Lambda impact

The objective function of QGMM is $NLL+\lambda*C$. In the research of the validity of the objective function, we set the lambda to 1. In this research, we'll look into the impact of lambda in the objective function by changing the value.

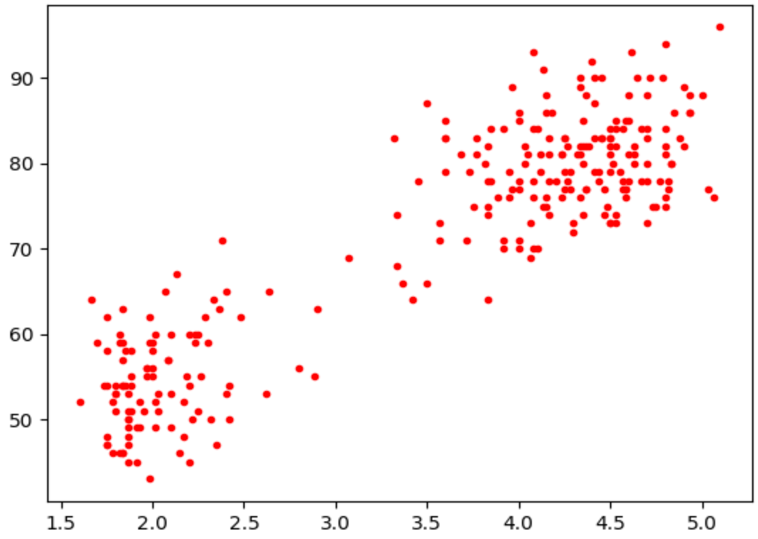
Aim

For cases that were tricky or hard in the validity research, we'll re-train the parameters with the more constrained optimization by adjusting the lambda.

Dataset

Dataset: *Data/Dataset/faithful.csv*

Like the validity research, we'll use the 'Old faithful' data set.



[Fig 0. Old faithful]

Parameters and Data

These are the records of the parameters in each experiment, so each experiment can be reproduced later.

- **Optimizer**
  Adam

- **Common variables (t1 ~ t2)**

  learning rate = 0.01

  alphas = [0.5, 0.5]

  ```
  covs1 = [ [0.08, 0.1],
            [0.1,   3.3] ]

  covs2 = [ [0.08, 0.1],
            [0.1,   3.3] ]
  ```

- **t1** - Good
  lambda = 53
  mean1 = [2.756031811312966, 76.62447648112042]
  mean2 = [2.9226572802266397, 88.3509418943818]

- **t2** - Good
  lambda = 139
  mean1 = [4.171021823127277, 83.66322004888708]
  mean2 = [1.781079954983019, 95.411542531776]

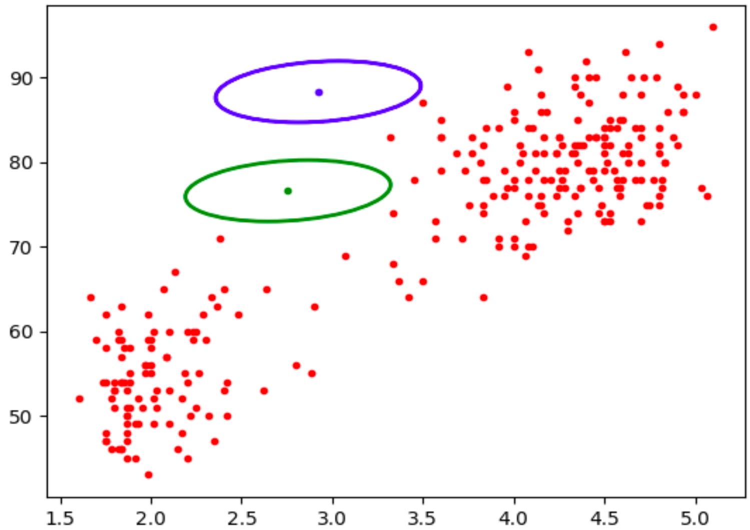
All the data that were used in this experiment are in this directories.

1. Videos: *Data/Videos/Lambda impact*
2. CSVs: *Data/Csvs/Lambda impact*
3. Images: *Data/Images/Lambda impact*
4. Graphs: *Data/Graphs/Lambda impact*

Results

[t1]

A (lambda = 1)

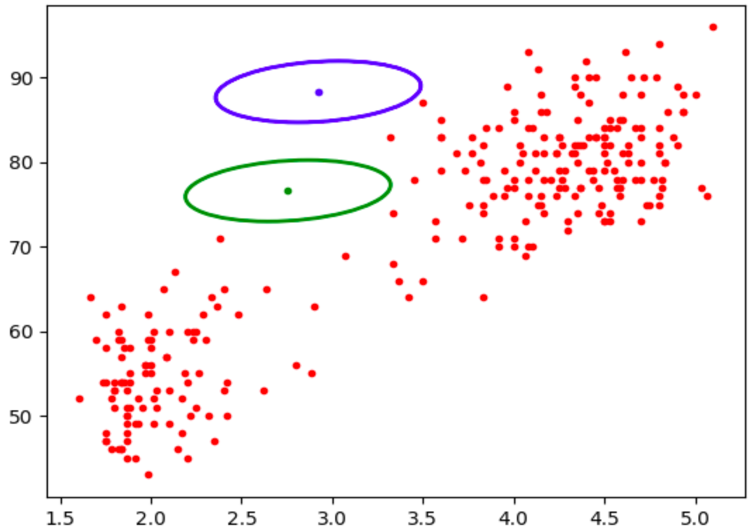


[Fig 1-1. Initial state]

[Green, Blue]

- **Alphas**
  : 0.5, 0.5
- **Unnormalized Gaussians**
  : 0.5354648, 0.26814005
- **Cosine**
  : 1828.4346
- **Sum of probabilities**
  : 0.51283795

B (lambda = 53)

