ECI 2017 Bayesian Models’ Answers

By Julián Bayardo[[1]](#footnote-1)

Table of Contents

[Remark 1](#_Toc491648086)

[Questions 1](#_Toc491648087)

[Answers for the default data 2](#_Toc491648088)

[Question 1 2](#_Toc491648089)

[Question 2 3](#_Toc491648090)

[Question 3 5](#_Toc491648091)

[Question 4 6](#_Toc491648092)

[Answers for the generated data 7](#_Toc491648093)

[Question 1 7](#_Toc491648094)

[Question 2 8](#_Toc491648095)

[Question 3 9](#_Toc491648096)

[Question 4 10](#_Toc491648097)

[References 10](#_Toc491648098)

# Remark

I wrote this document in order to avoid polluting the IPython notebook and making it hard to understand. All of the parameters for plots and experiments can be found on the first cell of the accompanying IPython notebook; plots can be regenerated by changing the parameters and running all cells again.

# Questions

1. What can you say about the obtained posterior distributions? What do they represent? How do these posterior distribution compare to the parameter estimates obtained from the EM algorithm?
2. Sample from the approximate posterior distribution and plot the GMM distributions corresponding to all the samples into a single figure. Comment on this plot. What do the individual GMM distributions represent?
3. Now average all the samples from the previous step. What can you say about the obtained average distribution? What does it represent?
4. How does the posterior predictive distribution compare to:
   1. The true training data distribution
   2. The GMM obtained using ML training (i.e. using EM algorithm)
   3. The average of GMM distributions obtained in the previous step by sampling

Regenerate all the plots with a larger number of training observations and comment on how they change from the previous experiments with a smaller training dataset.

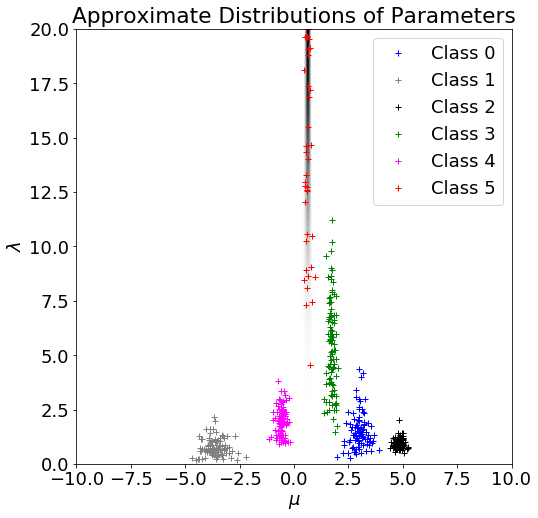
# Answers for the default data

## Question 1

The posterior distributions express the uncertainty over the values of a subset of parameters to our model. In the case of the approximate prior distribution of weights q(pi), each of the values represents the probability that one particular gaussian is used to generate the data; while q(mu, lambda) is the probability that one particular set of parameters is fed into the normal distribution that is used to generate the data when that specific component is selected.

Since q(pi) is also a proxy to how much data is given for one particular q(mu, lambda) to be approximated, it also means that there will be more confidence (i.e. lower variance) for the q(mu, lambda) distributions that have more data. This can be clearly seen in the plot below: class 2 has the highest probability in the approximate distribution, and accordingly a highly peaked distribution; while other classes have similar lower values, and are less concentrated.

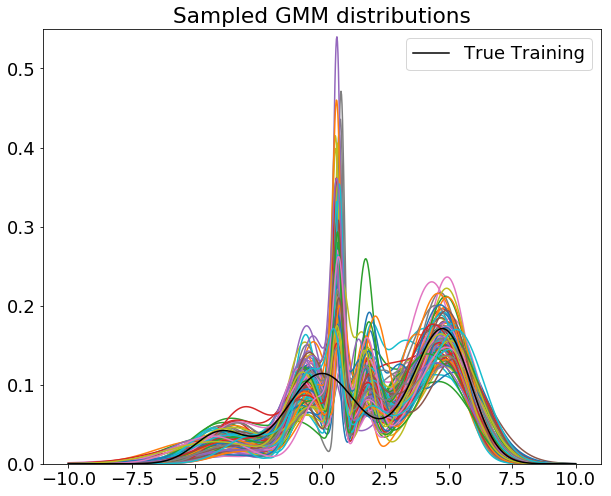
The EM parameters are akin to taking one sample from the approximate posterior distributions: the EM algorithm converges to a single set of parameters for the normal distributions, instead of a distribution over their possible parameters.

