LSAT: Item response

Tanmay Goyal
AI20BTECH11021

Tanay Yadav
AI20BTECH11026

Abstract

The LSAT test is a 5-question multiple choice test, with students scoring 1 point for each correct answer and 0 points for an incorrect answer. This gives 32 possible marking patterns. The data consists of the number of students who achieved a particular item response pattern.

Pattern index	Item response pattern	Freq (m)
1	0 0 0 0 0	3
2	0 0 0 0 1	6
3	0 0 0 1 0	2
.	. . . . .	.
.	. . . . .	.
.	. . . . .	.
30	1 1 1 0 1	61
31	1 1 1 1 0	28
32	1 1 1 1 1	298

We wish to estimate this data using the one-parameter Rasch Model, using two commonly used sampling techniques: MCMC and Variational Inference, using the library PyMC3. The two sampling techniques were then also implemented from scratch, and compared to the library implementations.

1. Introduction

The Rasch model, named after Georg Rasch, is a psychometric model for analyzing categorical data, such as answers to questions on a reading assessment or questionnaire responses. It models the probability of a specified response as a function of the ability of a person, as well as the difficulty of the question.

In general, the probability that the i^{th} student gets the j^{th} question correct is a function of θ_i, the i^{th} students' ability, as well as α_j, the j^{th} questions' difficulty and is given by:

$$\Pr[i^{th} \text{ student gets } j^{th} \text{question correct}] = \frac{e^{f(\theta_i, \alpha_j)}}{1 + e^{f(\theta_i, \alpha_j)}}$$

where $f(.)$ is a simple relation between the questions' difficulty and the students' ability. In the most general case, we can model it as the difference between the two, i.e $f(x, y) = x - y$

2. Diving deeper into MCMC

The one-parameter Rasch Model can be approximated using the fact that the probability that the probability p_{ij} of the i^{th} student getting the j^{th} question correct is modelled as a logistic function of θ_i, i^{th} students ability and α_j, j^{th} questions' difficulty. To use Bayesian modelling, we set priors on each of the variables involved in modelling. Thus, our model looks like:

$$\theta_i \sim \mathcal{N}(0, 1)\ 1 \leq i \leq 1000$$

$$\alpha_j \sim \mathcal{N}(0, 100)\ 1 \leq j \leq 5$$

$$\beta \sim Flat(0, \infty)$$

$$logit(p_{ij}) = \beta * \theta_i - \alpha_j$$

$$r_{ij} = Bernoulli(p_{ij})$$

Note that the $Flat$ prior is implemented in PyMC3 using the $HalfFlat$ function since we want $\beta \in [0, \infty)$. Also, the $Flat$ prior can be modelled using a Gaussian distribution using:

$$\beta \sim Flat(0, \infty) \Leftrightarrow \beta \sim \mathcal{N}(0, \infty) \times \mathbf{1}(\beta \geq 0)$$

We are now in a position to compute our conditionals and posteriors. Since MCMC requires us to know the distribution we wish to sample from, we will have to calculate the exact distributions. However, if we wish to sample from $f(x)$ and we know $f(x) \propto \frac{p(x)}{N}$ where N is a normalizing constant, then we can sample from $p(x)$ as well. Thus, knowing the exact distribution can be replaced by knowing an approximate proportional distribution.

Since the likelihood is Bernoulli, we can write:

$$\mathcal{L}(x|\{\alpha_j\}_{j=1}^{Q},\{\theta_i\}_{i=1}^{N},\beta)=\prod_{i=1}^{N}\prod_{j=1}^{Q}{}_{ij}^{x_{ij}}\ (1-p_{ij})^{1-x_{ij}}$$

$$=\prod_{i=1}^{N}\prod_{j=1}^{Q}\frac{\left(e^{\beta\theta_i-\alpha_j}\right)^{x_{ij}}}{1+e^{\beta\theta_i-\alpha_j}}$$

$$=\frac{\exp\left(\sum_{i,j}(\beta\theta_i-\alpha_j)\times x_{ij}\right)}{\prod_{i=1}^{N}\prod_{j=1}^{Q}1+\exp(\beta\theta_i-\alpha_j)}$$

