in the field of statistics.

The EM algorithm applies to problems of missing data, which has applications beyond the most direct use case of missing data imputation. For example, one may suspect that there is hidden structure in the data, and the missing data are labels identifying this structure (sex in a data set of heights; seasons in a data set of temperatures).

We may have data on an outcome Y which shows a bimodal distribution. We may suspect that this bimodal shape represents the missing data of sex - or in this case a missing variable, or label, that designates which sex the outcome observation belongs to. Or, we may have a set of covariate values for an outcome, and there are particular patterns in the values of these covariates that suggests the presence of a hidden, or latent, variable. This leads to the use of EM for the purposes of identifiying these missing or latent variables, with one approach being that of Latent Class Analysis (LCA).

The EM algorithm comprises two steps. First, we compute the expected value of the missing data, given the values of the observed data and ‘current’ parameter values.[[1]](#footnote-21) For a latent class model, we compute using the current parameter estimates. We then re-estimate the model probabilities and treating class membership as if it were observed and using the as weights. Somehow, each iteration of this process will improve the estimates until the algorithm converges to a solution.[[2]](#footnote-22)

## EM Demonstration Using LCA

We will demonstrate an application of the EM algorithm using Latent Class Analysis.

Consider a survey asking respondents about their attitude towards the limits of tolerance of anti-religious sentiment.[[3]](#footnote-24) The questions ask whether anti-religionists should be allowed to speak (y1), whether anti-religionists should be allowed to teach (y2), and whether anti-religious books ought to be allowed in public libraries (y3). For each item, 1 indicates agreement (allowed to speak, allowed to teach, allowed to keep books in library) and interpreted as high tolerance for anti-religious sentiment, and 2 indicates disagreement and interpreted as low tolerance for anti-religious sentiment.

d <- read\_csv("gss87 antirel.csv")  
flextable(d[1:5,])

| y1 | y2 | y3 | cum |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 1-1-1 |
| 1 | 1 | 1 | 1-1-1 |
| 1 | 2 | 2 | 1-2-2 |
| 1 | 1 | 1 | 1-1-1 |
| 1 | 1 | 1 | 1-1-1 |

#| output: asis  
  
tab\_stackfrq(d[,1:3],  
 digits=1)

|  |  |  |
| --- | --- | --- |
|  | 1 | 2 |
| y1 | 68.2 % | 31.8 % |
| y2 | 47.7 % | 52.3 % |
| y3 | 66.0 % | 34.0 % |

A slight majority of respondents agreed with items one and three, while item two was about evenly split. Perhaps there are two broad groups of respondents, those who are typically tolerant of anti-religious sentiment, and those who are not. We will now go through a series of steps in implementing the EM algorithm to identify these latent variables.

### 1. Tabulate the response patterns

We are interested in using the patterns of responses on the y1, y2, y3 variables to identify these two broad classes of respondents, so we created a new variable that captures their pattern across each respondent. We can then tabulate the frequencies of these patterns.

freqs <- data.frame(frq(d$cum)) %>%  
 .[1:8, c(2,4,5)] %>%  
 mutate(perc=round(raw.prc/100,3)) %>%  
 dplyr::select(-raw.prc)  
  
flextable(freqs)

| val | frq | perc |
| --- | --- | --- |
| 1-1-1 | 696 | 0.406 |
| 1-1-2 | 68 | 0.040 |
| 1-2-1 | 275 | 0.160 |
| 1-2-2 | 130 | 0.076 |
| 2-1-1 | 34 | 0.020 |
| 2-1-2 | 19 | 0.011 |
| 2-2-1 | 125 | 0.073 |
| 2-2-2 | 366 | 0.214 |

We suspect the pattern of responses of each of the three indicator variables arise due to the existence of two components, or latent variables, in the data. We will call this latent variable X, which can take on values of component 1 (X=1) or component 2 (X=2). We will refer to the observed indicator variables as .

We want to use the observed frequencies of the response patterns to infer the existence of the components X. To do that, we posit a data-generating process, a model of described in the table above. There are two key assumptions of this model:

1. The observed frequencies of the pattern of responses is the result of a mixture of two separate components, each with their own distribution. If this were true, then the observed frequency could be expressed as

* Note that the probabilities of the indicator variables for each component[[4]](#footnote-27), as well as the share of each component in the sample[[5]](#footnote-28) , are not observed - they are the parameters of the model.

1. Within each component, responses are independent (*local independence*). Within each component, knowing offers no information about . If this is true, then the probability of observing a given pattern of responses is simply the product of their probabilities.