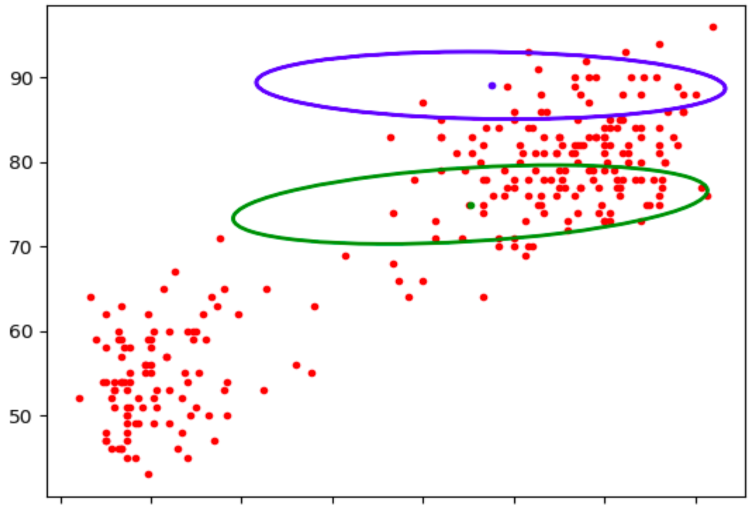


[Fig 1-2. Initial state]

- **Alphas**
  : 0.5, 0.5
- **Unnormalized Gaussians**
  : 0.5354648, 0.26814005
- **Cosine**
  : 1828.4346
- **Sum of probabilities**
  : 0.51283795

The initial states of the A and B were the same, and it seems that there is no problem yet. Seeing that the unnormalized Gaussians of A and B, the green one has more observations than the blue one.

A (lambda = 1)

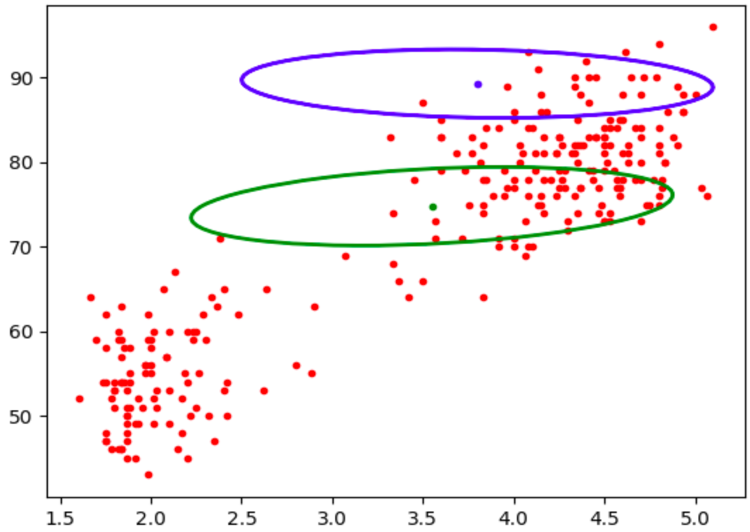


[Fig 1-3. Iteration 200]

[Green, Blue]

- **Alphas**
  : 0.75765216, 0.4059952
- **Unnormalized Gaussians**
  : 17.75705, 7.7367334
- **Cosine**
  : 16.371122
- **Sum of probabilities**
  : 2.5415406

B (lambda = 53)

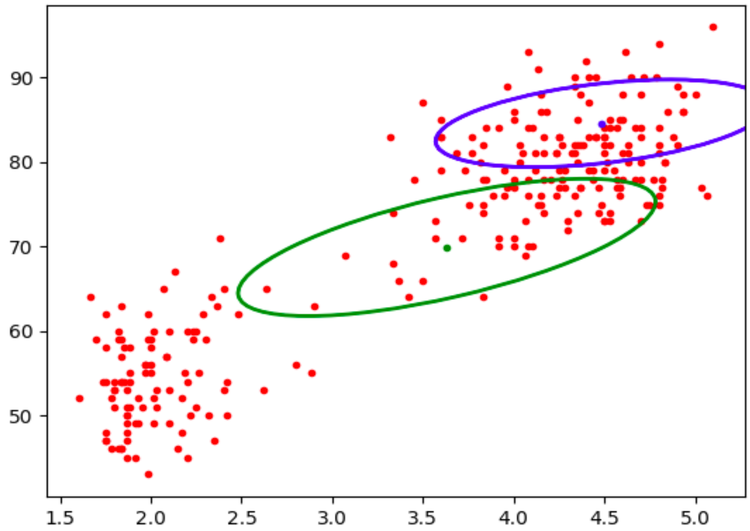


[Fig 1-4. Iteration 200]

- **Alphas**
  : 0.65228236, 0.41737202
- **Unnormalized Gaussians**
  : 15.466022, 7.2078676
- **Cosine**
  : 36.638184
- **Sum of probabilities**
  : 1.7987173

It can be seen that B has relatively less variation. We can guess it is due to higher lambda because the only parameter difference of them is it.

A (lambda = 1)

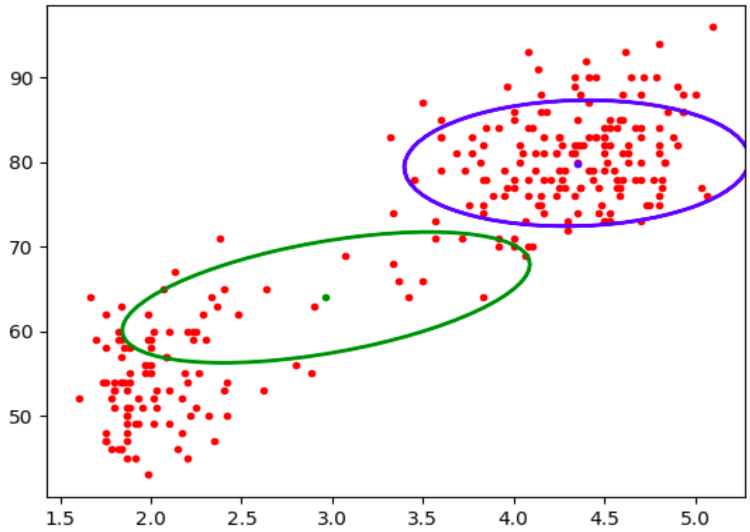


[Fig 1-5. Iteration 1200]

[Green, Blue]

- **Alphas**
  : 7.010574, 4.4088573
- **Unnormalized Gaussians**
  : 15.568036,22.909916
- **Cosine**
  : -1.8566898
- **Sum of probabilities**
  : 156.8657

B (lambda = 53)