$$=\frac{\exp(\beta\sum_i\theta_i\sum_j x_{ij}-\sum_j\alpha_j\sum_i x_{ij})}{\prod_{i=1}^{N}\prod_{j=1}^{Q}1+\exp(\beta\theta_i-\alpha_j)}$$

$$=\frac{\exp\left(\beta\sum_{i=1}^{N}\theta_i r_i-\sum_{j=1}^{Q}\alpha_j s_j\right)}{\prod_{i=1}^{Q}\prod_{j=1}^{N}1+\exp\left(\beta\theta_i-\alpha_j\right)}$$

where r_i is the score of the student i and s_j is the score on question j.

We also have the priors:

$$\pi(\alpha)=\frac{1}{\sqrt{2\pi\sigma_\alpha^2}}exp\left(-\frac{(\alpha-\mu_\alpha)^2}{2\sigma_\alpha^2}\right)$$

$$\pi(\beta)=\frac{1}{\sqrt{2\pi\sigma_\beta^2}}exp\left(-\frac{(\beta-\mu_\beta)^2}{2\sigma_\beta^2}\right)$$

$$\pi(\theta)=\frac{1}{\sqrt{2\pi\sigma_\theta^2}}exp\left(-\frac{(\theta-\mu_\theta)^2}{2\sigma_\theta^2}\right)$$

Thus, our posterior is of the form:

$$\pi(\{\alpha_j\}_{j=1}^{Q},\{\theta_i\}_{i=1}^{N},\beta|X)$$
$$\propto\mathcal{L}(X|\{\alpha_j\}_{j=1}^{Q},\{\theta_i\}_{i=1}^{N},\beta)\prod_{i=1}^{N}\pi(\theta_i)\prod_{j=1}^{Q}\pi(\alpha_j)\pi(\beta)$$

We shall also find the conditionals for each of the variables as follows:

$$\pi(\alpha_j|\alpha_1,\ldots,\alpha_{j-1},\alpha_{j+1},\ldots,\alpha_Q,\{\theta_i\}_{i=1}^{N},\beta,x)$$
$$=\mathcal{L}\times\pi(\alpha_j)$$
$$=\frac{\exp\left(\beta\sum_{i=1}^{N}\theta_i r_i-\sum_{j=1}^{Q}\alpha_j s_j\right)}{\prod_{i=1}^{Q}\prod_{j=1}^{N}1+\exp\left(\beta\theta_i-\alpha_j\right)}\times\frac{1}{\sqrt{2\pi\sigma_\alpha}}exp\left(-\frac{\alpha_j^2}{2\sigma_\alpha^2}\right)$$
$$=\frac{\exp\left(\beta\sum_{i=1}^{N}\theta_i r_i-\sum_{j=1}^{Q}\alpha_j s_j-\frac{\alpha_j^2}{2\sigma_{\alpha^2}}\right)}{\prod_{i=1}^{Q}\prod_{j=1}^{N}1+\exp\left(\beta\theta_i-\alpha_j\right)\times\sqrt{2\pi\sigma_\alpha^2}}$$
$$=\frac{\exp\left(\beta\sum_{i=1}^{N}\theta_i r_i\right)\times\exp\left(-\sum_{k\neq j}\alpha_k s_k\right)\times\exp\left(-\alpha_j s_j-\frac{\alpha_j^2}{2\sigma_\alpha^2}\right)}{\prod_{i=1}^{Q}\prod_{j=1}^{N}1+\exp\left(\beta\theta_i-\alpha_j\right)\times\sqrt{2\pi\sigma_\alpha^2}}$$
$$\propto\frac{\exp\left(-\alpha_j s_j-\frac{\alpha_j^2}{2\sigma_\alpha^2}\right)}{\prod_{i=1}^{Q}\prod_{j=1}^{N}1+\exp\left(\beta\theta_i-\alpha_j\right)}$$

using the fact that $\mu_\alpha=0$. We remove the constants to obtain a simple and easy to compute conditional.