* Again note that the notation above refers not to observed probabilities, but the parameter probabilities of the model.

The above model assumptions identify two sets of parameters that will generate our observed information. The first set of parameters is the proportion of the sample belonging to each component. This is from model assumption (1) above. Second, we need the proportion of each indicator variable (the y1, y2, y3) within each assigned component. This is . Knowing these parameter values allows us to generate the joint probability of the pattern of responses in the indicator variables described in model assumption (2).

### 2. Set initial values of model parameters

We will start by arbitrarily setting initial values for these parameters, and then use these values to estimate the probabilities that each observation belongs to each component. Let’s start off by saying that components 1 and 2 are evenly apportioned in the sample, and that the probabilities of an observation being 1 for each of the indicator variables is 60 percent.

p\_x1 <- .5 # share of component 1, iteration 1  
p\_x2 <- .5  
  
y1\_x1 <- .6 # probability of y1 for component 1, iteration 1  
y2\_x1 <- .6  
y3\_x1 <- .6  
  
y1\_x2 <- .4 # probability of y1 for component 2, iteration 1  
y2\_x2 <- .4  
y3\_x2 <- .4  
  
params1 <- data.frame(estimate="current",  
 probs=c("P(X)","P(y1=1|X)","P(y2=1|X)", "P(y3=1|X)"),  
 iteration=1,  
 X1=c(.5,.6,.6,.6),  
 X2=c(.5,.4,.4,.4))  
  
flextable(params1) %>%  
 set\_header\_labels(X1="X=1",  
 X2="X=2")

| estimate | probs | iteration | X=1 | X=2 |
| --- | --- | --- | --- | --- |
| current | P(X) | 1 | 0.5 | 0.5 |
| current | P(y1=1|X) | 1 | 0.6 | 0.4 |
| current | P(y2=1|X) | 1 | 0.6 | 0.4 |
| current | P(y3=1|X) | 1 | 0.6 | 0.4 |

### 3. Estimate probabilities of response patterns

We now use the initial indicator parameters to estimate a probability of observing a given response pattern , where the are the observed patterns of responses in the sample, and the Xs are the current parameter estimates of the probability of each indicator variable for each component. If you’re trying to follow along, here are the observed indicator response pattern frequencies and initial parameter estimates.

freqs %>%  
 kable("html", align = 'clc', caption = 'Observed pattern of responses') %>%  
 kable\_styling(full\_width = F, position = "float\_left")  
  
params1 %>%  
 kable("html", align = 'clc', caption = 'Initial parameter values') %>%  
 kable\_styling(full\_width = F, position = "float\_right")

Observed pattern of responses

| val | frq | perc |
| --- | --- | --- |
| 1-1-1 | 696 | 0.406 |
| 1-1-2 | 68 | 0.040 |
| 1-2-1 | 275 | 0.160 |
| 1-2-2 | 130 | 0.076 |
| 2-1-1 | 34 | 0.020 |
| 2-1-2 | 19 | 0.011 |
| 2-2-1 | 125 | 0.073 |
| 2-2-2 | 366 | 0.214 |

Initial parameter values

| estimate | probs | iteration | X1 | X2 |
| --- | --- | --- | --- | --- |
| current | P(X) | 1 | 0.5 | 0.5 |
| current | P(y1=1|X) | 1 | 0.6 | 0.4 |
| current | P(y2=1|X) | 1 | 0.6 | 0.4 |
| current | P(y3=1|X) | 1 | 0.6 | 0.4 |

The most common response pattern is 1-1-1, which occurred in 41 percent of respondents.

Given our initial model parameters that each indicator value has a 60 percent probability of belonging to component 1 and a 40 percent probability of belonging to component 2, then we would expect to see the pattern (1-1-1) to be = 21.6% of the time in component 1, and = 6.4% of the time in component 2. These estimates represent .

Probability of response patterns, if they belonged to component 1.

x1\_111 <- y1\_x1\*y2\_x1\*y3\_x1   
# probability of 1-1-1 belonging to component 1  
x1\_112 <- y1\_x1\*y1\_x1\*(1-y3\_x1)  
# probability of 1-1-2 belonging to component 1  
x1\_121 <- y1\_x1\*(1-y2\_x1)\*y3\_x1  
# probability of 1-2-1 belonging to component 1  
x1\_122 <- y1\_x1\*(1-y2\_x1)\*(1-y3\_x1)  
x1\_211 <- (1-y1\_x1)\*y2\_x1\*y3\_x1  
x1\_212 <- (1-y1\_x1)\*y2\_x1\*(1-y3\_x1)  
x1\_221 <- (1-y1\_x1)\*(1-y2\_x1)\*(y3\_x1)  
x1\_222 <- (1-y1\_x1)\*(1-y2\_x1)\*(1-y3\_x1)  
  
jointy\_x1 <- c(x1\_111, x1\_112, x1\_121, x1\_122, x1\_211, x1\_212, x1\_221, x1\_222)