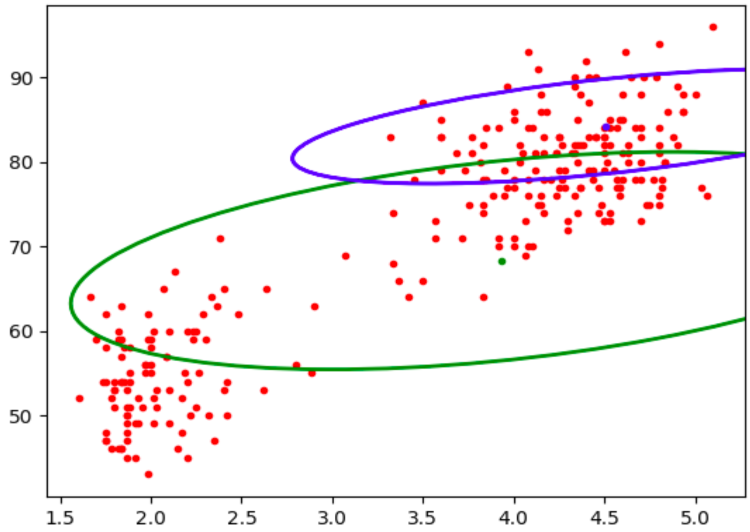


[Fig 1-6. Iteration 1200]

- **Alphas**
  : 0.9592279, 0.6766921
- **Unnormalized Gaussians**
  : 10.033731, 30.683979
- **Cosine**
  : -0.97292846
- **Sum of probabilities**
  : 3.8674023

At iteration 1200, the A and B were different in the alphas and the sum of probabilities mainly. The alphas and the sum of probabilities of A were very big, while the alphas and the sum of probabilities of B were moderate. I think it was because of the constrained optimization.

A (lambda = 1)

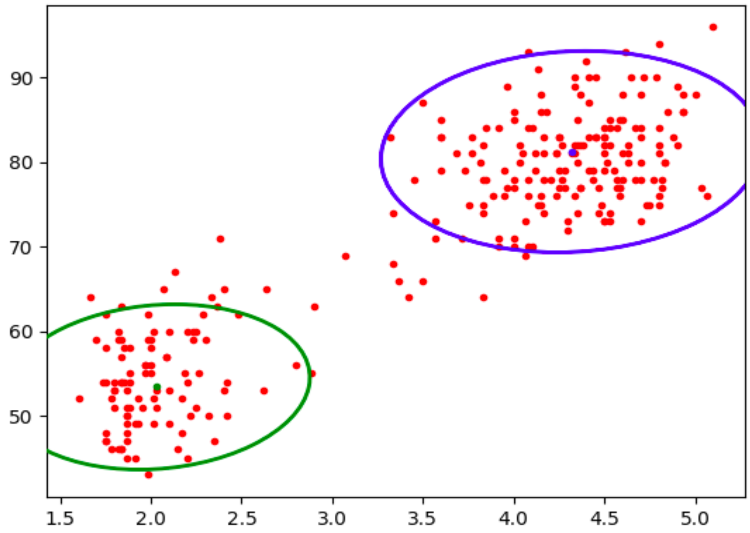


[Fig 1-7. Iteration 6000]

[Green, Blue]

- **Alphas**
  : 18.965605, 10.960408
- **Unnormalized Gaussians**
  : 15.73049, 20.505314
- **Cosine**
  : -1.2735692
- **Sum of probabilities**
  : 449.18265

B (lambda = 53)



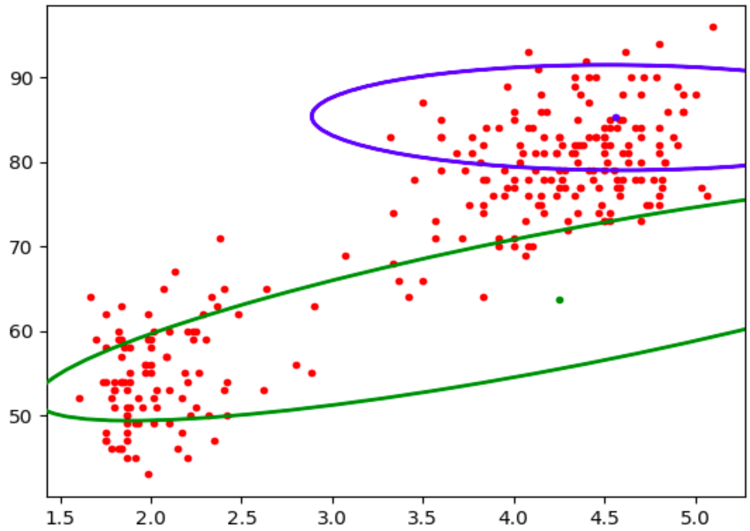
[Fig 1-8. Iteration 6000]

- **Alphas**
  : 0.5514179, 0.7076884
- **Unnormalized Gaussians**
  : 18.865803, 28.529137
- **Cosine**
  : 86.465195
- **Sum of probabilities**
  : 3.9826922

This is very interesting. When using lambda 1, we couldn't train the parameters properly, however, with the constrained optimization by increasing the lambda moderately, the train was going to be well.

Main differences between them were the alphas and the sum of probabilities. While A's alpha and sum of probabilities diverged, B's doesn't diverged with constrained.

A (lambda = 1)

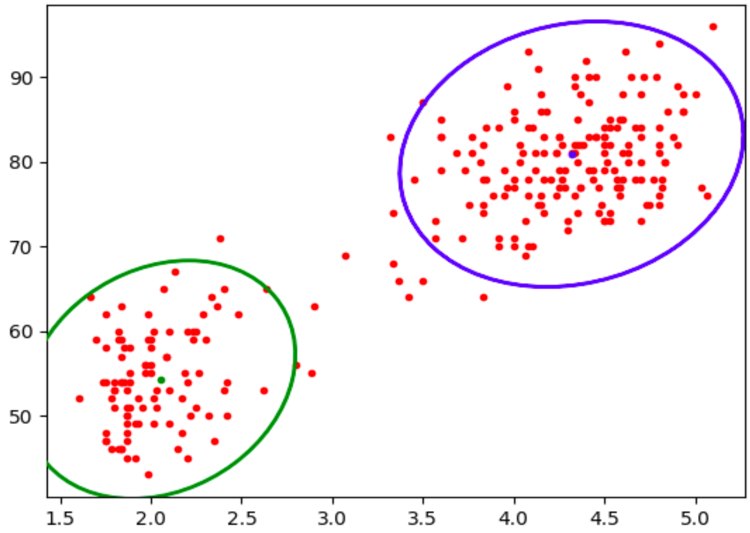


[Fig 1-9. Iteration 19900]

[Green, Blue]