In a similar manner, we get the conditionals for β and θ_i as:

$$\pi(\beta|\{\alpha_j\}_{j=1}^{Q},\{\theta_i\}_{i=1}^{N},x)\propto\frac{\exp\left(\beta\sum_{i=1}^{N}\theta_i r_i-\frac{\beta^2}{2\sigma_\beta^2}\right)}{\prod_{i=1}^{Q}\prod_{j=1}^{N}1+\exp\left(\beta\theta_i-\alpha_j\right)}$$

$$\pi(\theta_i|\theta_1,\ldots,\theta_{i-1},\theta_{i+1},\ldots,\theta_N,\{\alpha_j\}_{j=1}^{Q},\beta,x)$$
$$\propto\frac{\exp\left(\beta\theta_i r_i-\frac{\theta_i^2}{2\sigma_\theta^2}\right)}{\prod_{i=1}^{Q}\prod_{j=1}^{N}1+\exp\left(\beta\theta_i-\alpha_j\right)}$$

We use the Metropolis-Hastings Algorithm to sample. The algorithm is given in Algorithm 1.

While implementing MCMC from scratch, we have taken the following assumptions/decisions:

1. We work with logarithms to prevent calculations from blowing up.

2. The log of the denominator is clipped between 0 and 2×10^5

Algorithm 1 Metropolis-Hastings Algorithm

Suppose we wish to sample from p(x).
1. Select an initial point x_0, which can be done randomly or based on the prior.
2. Let $x_{curr} = x_0$
3. For the desired number of steps or till convergence,
   4. Find x^* using the proposal distribution $q(x_{curr}, x^*)$
   5. Find the acceptance probability $\alpha = \min\{1, \frac{p(x^*)}{p(x_{curr})}\}$
   6. Sample $u \sim \mathcal{U}[0, 1]$.
   7.If $u < \alpha$, accept the point x^* and set $x_{curr} = x^*$. Else reject the point and stay at x_{curr}.

3. We have reduced σ_α to be 1 and σ_β to be 10. This prevents the numbers from blowing up.

4. The proposal distributions are Gaussians with $\sigma_\alpha, \sigma_\theta = 2$ and $\sigma_\beta = 0.5$ and mean as the current point in parameter space. This again is to limit the exploration space and prevent the blowing up of numbers.

5. We find that a lot of samples per chain causes the answers to stray from convergence. Thus, it makes sense to have a lot of chains with a lesser number of samples. This ensures we can start randomly at many points and explore from there. A simple example of why restarting random walks is often better is as follows:

   Suppose $G = (V, E)$ and $V = \{1, 2, 3, \ldots 10^6\}$. Suppose the probability of reaching vertices $1, 2 \ldots 10$ equals $\frac{1}{2}$. The other vertices have lower probability, but are very densely connected, such that it is difficult to exit once we enter the space of these vertices. In this case, if we wish to keep doing a random walk till we reach a vertex numbered lower than 10,the expected steps would be 2, but if we end up in the dense region, we cannot guarantee the number of samples it would take. In this case restarting the walk is always a better option.

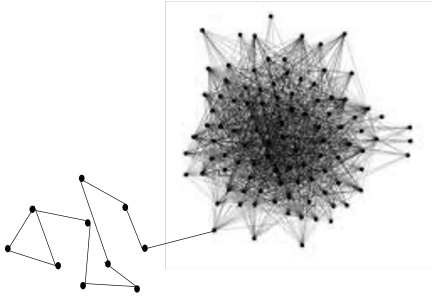

Figure 1. An example of a graph where restarting random walk would be better.

The results are tabulated below:

No.	Samples	α_0	α_1	α_2	α_3	α_4	β	Acc.
1.	20000	0.75	0.67	-0.57	-0.52	-0.26	1.58	74.22
2.	3500	1.56	-0.59	-1.29	1.37	-1.05	5.00	49.50
3.	7000	-0.40	-0.70	0.52	0.06	0.53	7.98	51.4
4.	14000	-0.77	0.7	-0.12	-0.59	0.78	5.92	51.92

Here, row number 1 was done using PyMC3. It consisted of 4 chains with 1000 burn-in steps and 5000 steps. The next 3 rows were done using the implementation from scratch. Each of them had a burn-in period of 10 steps and 140 samples after that. The number of chains was kept to 25 for row 2, 50 for row 3, and 100 for row 4.

