ECI 2017 Bayesian Models' Answers

By Julián Bayardo[1]

Table of Contents

[1] julian@bayardo.info. LU: 850/13. DNI: 37 835 049.

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:
    a. The true training data distribution
    b. The GMM obtained using ML training (i.e. using EM algorithm)
    c. 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.

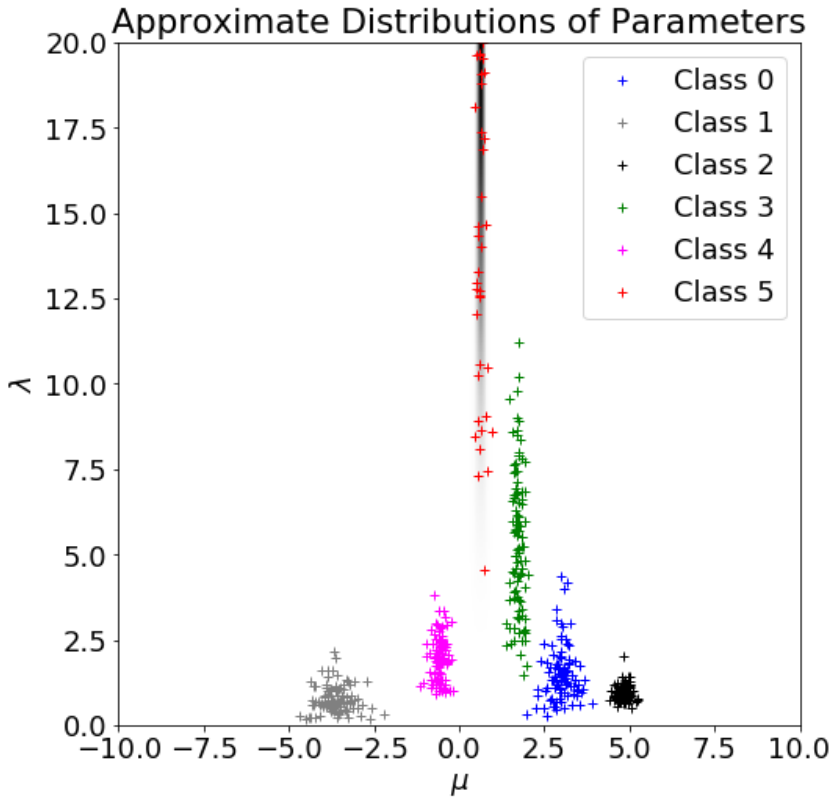
Answers for the default data

Question 1

The posterior distributions express the uncertainty over the values of a subset of parameters to our model. For the approximate prior distribution of weights q(pi), each value represents the probability that one component is selected to generate the data; while q(mu, lambda) is a distribution over the parameters to each gaussian.

Observe in the figure below that the highest value for q(pi) corresponds to class 2, which is also the most clustered: it would be expected that the distribution will almost certainly have mu ~ 5 and lambda ~ 1. It is also clear that there is little overlap between the values of mu, and hence it could be expected to see distinctive gaussian "mounds" in the GMM distributions.