Probability of response patterns, if they belonged to component 2.

x2\_111 <- y1\_x2\*y2\_x2\*y3\_x2   
# probability of 1-1-1 belonging to component 1  
x2\_112 <- y1\_x2\*y1\_x2\*(1-y3\_x2)  
# probability of 1-1-2 belonging to component 1  
x2\_121 <- y1\_x2\*(1-y2\_x2)\*y3\_x2  
# probability of 1-2-1 belonging to component 1  
x2\_122 <- y1\_x2\*(1-y2\_x2)\*(1-y3\_x2)  
x2\_211 <- (1-y1\_x2)\*y2\_x2\*y3\_x2  
x2\_212 <- (1-y1\_x2)\*y2\_x2\*(1-y3\_x2)  
x2\_221 <- (1-y1\_x2)\*(1-y2\_x2)\*(y3\_x2)  
x2\_222 <- (1-y1\_x2)\*(1-y2\_x2)\*(1-y3\_x2)  
  
jointy\_x2 <- c(x2\_111, x2\_112, x2\_121, x2\_122, x2\_211, x2\_212, x2\_221, x2\_222)  
  
freqs1 <- freqs %>%  
 mutate(iteration=1,  
 joint\_x1=jointy\_x1,  
 joint\_x2=jointy\_x2)  
  
flextable(freqs1)

| val | frq | perc | iteration | joint\_x1 | joint\_x2 |
| --- | --- | --- | --- | --- | --- |
| 1-1-1 | 696 | 0.406 | 1 | 0.216 | 0.064 |
| 1-1-2 | 68 | 0.040 | 1 | 0.144 | 0.096 |
| 1-2-1 | 275 | 0.160 | 1 | 0.144 | 0.096 |
| 1-2-2 | 130 | 0.076 | 1 | 0.096 | 0.144 |
| 2-1-1 | 34 | 0.020 | 1 | 0.144 | 0.096 |
| 2-1-2 | 19 | 0.011 | 1 | 0.096 | 0.144 |
| 2-2-1 | 125 | 0.073 | 1 | 0.096 | 0.144 |
| 2-2-2 | 366 | 0.214 | 1 | 0.064 | 0.216 |

For each indicator selection pattern, we multiply the posited model probabilities together to get the joint probability of observing that pattern, if the posited parameter values were true. We do that for each posited component.

We then invoke Assumption 1[[6]](#footnote-32) to estimate the observed frequency of the response pattern as the sum of two mixtures.

freqs1 <- freqs1 %>%  
 mutate(joint\_y=joint\_x1\*p\_x1 + joint\_x2\*p\_x2)  
  
flextable(freqs1)

| val | frq | perc | iteration | joint\_x1 | joint\_x2 | joint\_y |
| --- | --- | --- | --- | --- | --- | --- |
| 1-1-1 | 696 | 0.406 | 1 | 0.216 | 0.064 | 0.14 |
| 1-1-2 | 68 | 0.040 | 1 | 0.144 | 0.096 | 0.12 |
| 1-2-1 | 275 | 0.160 | 1 | 0.144 | 0.096 | 0.12 |
| 1-2-2 | 130 | 0.076 | 1 | 0.096 | 0.144 | 0.12 |
| 2-1-1 | 34 | 0.020 | 1 | 0.144 | 0.096 | 0.12 |
| 2-1-2 | 19 | 0.011 | 1 | 0.096 | 0.144 | 0.12 |
| 2-2-1 | 125 | 0.073 | 1 | 0.096 | 0.144 | 0.12 |
| 2-2-2 | 366 | 0.214 | 1 | 0.064 | 0.216 | 0.14 |

If the parameter estimates are true, then the generated joint probability of the indicator variables (joint\_y) should match the observed joint probabilities of the indicator variables (perc). After this first iteration, they are not close.

### 4. Estimate the component probabilities within each component

The steps we have taken so far have given us the predicted joint probability of observing a pattern of indicator variables, if the posited parameter values are true. This may be expressed as . Recall from the introduction of the EM algorithm is that what we are seeking to estimate, the E step, is . This would be an assignation of a probability of belonging to each component *c*, given the observed response pattern of the indicator variables.