We see that as we increase the chains, more and more samples help improve the accuracy.

The error was calculated using Hamming Distance since the outputs could be interpreted as binary strings. Predictions were made using the mean values of the parameters and calculating the logistic function. If we got a predicted probability of greater than 0.5, we assumed the student will get the question correct, else it would answer the question incorrectly. The accuracy was then simply calculated as the number of positions in which are predicted output and true output agree.

3. Variational Inference: Mean field Inference

Variational Inference or Variational Bayes is another technique to compute an approximate distribution to the posterior. It is used when the posterior has hard-to-compute integrals and can not be expressed in a closed form. We then assume the distribution to be approximated has its own set of variational parameters, and by finding the best fit for these variational parameters by optimizing it, we can find the approximated distribution.

Let us assume our theoretical posterior to be found is given by $p(\theta|x)$ where θ is representational of the unknown parameters and/or latent parameters. We assume that $q(\theta)$ is the approximate distribution to p. Then, we wish to reduce the KL divergence between the two distributions.

$$\begin{aligned} KL[q||p] &= \int q(\theta) \log \left[\frac{q(\theta)}{p(\theta|x)}\right] \, d\theta \\ &= \int q(\theta) \log \left[\frac{q(\theta)p(x)}{p(\theta, x)}\right] \, d\theta \\ &= \int q(\theta) \log \left[\frac{q(\theta)}{p(\theta, x)}\right] \, d\theta + \log p(x) \int q(\theta) \, d\theta \\ &= \int q(\theta) \log \left[\frac{q(\theta)}{p(\theta, x)}\right] \, d\theta + \log p(x) \end{aligned}$$

$$\implies \log p(x) = KL[q||p] + \int q(\theta) \log \left[\frac{p(\theta, x)}{q(\theta)} \right] \, d\theta$$

$$\implies \log p(x) = KL[q||p] + \mathbb{E}_q[p(\theta, x)] - \mathbb{E}_q[q(\theta)]$$

From here, we get that minimizing the KL divergence between $q(\theta)$ and $p(\theta, x)$ is the same as maximizing the Evidence Lower Bound, or ELBO given by:

$$ELBO(q) = \mathbb{E}_q[p(\theta, x)] - \mathbb{E}_q[q(\theta)]$$

The ELBO is named so because we see that since the KL divergence is non negative, we have:

$$\log p(x) \geq ELBO(q)$$

i.e ELBO is a lower bound on the evidence.

The mean field inference allows us to assume the approximate distribution factorizes as:

$$q(\theta) = q(\theta_1, \theta_2, \dots \theta_n) = \prod_{i=1}^{n} q_i(\theta_i)$$

Thus, our ELBO term now becomes:

$$\int_{\theta_1 \theta_2 \dots \theta_n} \prod_{i=1}^{n} q_i(\theta_i) \left[\log p(\theta, x) - \sum_{i=1} \log q_i(\theta_i) \right] \, d\theta$$

$$= \int_{\theta_j} q_j(\theta_j) \int_{\theta_{m|m \neq j}} \prod_{i \neq j} q_i(\theta_i) \left[\log p(\theta, x) - \sum_{i=1}^{n} \log q_i(\theta_i) \right] \, d\theta$$

$$= \int_{\theta_j} q_j(\theta_j) \left[\int_{\theta_{m|m \neq j}} \prod_{i \neq j} q_i(\theta_i) \log p(\theta, x) \, d\theta_1 d\theta_2 \dots d\theta_n \right] -$$

$$\int_{\theta_j} q_j(\theta_j) \left[\int_{\theta_{m|m \neq j}} \prod_{i \neq j} q_i(\theta_i) \sum_{i=1}^{n} \log q_i(\theta_i) \, d\theta_1 d\theta_2 \dots d\theta_n \right]$$