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



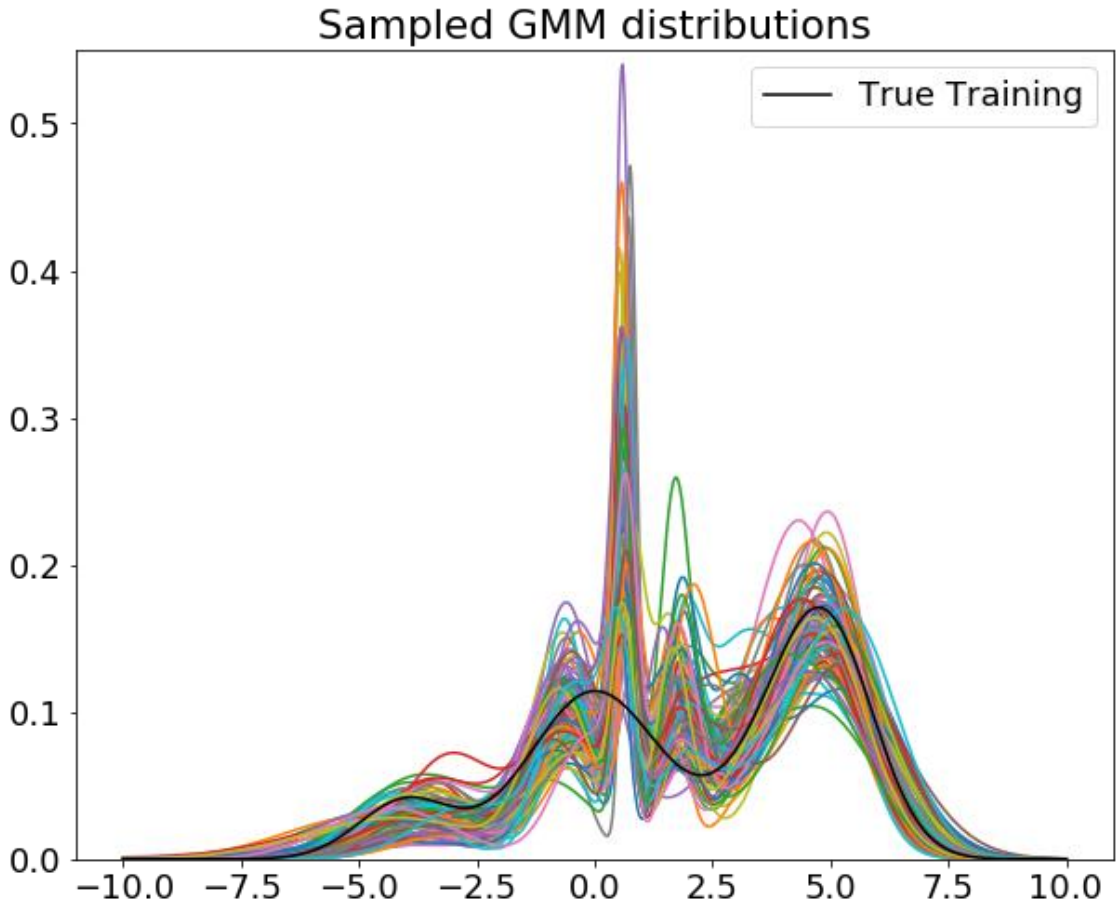
The EM method outputs probability distributions over the latent variables and maximum likelihood estimates for the **parameters of the gaussians**. In contrast, the variational method with mean field approximation outputs **parameters to the distributions used to approximate the parameters** to both the gaussians and latent variables (i.e. *hyperparameters*). A priori, the EM parameters could be one particular instance sampled from the learned distributions by the VBGMM algorithm[2].

[2] This is not exactly true: EM may converge to parameters that are very low probability for the distributions generated by the VBGMM, and VBGMM could output values that EM could never reach with that particular dataset because it would get stuck. Point is, inductive biases should be taken into account.