We can apply Bayes Theorem to recover our desired quantity.

| estimate | probs | iteration | X1 | X2 |
| --- | --- | --- | --- | --- |
| current | P(X) | 1 | 0.5 | 0.5 |
| current | P(y1=1|X) | 1 | 0.6 | 0.4 |
| current | P(y2=1|X) | 1 | 0.6 | 0.4 |
| current | P(y3=1|X) | 1 | 0.6 | 0.4 |

freqs1 <- freqs1 %>%  
 mutate(x1\_y=(p\_x1\*joint\_x1)/joint\_y,  
 x2\_y=(p\_x2\*joint\_x2)/joint\_y)  
  
flextable(freqs1)

| val | frq | perc | iteration | joint\_x1 | joint\_x2 | joint\_y | x1\_y | x2\_y |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1-1-1 | 696 | 0.406 | 1 | 0.216 | 0.064 | 0.14 | 0.771 | 0.229 |
| 1-1-2 | 68 | 0.040 | 1 | 0.144 | 0.096 | 0.12 | 0.600 | 0.400 |
| 1-2-1 | 275 | 0.160 | 1 | 0.144 | 0.096 | 0.12 | 0.600 | 0.400 |
| 1-2-2 | 130 | 0.076 | 1 | 0.096 | 0.144 | 0.12 | 0.400 | 0.600 |
| 2-1-1 | 34 | 0.020 | 1 | 0.144 | 0.096 | 0.12 | 0.600 | 0.400 |
| 2-1-2 | 19 | 0.011 | 1 | 0.096 | 0.144 | 0.12 | 0.400 | 0.600 |
| 2-2-1 | 125 | 0.073 | 1 | 0.096 | 0.144 | 0.12 | 0.400 | 0.600 |
| 2-2-2 | 366 | 0.214 | 1 | 0.064 | 0.216 | 0.14 | 0.229 | 0.771 |

### 5. Estimate frequency counts of estimated components

Using the values of our initial parameter estimates, we have generated estimates of the probability of the respondent belonging to a given component, given their response pattern on the indicator variables. We can now use these estimated probabilities to estimate the number of respondents we expect to belong to each component.

freqs1 <- freqs1 %>%  
 mutate(frq\_x1=x1\_y\*frq,  
 frq\_x2=x2\_y\*frq)  
  
flextable(freqs1)

| val | frq | perc | iteration | joint\_x1 | joint\_x2 | joint\_y | x1\_y | x2\_y | frq\_x1 | frq\_x2 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1-1-1 | 696 | 0.406 | 1 | 0.216 | 0.064 | 0.14 | 0.771 | 0.229 | 536.9 | 159.1 |
| 1-1-2 | 68 | 0.040 | 1 | 0.144 | 0.096 | 0.12 | 0.600 | 0.400 | 40.8 | 27.2 |
| 1-2-1 | 275 | 0.160 | 1 | 0.144 | 0.096 | 0.12 | 0.600 | 0.400 | 165.0 | 110.0 |
| 1-2-2 | 130 | 0.076 | 1 | 0.096 | 0.144 | 0.12 | 0.400 | 0.600 | 52.0 | 78.0 |
| 2-1-1 | 34 | 0.020 | 1 | 0.144 | 0.096 | 0.12 | 0.600 | 0.400 | 20.4 | 13.6 |
| 2-1-2 | 19 | 0.011 | 1 | 0.096 | 0.144 | 0.12 | 0.400 | 0.600 | 7.6 | 11.4 |
| 2-2-1 | 125 | 0.073 | 1 | 0.096 | 0.144 | 0.12 | 0.400 | 0.600 | 50.0 | 75.0 |
| 2-2-2 | 366 | 0.214 | 1 | 0.064 | 0.216 | 0.14 | 0.229 | 0.771 | 83.7 | 282.3 |

Now that we have the predicted frequencies of cases exhibiting a particular response pattern for each component, we will use these probabilities to predict overall frequency counts of the parameters.

x1\_freqs <- sum(freqs1$frq\_x1)  
#x1\_freqs2  
  
x2\_freqs <- sum(freqs1$frq\_x2)  
#x2\_freqs2  
  
f\_y1\_c1 <- sum(freqs1[1:4,10])  
#f1\_y1\_c1  
  
f\_y1\_c2 <- sum(freqs1[1:4,11])  
#f1\_y1\_c2  
  
f\_y2\_c1 <- sum(freqs1[c(1:2, 5:6),10])  
#f1\_y2\_c1  
  
f\_y2\_c2 <- sum(freqs1[c(1:2, 5:6), 11])  
#f1\_y2\_c2  
  
f\_y3\_c1 <- sum(freqs1[c(1,3,5,7), 10])  
#f1\_y3\_c1  
  
f\_y3\_c2 <- sum(freqs1[c(1,3,5,7), 11])  
#f1\_y3\_c2  
  
newfreqs <- data.frame(estimate="new",  
 probs=params1$probs,  
 iteration=1) %>%  
 mutate(X1=c(x1\_freqs,  
 f\_y1\_c1,  
 f\_y2\_c1,  
 f\_y3\_c1),  
 X2=c(x2\_freqs,  
 f\_y1\_c2,  
 f\_y2\_c2,  
 f\_y3\_c2))  
   