We denote

$$\mathbb{E}_{m:m \neq j}[\log p(x, \theta)]$$
$$= \int_{\theta_{m|m \neq j}} \prod_{i \neq j} q_i(\theta_i) \log p(\theta, x) d\theta_1 \dots d\theta_{j-1} d\theta_{j+1} \dots d\theta_n$$

Thus, our first term now becomes

$$\int_{\theta_j} q_j(\theta_j) \mathbb{E}_{m:m \neq j}[\log p(x, \theta)] \, d\theta_j$$

Our second term can be written as:

$$\int_{\theta_j} q_j(\theta_j) \log q_j(\theta_j) \int_{\theta_{m|m \neq j}} \prod_{i \neq j} q_i(\theta_i) d\theta_1 \dots d\theta_{j-1} d\theta_{j+1} \dots d\theta_n$$

$$- \int_{\theta_j} q_j(\theta_j) \left[\int_{\theta_{m|m \neq j}} \prod_{i \neq j} q_i(\theta_i) \sum_{i \neq j} \log q_i(\theta_i) \, d\theta_1 d\theta_2 \dots d\theta_n \right]$$

$$= \int_{\theta_j} q_j(\theta_j) \log q_j(\theta_j) d\theta_j - F(\theta_1, \dots \theta_{j-1}, \theta_{j+1} \dots \theta_n)$$

Thus, our overall ELBO term can be written as:

$$ELBO(q) = \int_{\theta_j} q_j(\theta_j) [\mathbb{E}_{m:m \neq j}[\log p(x, \theta)] - \log q_j(\theta_j)] + F(\theta_{-j})$$

where $\theta_{-j} = \theta_1, \dots \theta_{j-1}, \theta_{j+1}, \dots \theta_n$

Now, we wish to maximize our ELBO term. Using Lagrange Multipliers, we can write the term to be maximized as

$$l = ELBO(q) - \sum_{i=1}^{n} \lambda_i \int_{\theta_i} q_i(\theta_i) \, d\theta_i$$

where the second term involving the Lagrange Multipliers enforce the condition of the probability distributions summing to unity. On taking partial derivative wrt $q_j(\theta_j)$ we get:

$$\frac{\partial L}{\partial q_j(\theta_j)} = q_j(\theta_j)[\mathbb{E}_{m:m \neq j}[\log p(x, \theta)] - \log q_j(\theta_j)] - \lambda_j q_j(\theta_j)\}$$
$$= \mathbb{E}_{m:m \neq j}[\log p(x, \theta)] - \log q_j(\theta_j)] - 1 - \lambda_j$$
$$= 0$$

Solving this, we get

$$q_j(\theta_j) = \frac{\exp(\mathbb{E}_{m:m \neq j}[\log p(x, \theta)])}{NC}$$

where NC is the normalizing constant.

Since this might involve terms requiring expectations of various variables, which may be unknown or latent, we may not find a closed form in each case. However, we can use an iterative algorithm such as the Expectation-Maximization algorithm to find the solution. This involves starting with an initial guess, and then iteratively updating each variable and using the updated value to update the other.

For our problem, let us compute the logarithm of the joint probability.

$$\log \mathcal{L}(x, \{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i=1}^{N}, \beta)$$
$$= \log \mathcal{L}(X|\{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i=1}^{N}, \beta) + \sum_{i=1}^{N} \log \pi(\theta_i) +$$
$$\sum_{j=1}^{Q} \log \pi(\alpha_j) + \log \pi(\beta)$$
$$= \beta \sum_{i=1}^{N} \theta_i r_i - \sum_{j=1}^{Q} \alpha_j s_j - \sum_{i=1}^{Q} \sum_{j=1}^{N} \log\left(1 + \exp\left(\beta\theta_i - \alpha_j\right)\right)$$
$$+ Q \log\left(\frac{1}{\sqrt{2\pi\sigma_\alpha^2}}\right) + \log\left(\frac{1}{\sqrt{\pi\sigma_\beta^2}}\right) + N \log\left(\frac{1}{\sqrt{2\pi\sigma_\theta^2}}\right)$$
$$- \sum_{j=1}^{Q} \frac{\alpha_j^2}{2\sigma_\alpha^2} - \sum_{i=1}^{N} \frac{\theta_i^2}{2\sigma_\theta^2} - \frac{\beta^2}{2\sigma_\beta^2}$$