Question 2

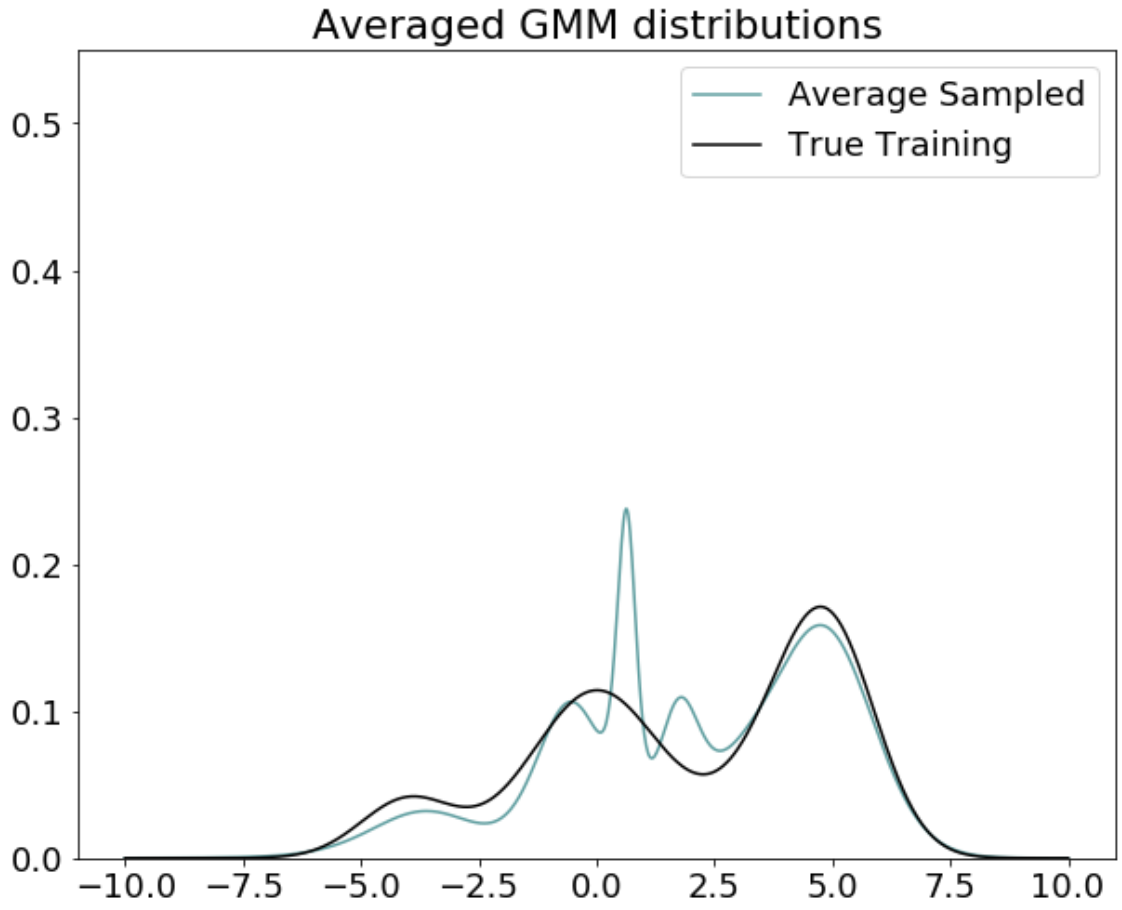
Each sample from q(pi, mu, lambda) is one possible set of parameters for the Bayesian inference network according to the approximate posterior distribution over the parameters. Hence, plotting the distribution given by the sample corresponds to one possible probability distribution over the data generated 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 (or the variance of the joint distribution over the parameters): the variability is inversely proportional to the certainty. If there was confidence on the parameters' values, all of the distributions would be close to the posterior predictive distribution.