newfreqs\_flex <- newfreqs %>%  
flextable()   
  
newfreqs\_flex

| estimate | probs | iteration | X1 | X2 |
| --- | --- | --- | --- | --- |
| new | P(X) | 1 | 956 | 757 |
| new | P(y1=1|X) | 1 | 795 | 374 |
| new | P(y2=1|X) | 1 | 606 | 211 |
| new | P(y3=1|X) | 1 | 772 | 358 |

### 6. Estimate new model parameters

Now we use these predicted frequency counts to generate new parameter probabilities

pnew\_x1 <- x1\_freqs / (x1\_freqs + x2\_freqs)  
#pnew\_x1  
  
pnew\_x2 <- x2\_freqs / (x1\_freqs + x2\_freqs)  
#pnew\_x2  
  
# y's for component 1  
  
y1\_x1 <- f\_y1\_c1 / x1\_freqs  
#y1\_x1  
  
y2\_x1 <- f\_y2\_c1 / x1\_freqs  
#y2\_x1  
  
y3\_x1 <- f\_y3\_c1 / x1\_freqs  
#y3\_x1  
  
# y's for component 2  
  
y1\_x2 <- f\_y1\_c2 / x2\_freqs  
#y1\_x2  
  
y2\_x2 <- f\_y2\_c2 / x2\_freqs  
#y2\_x2  
  
y3\_x2 <- f\_y3\_c2 / x2\_freqs  
#y3\_x2

Arrange new model parameters into a table.

new <- data.frame(estimate="new",  
 probs=params1$probs,  
 iteration=1) %>%  
 mutate(X1=c(pnew\_x1,  
 y1\_x1,  
 y2\_x1,  
 y3\_x1),  
 X2=c(pnew\_x2,  
 y1\_x2,  
 y2\_x2,  
 y3\_x2))  
   
#flextable(new)  
  
params1 <- bind\_rows(params1, new)  
  
flextable(params1)

| estimate | probs | iteration | X1 | X2 |
| --- | --- | --- | --- | --- |
| current | P(X) | 1 | 0.500 | 0.500 |
| current | P(y1=1|X) | 1 | 0.600 | 0.400 |
| current | P(y2=1|X) | 1 | 0.600 | 0.400 |
| current | P(y3=1|X) | 1 | 0.600 | 0.400 |
| new | P(X) | 1 | 0.558 | 0.442 |
| new | P(y1=1|X) | 1 | 0.831 | 0.495 |
| new | P(y2=1|X) | 1 | 0.633 | 0.279 |
| new | P(y3=1|X) | 1 | 0.808 | 0.473 |

Comparing initial to new estimates of model parameters, we see that the estimate of the share of component 1 in the sample increases to 56 percent, while the estimate of the share of component 2 in the sample decreases to 44 percent. Probability of belonging to component 1 increased for all indicator variables, and decreased for all indicator variables belonging to comnponent 2.

What are we to make of this? There exists a proof that these new estimates better describe the observed data. We know this with certainty if we had known the hidden labels beforehand but blinded them in this analysis. But without the hidden knowledge, how do we know?

We can take the log likelihood of our response patterns at each iteration to see whether we are arriving at some sort of ground-truth. It can be shown that the log likelihood is guaranteed to increase after each iteration of the algorithm.