Also, note that $\mathbb{E}[1 + \exp\left(\beta\theta_i - \alpha_j\right)]$ can be expanded using Taylor Series in linear and quadratic terms. The linear term expectations shall all become zero and the quadratic term expectations shall all become constant and hence, we push them into the constant term. We shall ignore the term for further calculations.

Thus, after removing the constants, we get the logarithm of the joint probability as

$$\log \mathcal{L}(x, \{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i=1}^{N}, \beta)$$
$$= \beta \sum_{i=1}^{N} \theta_i r_i - \sum_{j=1}^{Q} \alpha_j s_j - \sum_{j=1}^{Q} \frac{\alpha_j^2}{2\sigma_\alpha^2} - \sum_{i=1}^{N} \frac{\theta_i^2}{2\sigma_\theta^2} - \frac{\beta^2}{2\sigma_\beta^2} + C$$

We assume the posterior has no fixed distribution or form and is factorizable as:

$$p(\{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i=1}^{N}, \beta) \approx q(\{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i=1}^{N}, \beta)$$
$$= \prod_{j=1}^{Q} q_{\alpha_j}(\alpha_j) \prod_{i=1}^{N} q_{\theta_i}(\theta_i) \times q_\beta(\beta)$$

Now, we note the following:

$$\mathbb{E}[\alpha_j] = \mathbb{E}[\beta] = \mathbb{E}[\theta_i] = 0$$

$$\mathbb{E}[\alpha_j] = \sigma_\alpha^2 \ ; \ \mathbb{E}[\theta_i] = \sigma_\theta^2 \ ; \ \mathbb{E}[\beta_j] = \sigma_\beta^2$$

$$\mathbb{E}[\beta\theta] \approx \mathbb{E}[\beta]\mathbb{E}[\theta] - r\sigma_\beta\sigma_\theta$$

Where r is the degree of correlation, we assume to be 1.

We have,

$$\log q_{\alpha_k}(\alpha_k) = \mathbb{E}_{\{\alpha_j\}_{j\neq k}, \{\theta_i\}_{i=1}^{N}, \beta}[\log \mathcal{L}]$$
$$= \sum_{i=1}^{N} \mathbb{E}[\beta\theta_i] r_i - \sum_{j\neq k} \mathbb{E}[\alpha_j] s_j - \sum_{j\neq k} \frac{\mathbb{E}[\alpha_j^2]}{2\sigma_\alpha^2} -$$
$$\sum_{i=1}^{N} \frac{\mathbb{E}[\theta_i^2]}{2\sigma_\theta^2} - \frac{\mathbb{E}[\beta^2]}{2\sigma_\beta^2} + C$$
$$= \sum_{i=1}^{N} \sigma_\theta\sigma_\beta r_i - \sum_{j=1}^{Q} \frac{\sigma_\alpha^2}{2\sigma_\alpha^2} - \sum_{i=1}^{N} \frac{\sigma_\theta^2}{2\sigma_\theta^2} - \frac{\sigma_\beta^2}{2\sigma_\beta^2} + C$$
$$= \sum_{i=1}^{N} \sigma_\theta\sigma_\beta r_i + C$$

$$\log q_\beta(\beta) = \mathbb{E}_{\{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i=1}^{N}}[\log \mathcal{L}]$$
$$= \sum_{i=1}^{N} \beta\mathbb{E}[\theta_i] r_i - \sum_{j=1}^{Q} \mathbb{E}[\alpha_j] s_j - \sum_{j=1}^{Q} \frac{\mathbb{E}[\alpha_j^2]}{2\sigma_\alpha^2} - \sum_{i=1}^{N} \frac{\mathbb{E}[\theta_j^2]}{2\sigma_\theta^2}$$
$$= 0$$