Question 3

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

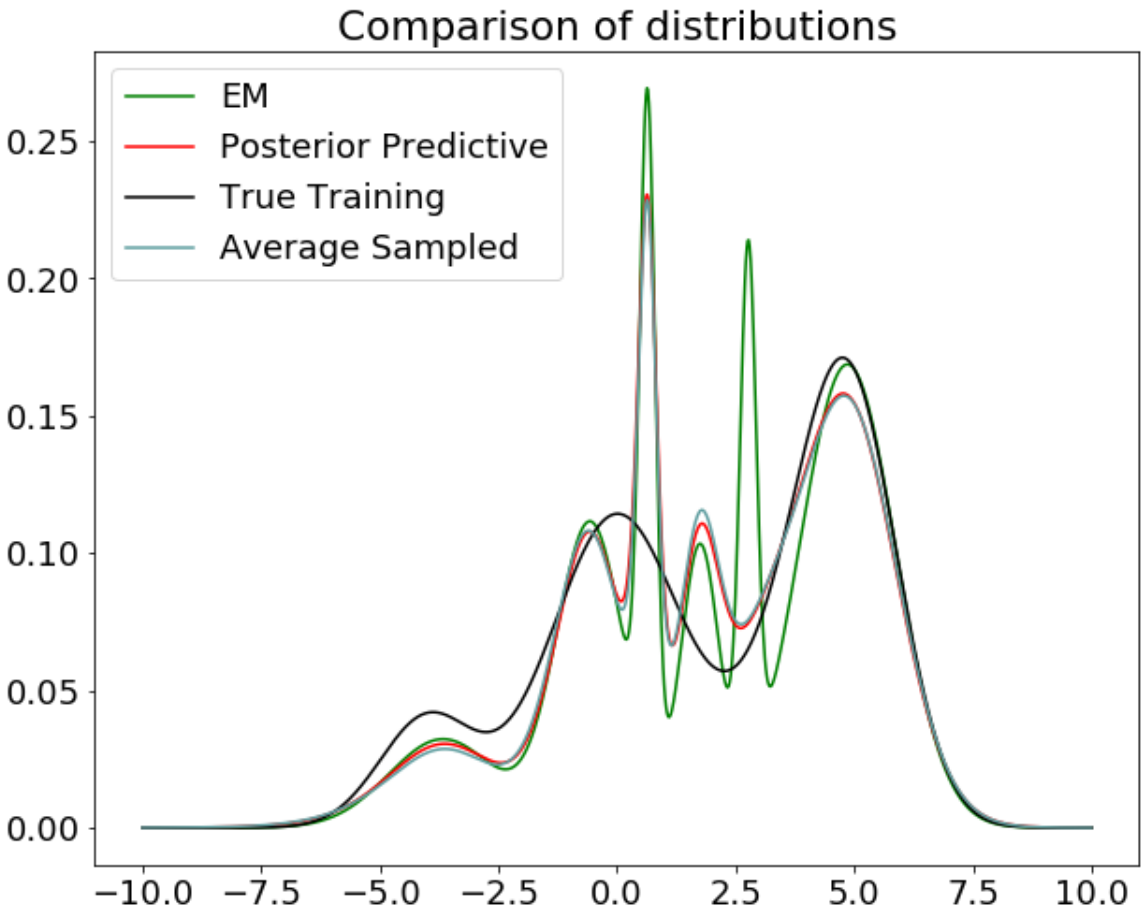


Comparing the previous plot with this one, it is clear that it resembles the distributions from the previous plot, and has the same "mistake" trends (i.e. the humps at ~0.5 and ~2.0).

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 closely. This makes sense as the posterior predictive distribution considers the uncertainty over the parameters that the averaged does not. With enough samples, the averaged distribution would approach the posterior predictive distribution, as the confidence would be accounted for in the generating process for the samples. This is easy to see by averaging more samples (i.e. 10000).
- 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” (as in the previous bullet point), 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

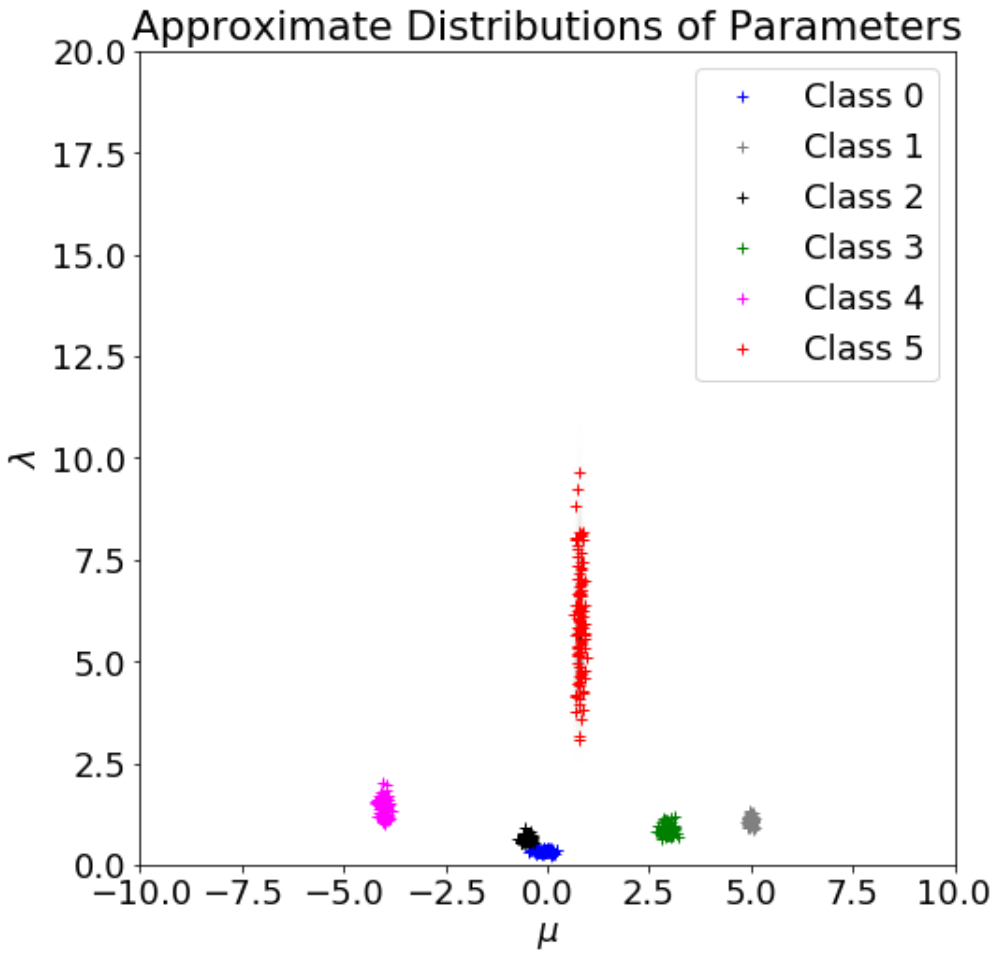
I regenerated the data using the same process, but with N=1000 for the training set; everything else is left the same as in the previous experiments.