freqs1 <- freqs1 %>%  
 mutate(ll=frq\*log(joint\_y))  
  
flextable(freqs1)

| val | frq | perc | iteration | joint\_x1 | joint\_x2 | joint\_y | x1\_y | x2\_y | frq\_x1 | frq\_x2 | ll |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1-1-1 | 696 | 0.406 | 1 | 0.216 | 0.064 | 0.14 | 0.771 | 0.229 | 536.9 | 159.1 | -1,368.4 |
| 1-1-2 | 68 | 0.040 | 1 | 0.144 | 0.096 | 0.12 | 0.600 | 0.400 | 40.8 | 27.2 | -144.2 |
| 1-2-1 | 275 | 0.160 | 1 | 0.144 | 0.096 | 0.12 | 0.600 | 0.400 | 165.0 | 110.0 | -583.1 |
| 1-2-2 | 130 | 0.076 | 1 | 0.096 | 0.144 | 0.12 | 0.400 | 0.600 | 52.0 | 78.0 | -275.6 |
| 2-1-1 | 34 | 0.020 | 1 | 0.144 | 0.096 | 0.12 | 0.600 | 0.400 | 20.4 | 13.6 | -72.1 |
| 2-1-2 | 19 | 0.011 | 1 | 0.096 | 0.144 | 0.12 | 0.400 | 0.600 | 7.6 | 11.4 | -40.3 |
| 2-2-1 | 125 | 0.073 | 1 | 0.096 | 0.144 | 0.12 | 0.400 | 0.600 | 50.0 | 75.0 | -265.0 |
| 2-2-2 | 366 | 0.214 | 1 | 0.064 | 0.216 | 0.14 | 0.229 | 0.771 | 83.7 | 282.3 | -719.6 |

ll\_1 <- sum(freqs1$ll)  
ll\_1

[1] -3468

## Iterate the model

Let’s now go through the iteration again, using the new model parameter estimates.

params2 <- new %>%  
 mutate(estimate="current",  
 iteration=2)  
  
flextable(params2)

| estimate | probs | iteration | X1 | X2 |
| --- | --- | --- | --- | --- |
| current | P(X) | 2 | 0.558 | 0.442 |
| current | P(y1=1|X) | 2 | 0.831 | 0.495 |
| current | P(y2=1|X) | 2 | 0.633 | 0.279 |
| current | P(y3=1|X) | 2 | 0.808 | 0.473 |

p\_x1 <- params2[1,4] # share of component 1  
p\_x2 <- params2[1,5]  
  
y1\_x1 <- params2[2,4] # probability of y1 for component 1  
y2\_x1 <- params2[3,4]  
y3\_x1 <- params2[4,4]  
  
y1\_x2 <- params2[2,5] # probability of y1 for component 2  
y2\_x2 <- params2[3,5]  
y3\_x2 <- params2[4,5]

Probability of response patterns, if they belonged to component 1.

x1\_111 <- y1\_x1\*y2\_x1\*y3\_x1   
# probability of 1-1-1 belonging to component 1  
x1\_112 <- y1\_x1\*y1\_x1\*(1-y3\_x1)  
# probability of 1-1-2 belonging to component 1  
x1\_121 <- y1\_x1\*(1-y2\_x1)\*y3\_x1  
# probability of 1-2-1 belonging to component 1  
x1\_122 <- y1\_x1\*(1-y2\_x1)\*(1-y3\_x1)  
x1\_211 <- (1-y1\_x1)\*y2\_x1\*y3\_x1  
x1\_212 <- (1-y1\_x1)\*y2\_x1\*(1-y3\_x1)  
x1\_221 <- (1-y1\_x1)\*(1-y2\_x1)\*(y3\_x1)  
x1\_222 <- (1-y1\_x1)\*(1-y2\_x1)\*(1-y3\_x1)  
  
jointy\_x1 <- c(x1\_111, x1\_112, x1\_121, x1\_122, x1\_211, x1\_212, x1\_221, x1\_222)

Probability of response patterns, if they belonged to component 2.

x2\_111 <- y1\_x2\*y2\_x2\*y3\_x2   
# probability of 1-1-1 belonging to component 1  
x2\_112 <- y1\_x2\*y1\_x2\*(1-y3\_x2)  
# probability of 1-1-2 belonging to component 1  
x2\_121 <- y1\_x2\*(1-y2\_x2)\*y3\_x2  
# probability of 1-2-1 belonging to component 1  
x2\_122 <- y1\_x2\*(1-y2\_x2)\*(1-y3\_x2)  
x2\_211 <- (1-y1\_x2)\*y2\_x2\*y3\_x2  
x2\_212 <- (1-y1\_x2)\*y2\_x2\*(1-y3\_x2)  
x2\_221 <- (1-y1\_x2)\*(1-y2\_x2)\*(y3\_x2)  
x2\_222 <- (1-y1\_x2)\*(1-y2\_x2)\*(1-y3\_x2)  
  
jointy\_x2 <- c(x2\_111, x2\_112, x2\_121, x2\_122, x2\_211, x2\_212, x2\_221, x2\_222)  
  