- **Alphas**
  : 58.56975,25.336065
- **Unnormalized Gaussians**
  : 8.825818,18.543152
- **Cosine**
  : -26.35208
- **Sum of probabilities**
  : 4.638092

B (lambda = 53)



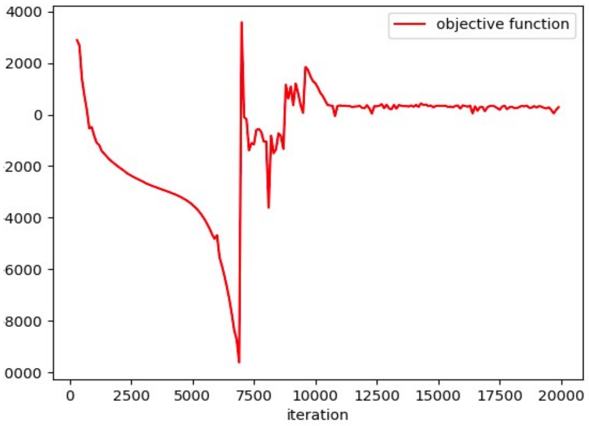
[Fig 1-10. Iteration 13700]

- **Alphas**
  : 0.57815975, 0.73033047
- **Unnormalized Gaussians**
  : 18.775711, 27.678883
- **Cosine**
  : 18.59792
- **Sum of probabilities**
  : 3.8863723

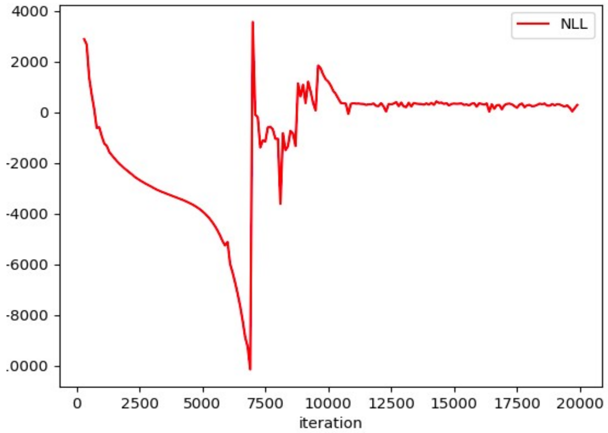
The training result of them was different totally. The alphas of A was so big, but the B's was moderate.

Besides, until the training finished, the parameters of A were changed sharply, but the parameters of B were changed smoothly.

A (lambda = 1)

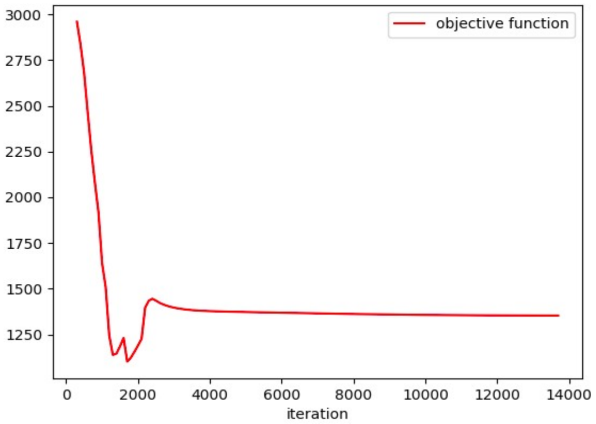


[Fig 1-11. Objective function]

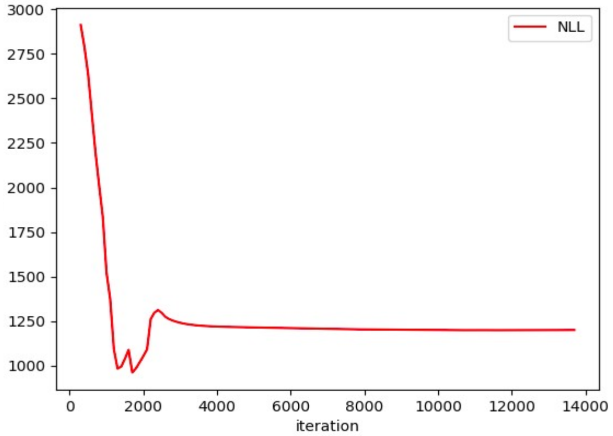


[Fig 1-12. Negative log-likelihood]

B (lambda = 53)



[Fig 1-13. Objective function]

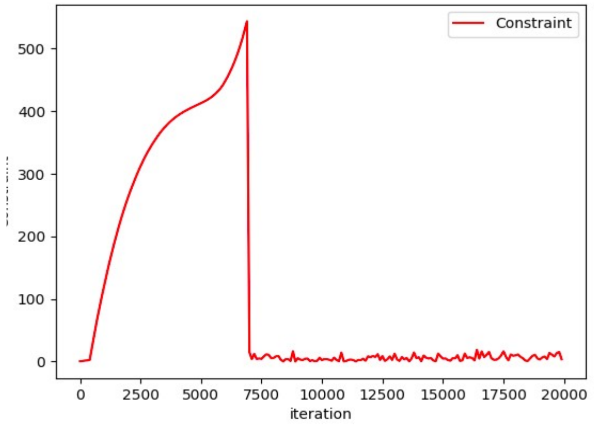


[Fig 1-14. Negative log-likelihood]

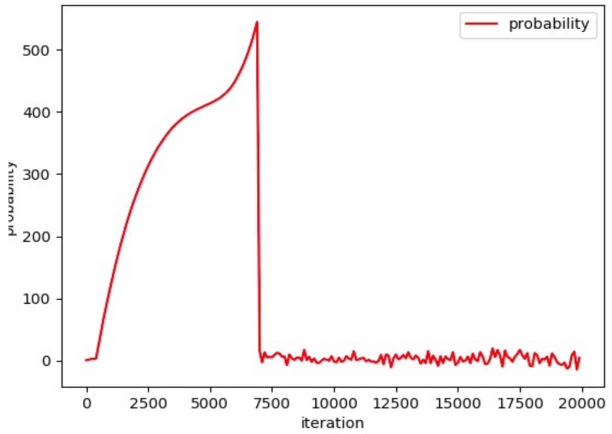
The objective function and NLL of A were so jagged, but them in B showed smooth changes except for the iterations between 1500 ~ 2500.

We can figure out that the training process of B was more stable than A.