Question 1

Looking at the plot now, there is a much more concentrated distribution for all but the fifth component (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 GMM.

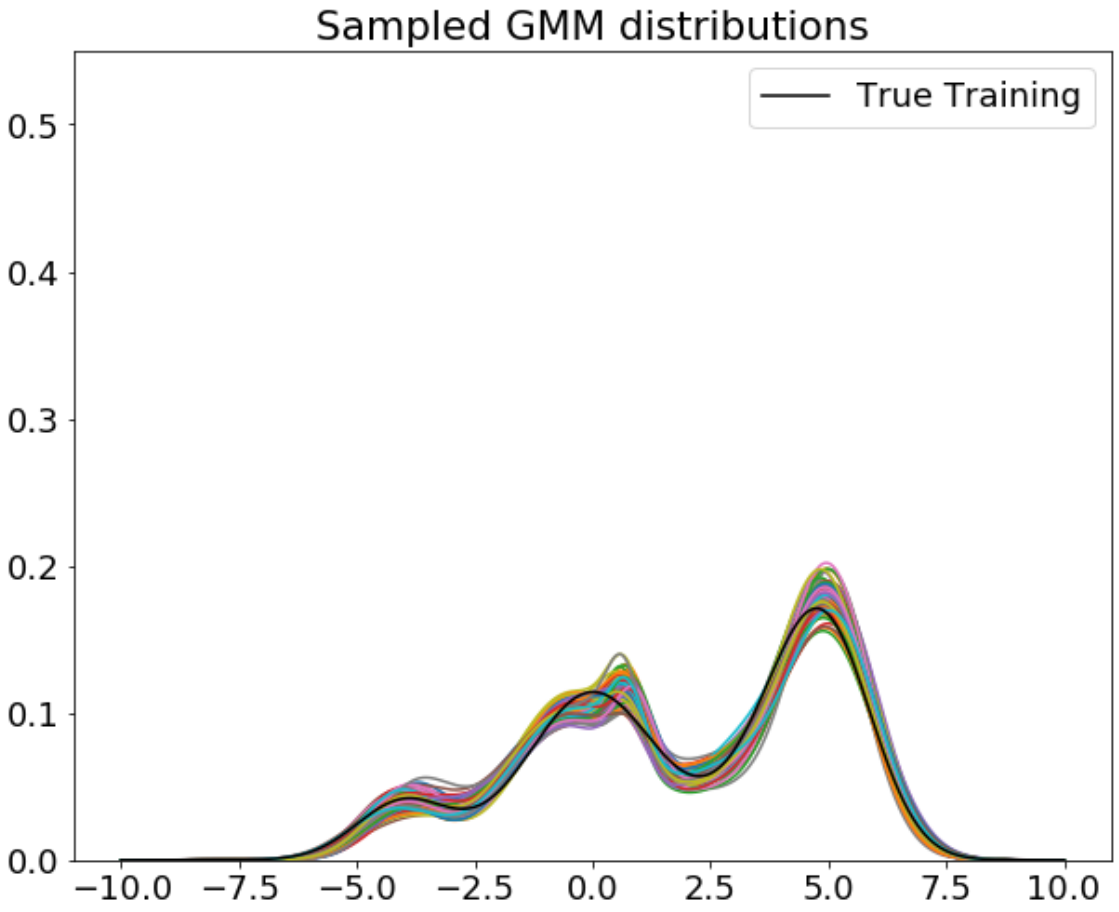
Notice as well that components 0 and 2 have parameters that are really close together; this means that they will be "collapsed" into almost the same distribution. Concretely, the algorithm learned to "erase" the fifth component and "merge" together components 0 and 2.