$$\log q_{\theta_k}(\theta_k) = \mathbb{E}_{\{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i\neq k}, \beta}[\log \mathcal{L}]$$
$$= \sum_{i\neq k} \mathbb{E}[\beta\theta_i] r_i - \sum_{j=1}^{Q} \mathbb{E}[\alpha_j] s_j - \sum_{j=1}^{Q} \frac{\mathbb{E}[\alpha_j^2]}{2\sigma_\alpha^2} -$$
$$\sum_{i\neq k} \frac{\mathbb{E}[\theta_i^2]}{2\sigma_\theta^2} - \frac{\mathbb{E}[\beta^2]}{2\sigma_\beta^2} + C$$
$$= \sum_{i\neq k}^{N} \sigma_\theta\sigma_\beta r_i - \sum_{j=1}^{Q} \frac{\sigma_\alpha^2}{2\sigma_\alpha^2} - \sum_{i\neq k} \frac{\sigma_\theta^2}{2\sigma_\theta^2} - \frac{\sigma_\beta^2}{2\sigma_\beta^2} + C$$
$$= \sum_{i\neq k} \sigma_\theta\sigma_\beta r_i + C$$

Thus, we see that none of our expressions require any iterative approach to solving them. This is because there are no hyperpriors and all the parameters of the priors are fixed and constant. Thus, there are no latent parameters to be estimated. Thus, we get the approximate distributions as:

$$q_{\alpha_k}(\alpha_k) \sim e^{\sum_{i=1}^{N} \sigma_\theta\sigma_\beta r_i}$$

$$q_\beta(\beta) \sim e^0$$

$$q_{\theta_k}(\theta_k) \sim e^{\sum_{i\neq k} \sigma_\theta\sigma_\beta r_i}$$

However, these imply our posterior is a constant distribution, which seems incorrect.

Let us now assume a certain form of posterior. We assume that

$$q_\alpha(\alpha) \sim \mathcal{N}(\mu_\alpha, 1)$$
$$q_\beta(\beta) \sim \mathcal{N}(\mu_\beta, \frac{1}{2})$$
$$q_\theta(\theta) \sim \mathcal{N}(\mu_\theta, 1)$$

We also assume independence between all our variables, arriving at the following posterior,

$$q = \prod_{i=1}^{N} \mathcal{N}(\mu_\theta, 1) \prod_{j=1}^{Q} \mathcal{N}(\mu_\alpha, 1) \times \mathcal{N}(\mu_\beta, \frac{1}{2})$$

Our optimal posterior q^* is given by:

$$\begin{aligned} q^* &= \arg\max \mathbb{E}_{z\sim q(z)}[ELBO(q)] \\ &= \arg\max \mathbb{E}_{z\sim q(z)} \left[\log\left(\frac{p(\vec{z}, x)}{q(\vec{z})}\right)\right] \\ &= \arg\max \left\{\mathbb{E}_{z\sim q(z)} \left[\log p(\vec{z}, x)\right] - \mathbb{E}_{z\sim q(z)} \left[\log q(\vec{z})\right]\right\} \end{aligned}$$

where $\vec{z}$ is the set of all parameters.

We have already found the logarithm of the joint probability $\log \mathcal{L}(x, \{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i=1}^{N}, \beta)$. Taking the expectation results in:

$$\begin{aligned} &\mathbb{E}\left[\log \mathcal{L}(x, \{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i=1}^{N}, \beta)\right] \\ &= \mu_\beta\mu_\theta \sum_{i=1}^{N} r_i - \mu_\alpha \sum_{j=1}^{Q} s_j - \sum_{i=1}^{N}\sum_{j=1}^{Q} \mathbb{E}[\log(1 + \exp(\beta\theta_i - \alpha_j))] \end{aligned}$$