A (lambda = 1)

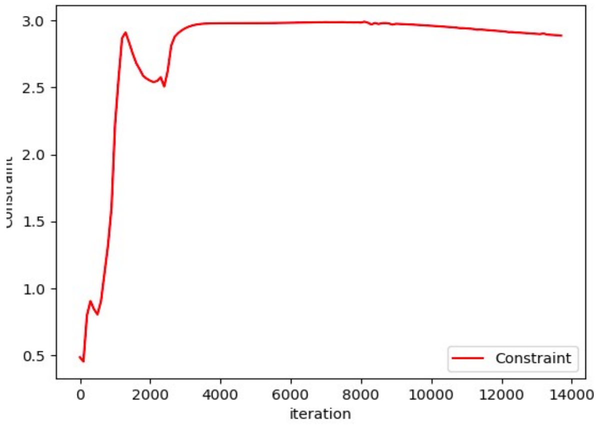


[Fig 1-15. Constraint]

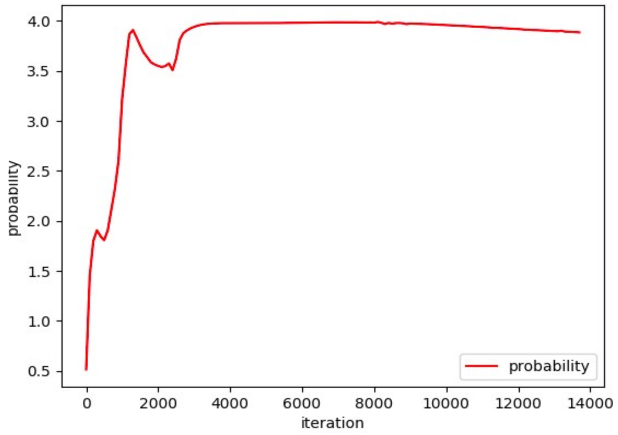


[Fig 1-12. Sum of probabilities]

B (lambda = 53)



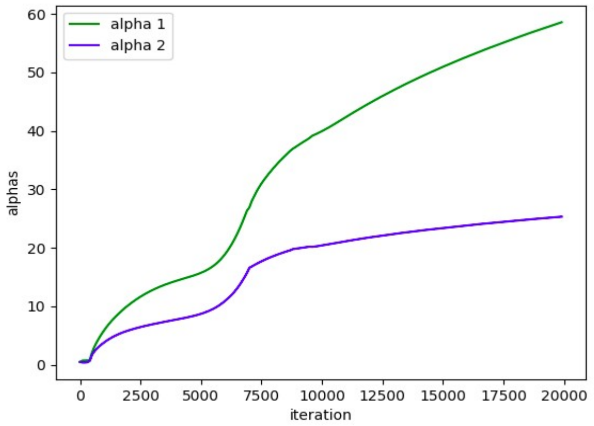
[Fig 1-13. Constraint]



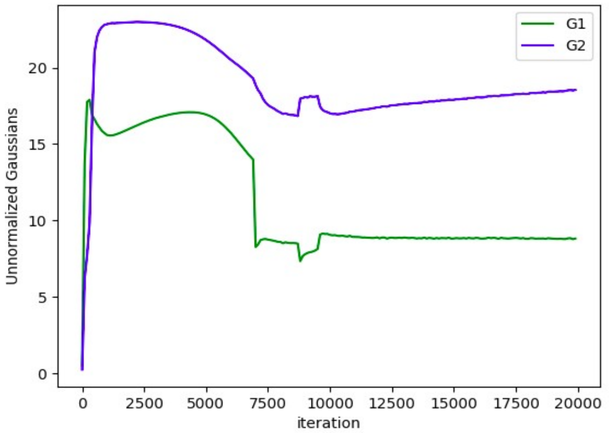
[Fig 1-14. Sum of probabilities]

In A, the constraint and the sum of probabilities were dropped into zero during the training. However, in B, although them seemed to be unstable in early stages, they became stable.

A (lambda = 1)

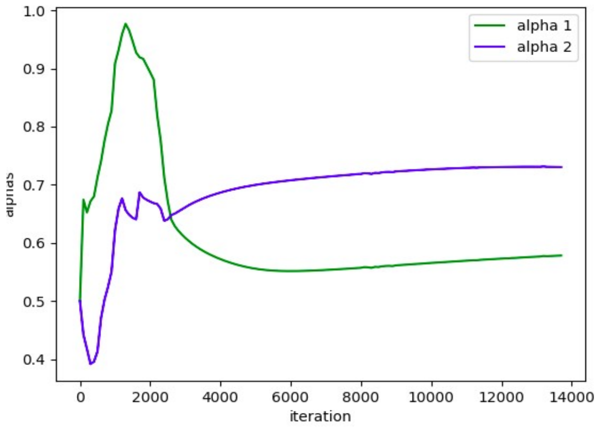


[Fig 1-15. Alphas]

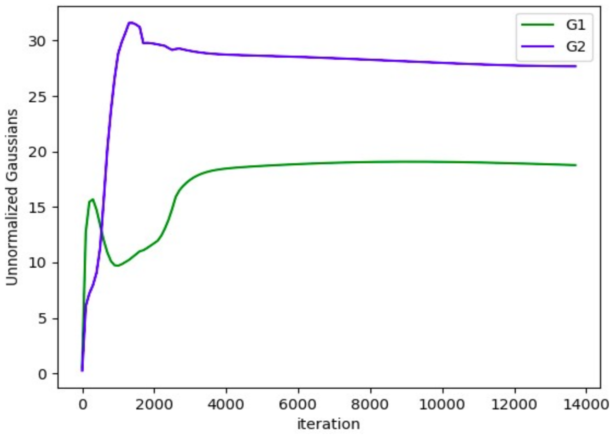


[Fig 1-16. Unnormalized Gaussians]

B (lambda = 53)



[Fig 1-17. Alphas]

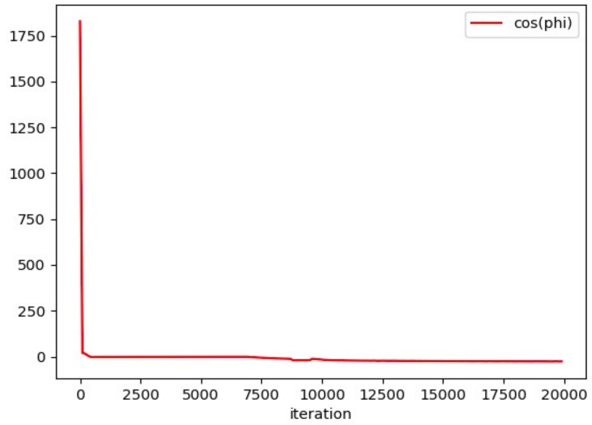


[Fig 1-18. Unnormalized Gaussians]