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



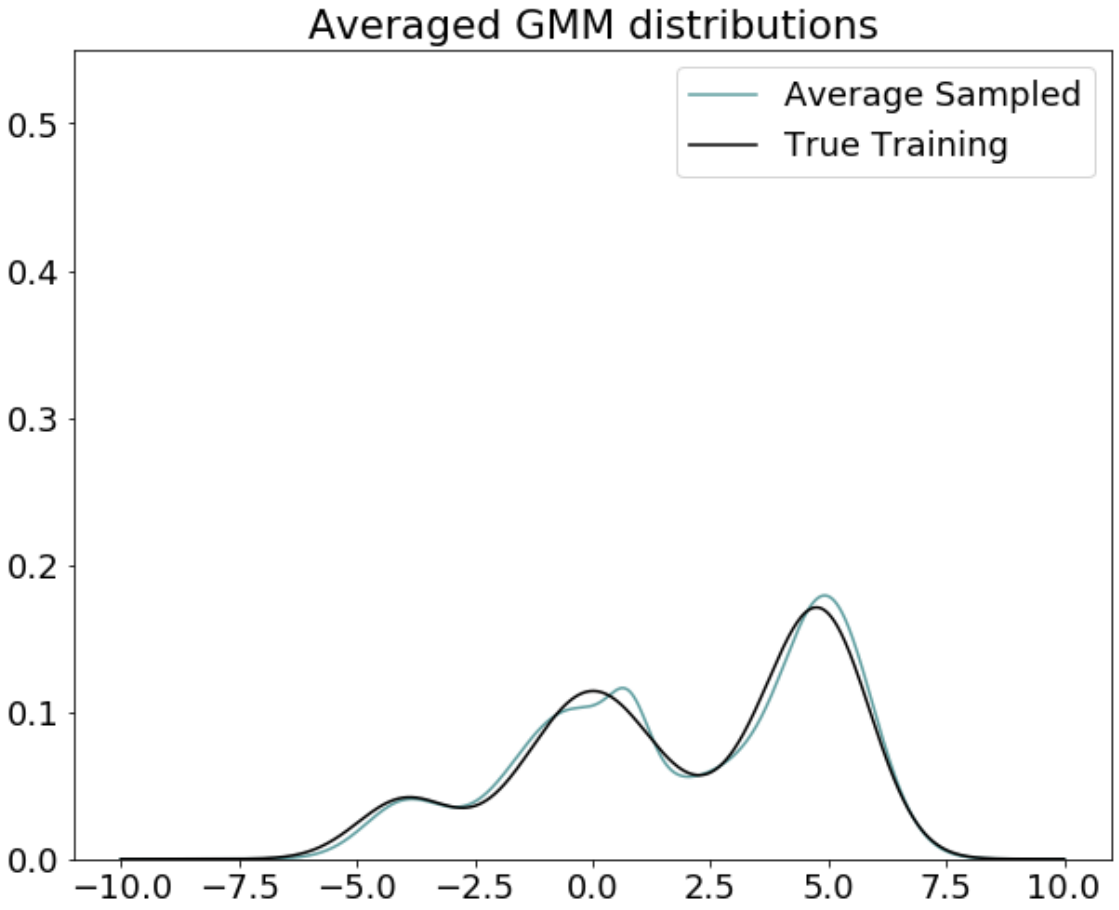
Question 2

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.