Approximate posterior distribution of weights (q(pi)): [0.09530115, 0.0962085, 0.40421586, 0.10629599, 0.19582768, 0.10215082]

## Question 2

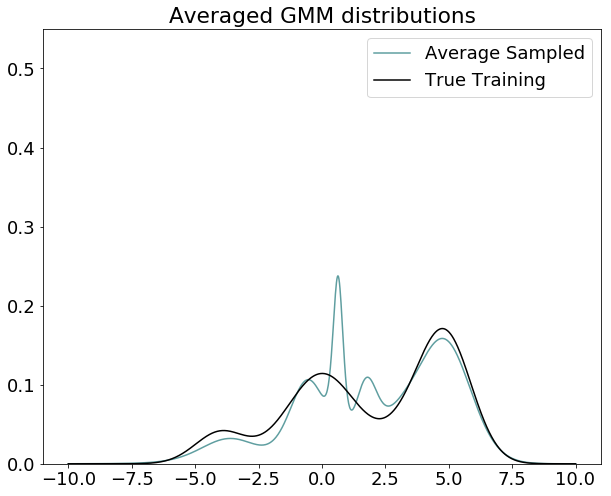
Each sample from q(pi, mu, lambda) is one possible set of parameters for the Bayesian inference network according to the approximate posterior distribution over the parameters. Hence, plotting the distribution given by the sample corresponds to one possible probability distribution over the data generated by the GMM process as learned by the VBGMM algorithm.

How much variability is seen in the sampled distributions is also a proxy for the uncertainty over the parameters to the model: the variability is inversely proportional to the certainty. If there was enough data to “be sure” about our parameters, all of the distributions would be within some small interval of the averaged distribution.



## Question 3

Averaging the distributions obtained by sampling from q(pi, mu, lambda):

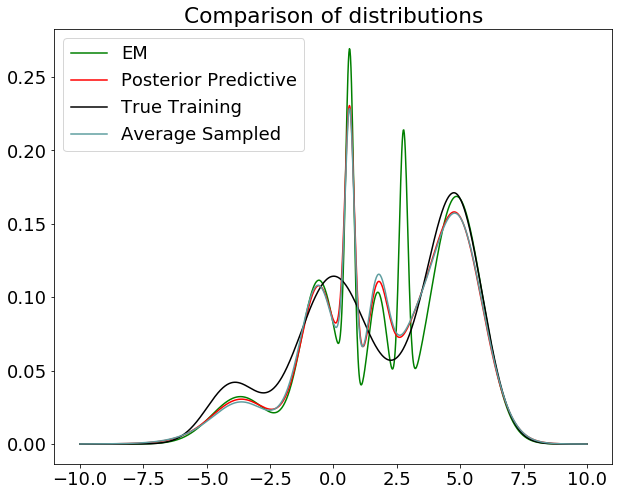


Comparing the previous plot with this one, it is clear that it resembles the distributions from the previous plot, and has the same “mistake” trends (i.e. the humps at ~0.5 and ~2.0).

Intuitively and informally, it makes sense to see this happening: it is what one would expect from the law of large numbers if it were applied to a random variable sample with range over the probability distributions that the algorithm can represent given the dataset, and probability distribution by “learnability” of a distribution using the algorithm.

Aside from that, it is interesting to see that the plot has 5 peaks, while the distribution learned by the EM algorithm has 6. This means that the VB algorithm is effectively “removing” one class (remember that C=6), either by weighting it too low, moving it out of range, or just compensating by making it overlap with another. This is indeed better behavior, although it’d be expected for the algorithm to merge the three classes that form the three unexpected peaks in the center.

## Question 4



There are a few key things to point out from this plot:

* The average sampled distribution follows the posterior predictive distribution really closely. This makes sense as the posterior predictive distribution is taking into consideration the uncertainty over the parameters that the averaged is not taking by itself. It would be expected that it should approach the posterior predictive distributions if enough samples were added, as the confidence in each set of parameters would be accounted for in the generating process for the samples.
* The EM algorithm converges to a solution that has more -and more pronounced- peaks; I can only assume this is related to the algorithm falling into singularities of the log likelihood and collapsing some of the gaussians unto a single data point (Bishop, 2006).
* There are peaks in the posterior predictive and average sampled distributions. The book mentions these peaks are because of the same “gaussian component collapsing into a single point” problem, but these can be removed (Bishop, 2006) by introducing a sufficiently strong prior over the parameters and using the maximum a posteriori estimate instead of maximum likelihood (which is not done in this implementation).
* It seems like this particular dataset is problematic: at every iteration of the training algorithm, the parameters changed by about ~9.6 in norm 2 (i.e. it never really converged). Every run would reduce the error within the first iteration and then seemingly get stuck alternating until the artificial limit in iterations.

# Answers for the generated data

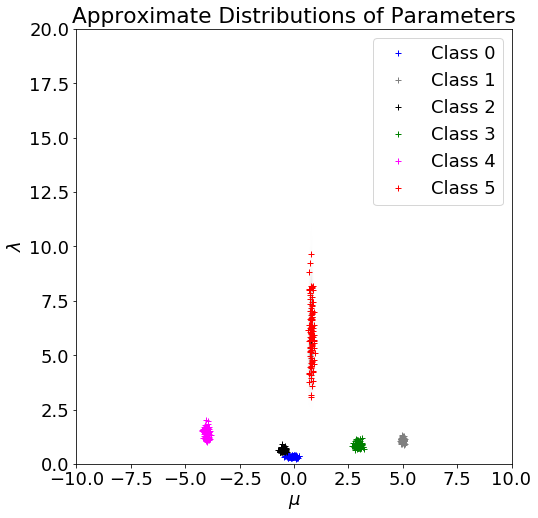
I regenerated the data using the same process, but with N=1000 for the training set; everything else is left the same as in the previous experiments.