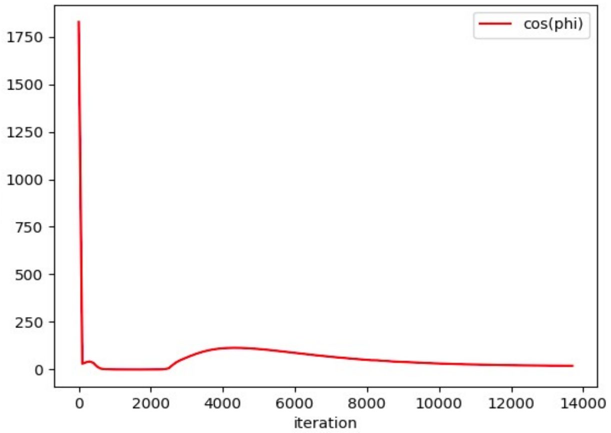
In A, the alphas have been increased continuously and it means that them weren't constrained. Besides, the unnormalized Gaussians of A were twisted in the middle of training.

In B, while the alphas and the sum of probabilities were twisted, they became gentle.

A (lambda = 1)	B (lambda = 53)



[Fig 1-19. Cosine]

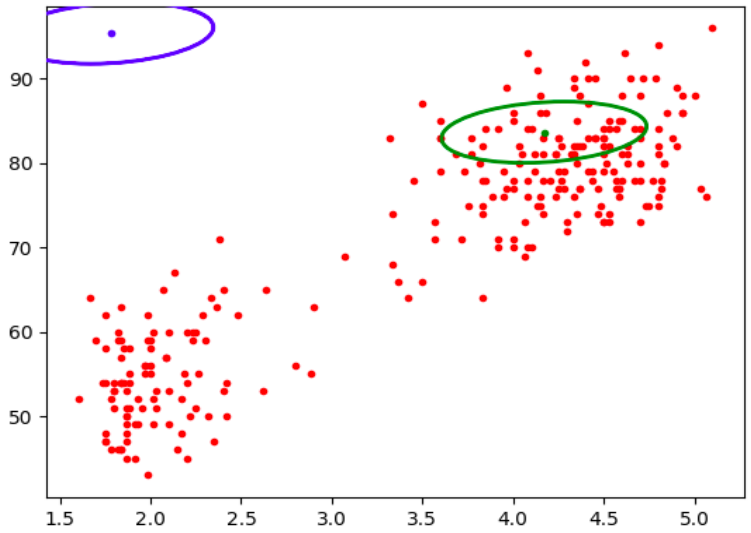


[Fig 1-20. Cosine]

The cosine of A converged to 305 and the cosine of B converged to 18.5. Especially, in B, we can figure out that the alphas or the unnormalized Gaussians were changed sharply in the middle of the training.

[t2]

A (lambda = 1)

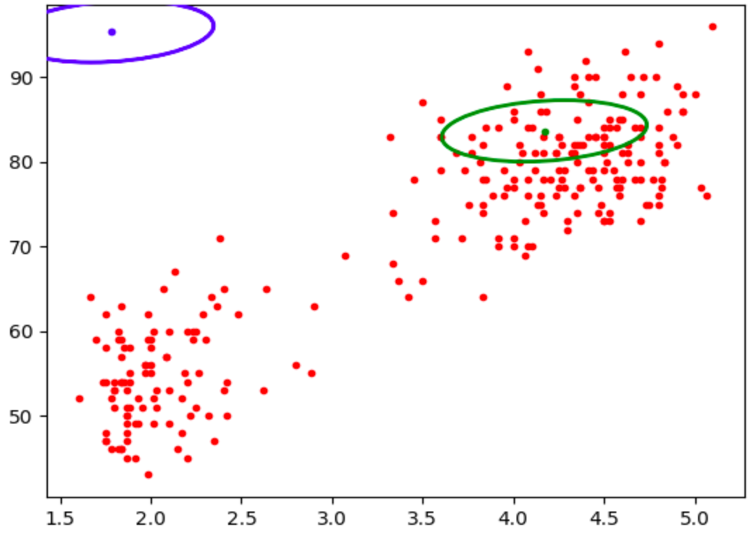


[Fig 2-1. Initial state]

[Green, Blue]

- **Alphas**
  : 0.5, 0.5
- **Unnormalized Gaussians**
  : 21.59848, 1.3045009e-08
- **Cosine**
  : 3122437400.0
- **Sum of probabilities**
  : 2.2268722

B (lambda = 139)

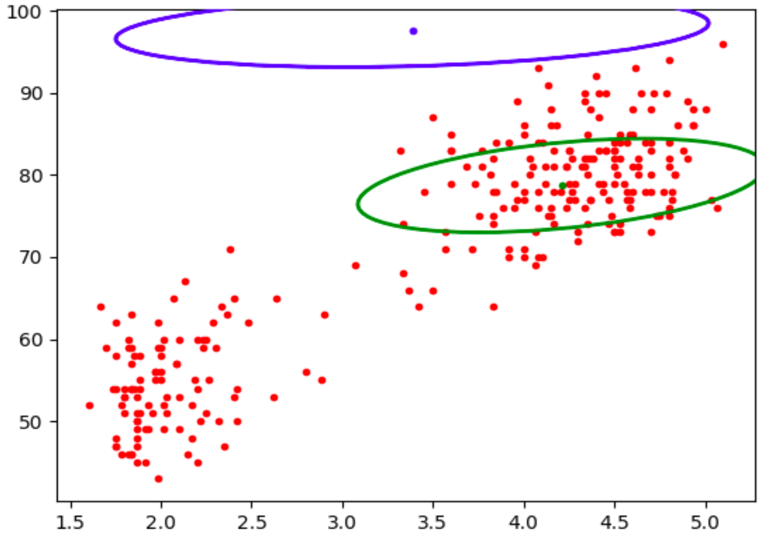


[Fig 2-2. Initial state]

- **Alphas**
  : 0.5, 0.5
- **Unnormalized Gaussians**
  : 21.59848, 1.3045009e-08
- **Cosine**
  : 3122437400.0
- **Sum of probabilities**
  : 2.2268722

The initial state of A and B was the same. Seeing that the alphas and the unnormalized Gaussians, the two clusters' scale was the same and the green cluster had more observations.