Question 3

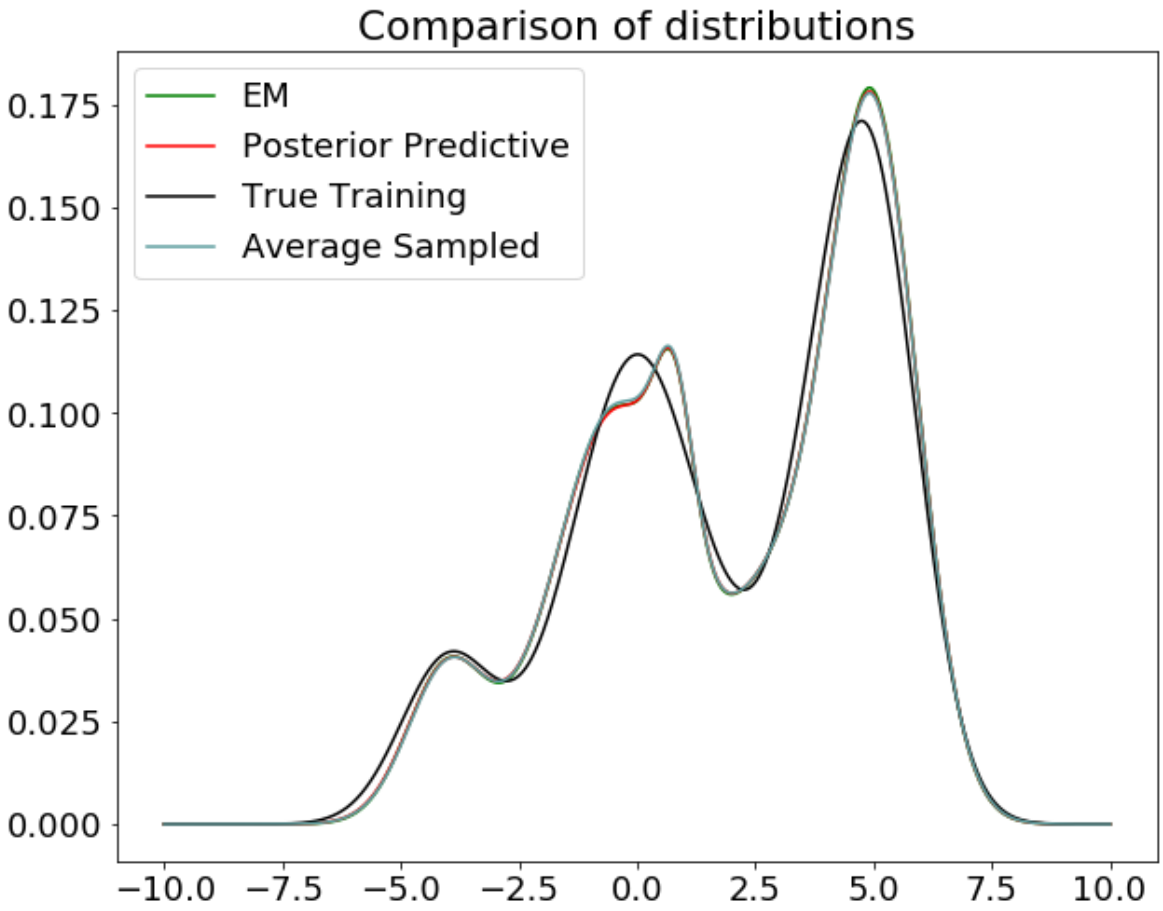
Observe that the average sampled distribution again corresponds in shape to what one would expect from the previous plot, as mentioned in the answer to question 3 in the previous section. However, it is now much better fit to what the training distribution looks like.



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.

Notice 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.