## Question 1

Looking at the plot now, there is a much more clustered distribution for all parameters except component 5 (in red). However, the posterior for pi is very low for component 5, which means the algorithm learnt to disregard component 5 entirely; thus, it does not really contribute to the process.

Notice as well that components 0 and 2 have parameters that are really close together; this means that they will be “collapsed” into almost the same distribution. Concretely, the algorithm learned to “erase” the fifth component and “merge” together components 0 and 2.

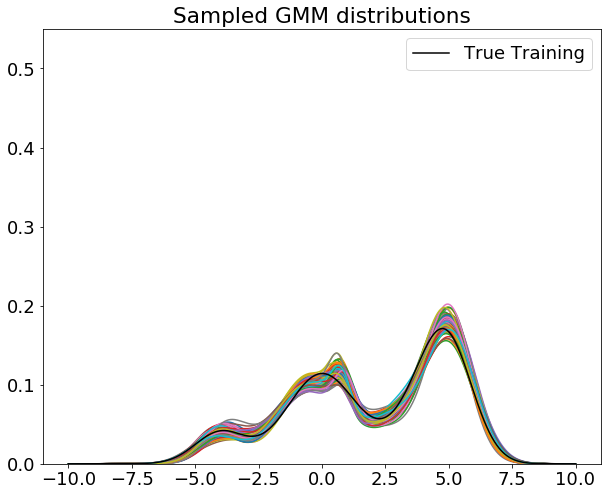
Approximate posterior distribution of weights (q): [0.13394064, 0.40431339, 0.21870086, 0.12229741, 0.07831246, 0.04243523]



## Question 2

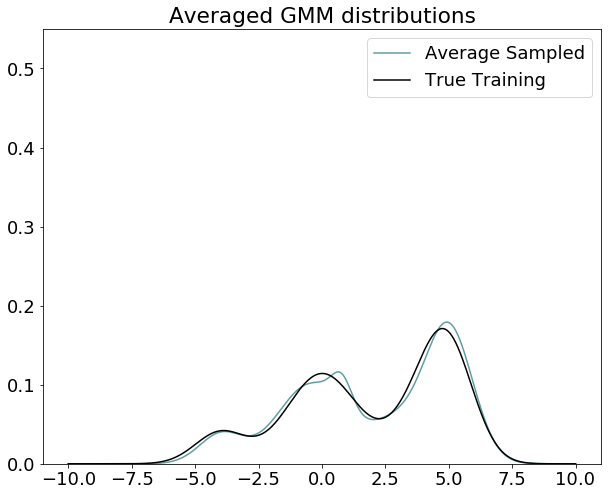
Here I sampled the parameters from the posterior distributions and plotted the GMMs as before. Observe that there are indeed 4 “mounds” corresponding to each one of the components as expected from the analysis in the previous question; and notice that around 0 we have exactly 2 peaks, corresponding to the “merged together” distributions.

As developed in the answer to question 2 with the original data, the generated GMMs are indeed clustered around a much smaller interval near the true training distribution, corresponding to the heightened certainty on the parameters to the model.



## Question 3

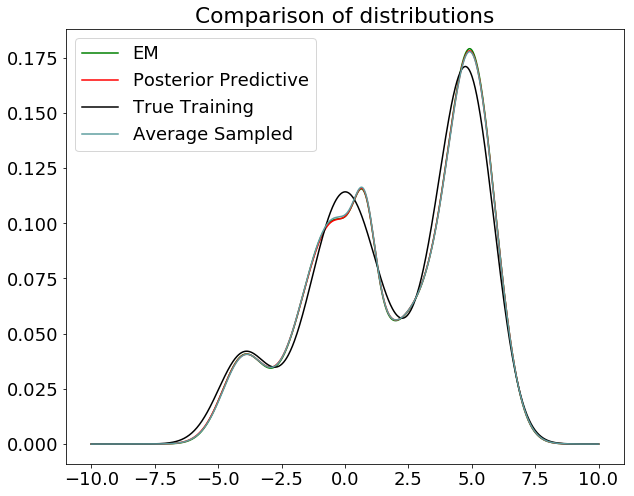
Observe that the average sampled distribution again corresponds in shape to what one would expect from the previous plot, as mentioned in the answer to question 3 in the previous section. However, it is now much better fit to what the training distribution looks like.



## Question 4

Unlike the case from the previous section, this run actually managed to converge (below the 0.00000001 threshold for change between parameters in iterations) within 2 runs after starting from the EM-learned parameters.

Observe that the average sampled distribution is strikingly similar to the EM algorithm’s; which means we did not gain much by using the VBGMM algorithm (although it has to be noted that we got lucky in that the EM algorithm did not fall into any singularity).



# References

**Bishop Cristopher M.** Pattern Recognition and Machine Learning [Book]. - Cambridge, UK : Springer, 2006.

1. [julian@bayardo.info](mailto:julian@bayardo.info). LU: 850/13. DNI: 37 835 049. [↑](#footnote-ref-1)