A (lambda = 1)

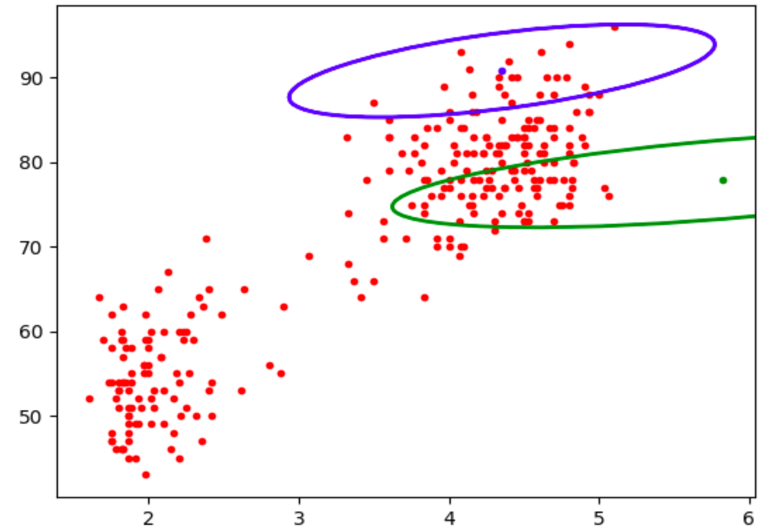


[Fig 2-3. Iteration 800]

[Green, Blue]

- **Alphas**
  : 0.92923874, 4.1735002e-05
- **Unnormalized Gaussians**
  : 27.723278, 0.31121016
- **Cosine**
  : 4713739.5
- **Sum of probabilities**
  : 5.6276226

B (lambda = 139)

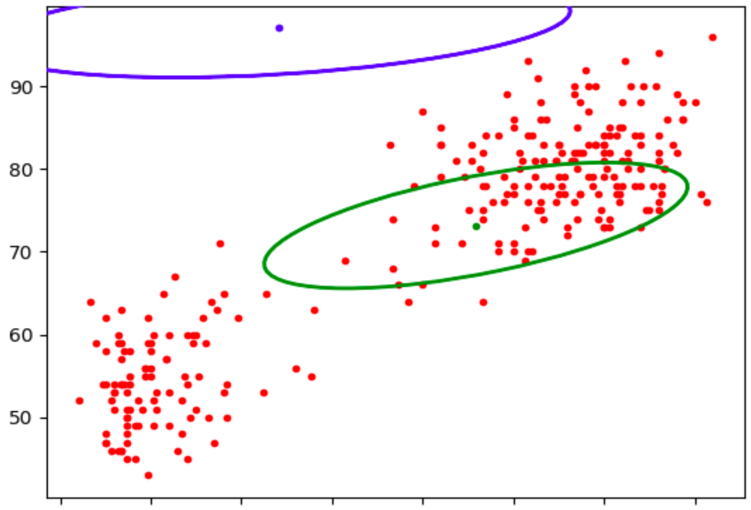


[Fig 2-4. Iteration 800]

- **Alphas**
  : 0.691048, 0.3179245
- **Unnormalized Gaussians**
  : 11.07081, 6.80551
- **Cosine**
  : 80.32282
- **Sum of probabilities**
  : 1.2066886

The two states A, B were totally different. While the alpha of the blue cluster in A was almost zero, B's alphas were balanced. Besides, the overall rate of change for the parameters in B was more moderate than A.

A (lambda = 1)

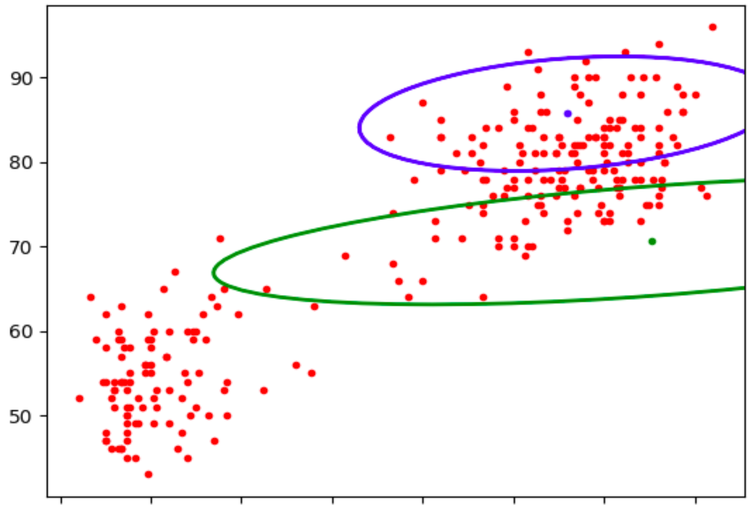


[Fig 2-5. Iteration 1500]

[Green, Blue]

- **Alphas**
  : 0.88845915, -0.0003968925
- **Unnormalized Gaussians**
  : 20.208536, 0.18553834
- **Cosine**
  : -1926247.2
- **Sum of probabilities**
  : 2.991683

B (lambda = 139)

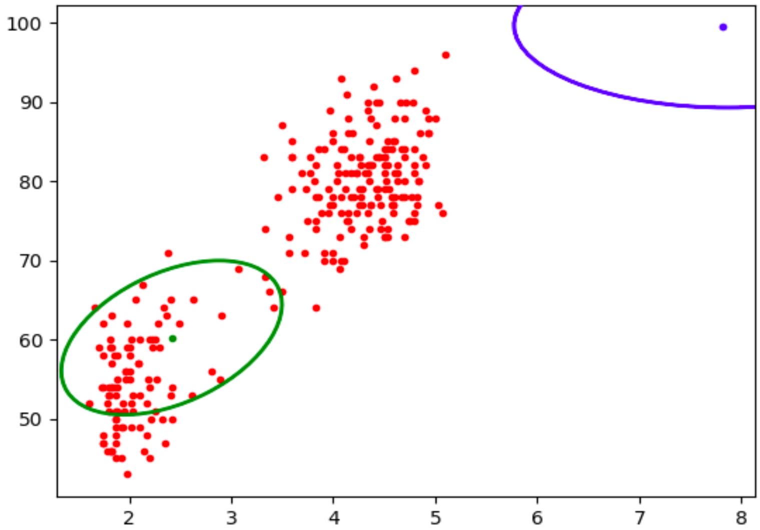


[Fig 2-6. Iteration 1500]

- **Alphas**
  : 0.84679234, 0.3822817
- **Unnormalized Gaussians**
  : 8.978367, 20.939293
- **Cosine**
  : 0.9488628
- **Sum of probabilities**
  : 1.3841966

We can figure out that the blue cluster of A keeps vanishing. In contrast, the blue cluster of B keeps growing.