freqs2 <- freqs %>%  
 mutate(iteration=2,  
 joint\_x1=jointy\_x1,  
 joint\_x2=jointy\_x2,  
 joint\_y=joint\_x1\*p\_x1 + joint\_x2\*p\_x2,  
 x1\_y=(p\_x1\*joint\_x1)/joint\_y,  
 x2\_y=(p\_x2\*joint\_x2)/joint\_y,  
 frq\_x1=x1\_y\*frq,  
 frq\_x2=x2\_y\*frq)  
  
flextable(freqs2)

| val | frq | perc | iteration | joint\_x1 | joint\_x2 | joint\_y | x1\_y | x2\_y | frq\_x1 | frq\_x2 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1-1-1 | 696 | 0.406 | 2 | 0.4250 | 0.0653 | 0.2661 | 0.8916 | 0.108 | 620.56 | 75.4 |
| 1-1-2 | 68 | 0.040 | 2 | 0.1329 | 0.1290 | 0.1312 | 0.5656 | 0.434 | 38.46 | 29.5 |
| 1-2-1 | 275 | 0.160 | 2 | 0.2460 | 0.1685 | 0.2118 | 0.6485 | 0.351 | 178.34 | 96.7 |
| 1-2-2 | 130 | 0.076 | 2 | 0.0586 | 0.1880 | 0.1158 | 0.2828 | 0.717 | 36.76 | 93.2 |
| 2-1-1 | 34 | 0.020 | 2 | 0.0865 | 0.0667 | 0.0777 | 0.6209 | 0.379 | 21.11 | 12.9 |
| 2-1-2 | 19 | 0.011 | 2 | 0.0206 | 0.0744 | 0.0444 | 0.2593 | 0.741 | 4.93 | 14.1 |
| 2-2-1 | 125 | 0.073 | 2 | 0.0500 | 0.1722 | 0.1040 | 0.2687 | 0.731 | 33.59 | 91.4 |
| 2-2-2 | 366 | 0.214 | 2 | 0.0119 | 0.1920 | 0.0915 | 0.0728 | 0.927 | 26.64 | 339.4 |

x1\_freqs <- sum(freqs2$frq\_x1)  
#x1\_freqs2  
  
x2\_freqs <- sum(freqs2$frq\_x2)  
#x2\_freqs2  
  
f\_y1\_c1 <- sum(freqs2[1:4,10])  
#f1\_y1\_c1  
  
f\_y1\_c2 <- sum(freqs2[1:4,11])  
#f1\_y1\_c2  
  
f\_y2\_c1 <- sum(freqs2[c(1:2, 5:6),10])  
#f1\_y2\_c1  
  
f\_y2\_c2 <- sum(freqs2[c(1:2, 5:6), 11])  
#f1\_y2\_c2  
  
f\_y3\_c1 <- sum(freqs2[c(1,3,5,7), 10])  
#f1\_y3\_c1  
  
f\_y3\_c2 <- sum(freqs2[c(1,3,5,7), 11])  
#f1\_y3\_c2  
  
newfreqs <- data.frame(estimate="new",  
 probs=params2$probs,  
 iteration=2) %>%  
 mutate(X1=c(x1\_freqs,  
 f\_y1\_c1,  
 f\_y2\_c1,  
 f\_y3\_c1),  
 X2=c(x2\_freqs,  
 f\_y1\_c2,  
 f\_y2\_c2,  
 f\_y3\_c2))  
   
newfreqs\_flex <- newfreqs %>%  
flextable()   
  
newfreqs\_flex

| estimate | probs | iteration | X1 | X2 |
| --- | --- | --- | --- | --- |
| new | P(X) | 2 | 960 | 753 |
| new | P(y1=1|X) | 2 | 874 | 295 |
| new | P(y2=1|X) | 2 | 685 | 132 |
| new | P(y3=1|X) | 2 | 854 | 276 |

pnew\_x1 <- x1\_freqs / (x1\_freqs + x2\_freqs)  
#pnew\_x1  
  
pnew\_x2 <- x2\_freqs / (x1\_freqs + x2\_freqs)  
#pnew\_x2  
  
# y's for component 1  
  
y1\_x1 <- f\_y1\_c1 / x1\_freqs  
#y1\_x1  
  
y2\_x1 <- f\_y2\_c1 / x1\_freqs  
#y2\_x1  
  
y3\_x1 <- f\_y3\_c1 / x1\_freqs  
#y3\_x1  
  
# y's for component 2  
  
y1\_x2 <- f\_y1\_c2 / x2\_freqs  
#y1\_x2  
  
y2\_x2 <- f\_y2\_c2 / x2\_freqs  
#y2\_x2  
  
y3\_x2 <- f\_y3\_c2 / x2\_freqs  
#y3\_x2

Arrange new model parameters into a table.