$\mathbb{E}[\log(1 + \exp(\beta\theta_i - \alpha_j))]$ can be calculated by substituting $\beta\theta_i - \alpha_j$ as k and then applying Taylor Series expansions:

$$\begin{aligned} \log(1 + \exp(k)) &= \exp(k) - \frac{exp(k)}{2} \\ &= 1 + k + \frac{k^2}{2} - \frac{(1 + k + \frac{k^2}{2})^2}{2} \\ &= \frac{1}{2} - \frac{k^4}{8} - \frac{k^3}{2} - \frac{k^2}{2} \\ &= -\frac{1}{2}(\beta^2\theta_i^2 + \alpha_j^2 - 2\alpha_j\beta\theta_i) \end{aligned}$$

ignoring constant and higher moment terms. Thus, we get $\mathbb{E}[\log(1 + \exp(\beta\theta_i - \alpha_j))] = \mu_\alpha\mu_\beta\mu_\theta$

Thus, we have

$$\begin{aligned} &\mathbb{E}\left[\log \mathcal{L}(x, \{\alpha_j\}_{j=1}^{Q}, \{\theta_i\}_{i=1}^{N}, \beta)\right] \\ &= \mu_\beta\mu_\theta \sum_{i=1}^{N} r_i - \mu_\alpha \sum_{j=1}^{Q} s_j - NQ\mu_\alpha\mu_\beta\mu_\theta \end{aligned}$$

The next term $\mathbb{E}[\log q(\vec{z})]$ shall be all constant using the fact that the variances of all the Gaussians are constant and $\mathbb{E}[(x - \mu_x)^2] = Var(x)$.

Thus, finally we get:

$$q* = \arg\max ELBO(q)$$

$$q^* = \arg\max \left\{\mu_\beta\mu_\theta \sum_{i=1}^{N} r_i - \mu_\alpha \sum_{j=1}^{Q} s_j - NQ\mu_\alpha\mu_\beta\mu_\theta\right\}$$

On taking the partial derivatives wrt $\mu_\alpha, \mu_\beta, \mu_\theta$:

$$\frac{\partial ELBO(q)}{\partial \mu_\alpha} = -\sum_{j=1}^{Q} s_j - NQ\mu_\beta\mu_\theta = 0$$
$$\implies \mu_\beta\mu_\theta = \frac{-\sum s_j}{NQ}$$

$$\frac{\partial ELBO(q)}{\partial \mu_\beta} = \mu_\theta \sum_{i=1}^{N} r_i - NQ\mu_\alpha\mu_\theta = 0$$
$$\implies \mu_\theta = 0 \text{ or } \mu_\alpha = \frac{\sum r_i}{NQ}$$

$$\frac{\partial ELBO(q)}{\partial \mu_\theta} = \mu_\beta \sum_{i=1}^{N} r_i - NQ\mu_\alpha\mu_\beta = 0$$
$$\implies \mu_\beta = 0 \text{ or } \mu_\alpha = \frac{\sum r_i}{NQ}$$

Thus, we set $\mu_\alpha = \frac{\sum r_i}{NQ}$. We see that μ_β and μ_θ are of opposite signs. Since, $\beta > 0$, $\mu_\beta > 0$. We let $|\mu_\beta| = |\mu_\theta| = \sqrt{\frac{\sum s_j}{NQ}}$

We allow some fluctuations in the different α_j and θ_i, sampling their means from $\mathcal{N}(\mu_\theta, 1)$ and $\mathcal{N}(\mu_t heta, 1)$. The results are tabulated below:

No.	α_0	α_1	α_2	α_3	α_4	β	Accuracy
1.	-0.01	0.01	-0.02	0.03	-0.01	0.77	79.58
2.	-0.80	-1.16	-0.53	-1.5	-0.62	1.122	46.46

Here, row 1 was done using PyMC3. The second row uses the implementation from scratch. Similar to the MCMC implementation, the calculation for error involves Hamming Distance since the predicted output can be interpreted as a binary string.If we got a predicted probability of greater than 0.5, we assumed the student will get the question correct, else it would answer the question incorrectly. The accuracy was then simply calculated as the number of positions in which are predicted output and true output agree.