new <- data.frame(estimate="new",  
 probs=params1$probs[1:4],  
 iteration=2) %>%  
 mutate(X1=c(pnew\_x1,  
 y1\_x1,  
 y2\_x1,  
 y3\_x1),  
 X2=c(pnew\_x2,  
 y1\_x2,  
 y2\_x2,  
 y3\_x2))  
   
#flextable(new)  
  
params2 <- bind\_rows(params2, new)  
  
flextable(params2)

| estimate | probs | iteration | X1 | X2 |
| --- | --- | --- | --- | --- |
| current | P(X) | 2 | 0.558 | 0.442 |
| current | P(y1=1|X) | 2 | 0.831 | 0.495 |
| current | P(y2=1|X) | 2 | 0.633 | 0.279 |
| current | P(y3=1|X) | 2 | 0.808 | 0.473 |
| new | P(X) | 2 | 0.561 | 0.439 |
| new | P(y1=1|X) | 2 | 0.910 | 0.392 |
| new | P(y2=1|X) | 2 | 0.713 | 0.175 |
| new | P(y3=1|X) | 2 | 0.889 | 0.367 |

As with iteration 1, the probability of component 1 has increased, while the probability for component 2 has decreased. Item agreement is more associated with component 1 than component 2.

Now let’s look at the log likelihood.

freqs2 <- freqs2 %>%  
 mutate(ll=frq\*log(joint\_y))  
  
flextable(freqs2)

| val | frq | perc | iteration | joint\_x1 | joint\_x2 | joint\_y | x1\_y | x2\_y | frq\_x1 | frq\_x2 | ll |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1-1-1 | 696 | 0.406 | 2 | 0.4250 | 0.0653 | 0.2661 | 0.8916 | 0.108 | 620.56 | 75.4 | -921.4 |
| 1-1-2 | 68 | 0.040 | 2 | 0.1329 | 0.1290 | 0.1312 | 0.5656 | 0.434 | 38.46 | 29.5 | -138.1 |
| 1-2-1 | 275 | 0.160 | 2 | 0.2460 | 0.1685 | 0.2118 | 0.6485 | 0.351 | 178.34 | 96.7 | -426.8 |
| 1-2-2 | 130 | 0.076 | 2 | 0.0586 | 0.1880 | 0.1158 | 0.2828 | 0.717 | 36.76 | 93.2 | -280.3 |
| 2-1-1 | 34 | 0.020 | 2 | 0.0865 | 0.0667 | 0.0777 | 0.6209 | 0.379 | 21.11 | 12.9 | -86.9 |
| 2-1-2 | 19 | 0.011 | 2 | 0.0206 | 0.0744 | 0.0444 | 0.2593 | 0.741 | 4.93 | 14.1 | -59.2 |
| 2-2-1 | 125 | 0.073 | 2 | 0.0500 | 0.1722 | 0.1040 | 0.2687 | 0.731 | 33.59 | 91.4 | -282.9 |
| 2-2-2 | 366 | 0.214 | 2 | 0.0119 | 0.1920 | 0.0915 | 0.0728 | 0.927 | 26.64 | 339.4 | -875.3 |

ll\_2 <- sum(freqs2$ll)  
ll\_1

[1] -3468

ll\_2

[1] -3071

The overall log likelihood has increased, which means we are closer to our convergance (maximization of the likelihood of the data).

These iterations continue, with each iteration guaranteed to increase the log likelihood of the model. When the iterations result in negligible increases to the log likelihood, the algorithm is said to have converged and we have arrived at a solution.

It’s possible that the algorithm identified a local maximum, rather than a global maximum. Initialize a few different random starts to see if you get to the same solution. If you do not, take the solution with the highest log likelihood.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. “Maximum Likelihood from Incomplete Data Via the EM Algorithm.” *Journal of the Royal Statistical Society: Series B (Methodological)* 39 (1): 1–22. <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>.

1. ‘Current’ only denoting that we first start off with a guess of parameter values, then subsequently use the updated parameter values generated from the previous iteration of the EM algorithm. [↑](#footnote-ref-21)
2. A proof exists, but I don’t yet understand intuitively how this is possible. [↑](#footnote-ref-22)
3. This data comes from the 1987 General Social Science (GSS) survey. See the [gssr](https://kjhealy.github.io/gssr/index.html) package for data on all years. [↑](#footnote-ref-24)
4. [↑](#footnote-ref-27)
5. [↑](#footnote-ref-28)
6. [↑](#footnote-ref-32)