A (lambda = 1)

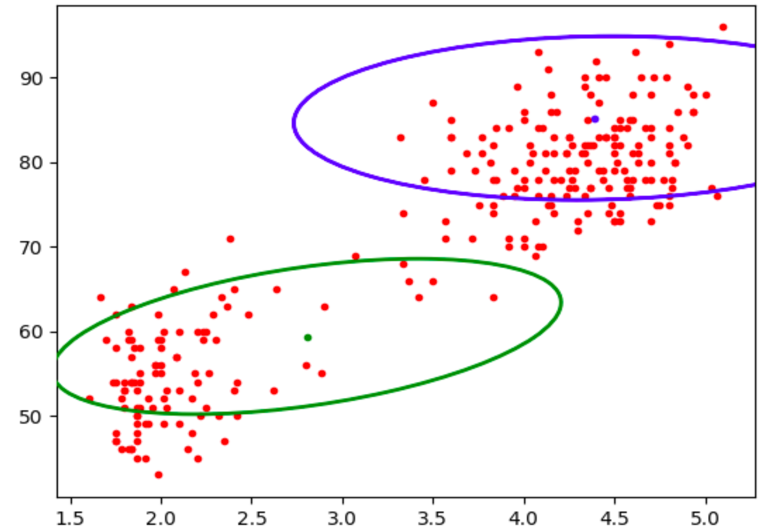


[Fig 2-7. Iteration 3300]

[Green, Blue]

- **Alphas**
  : 0.62676644, 0.00012729265
- **Unnormalized Gaussians**
  : 14.612187, 0.19712558
- **Cosine**
  : 114669470.0
- **Sum of probabilities**
  : 1.5372882

B (lambda = 139)

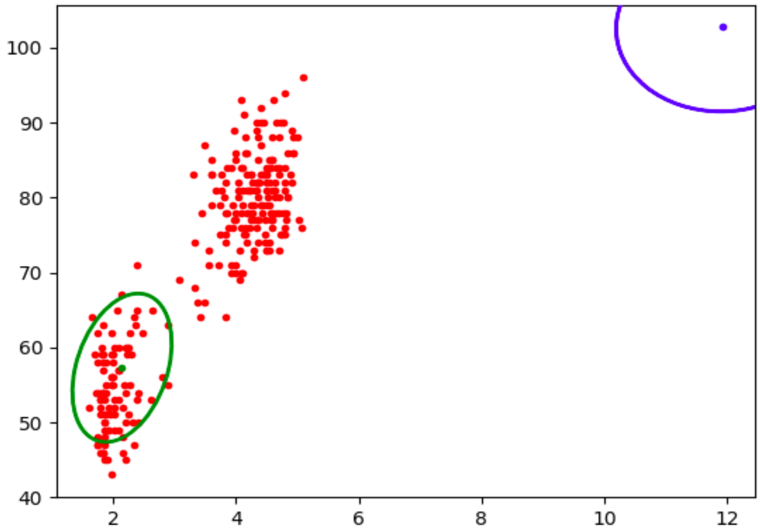


[Fig 2-8. Iteration 3300]

- **Alphas**
  : 0.6328515, 0.48733166
- **Unnormalized Gaussians**
  : 12.64649, 21.22799
- **Cosine**
  : 11.1616745
- **Sum of probabilities**
  : 1.7508601

At iteration 3300, we can see their difference obviously. The training process in A didn't work well because of the vanishing blue cluster. In B, the two clusters were balanced in the training.

A (lambda = 1)

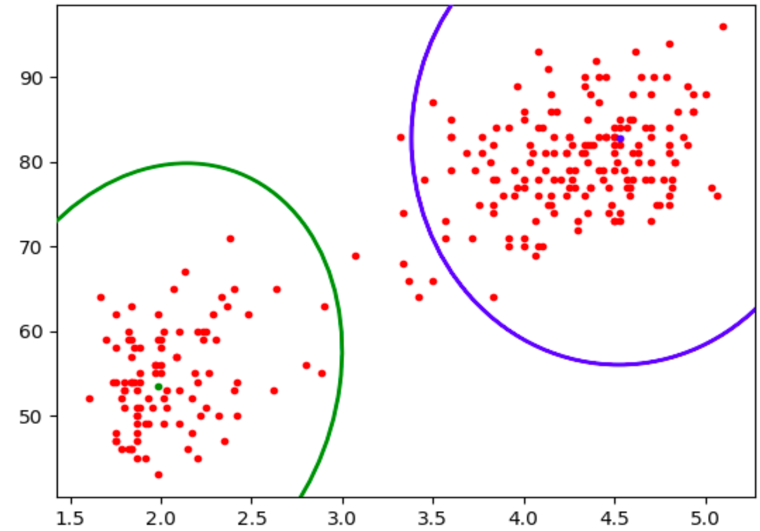


[Fig 2-9. Iteration 3900]

[Green, Blue]

- **Alphas**
  : 0.6055952, 0.00057936425
- **Unnormalized Gaussians**
  : 18.111656,7.9755985e-08
- **Cosine**
  : 1.945357e+16
- **Sum of probabilities**
  : 2.0234442

B (lambda = 139)

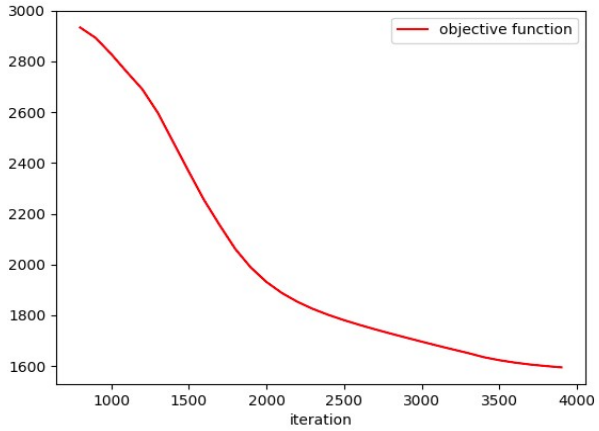


[Fig 2-10. Iteration 34600]

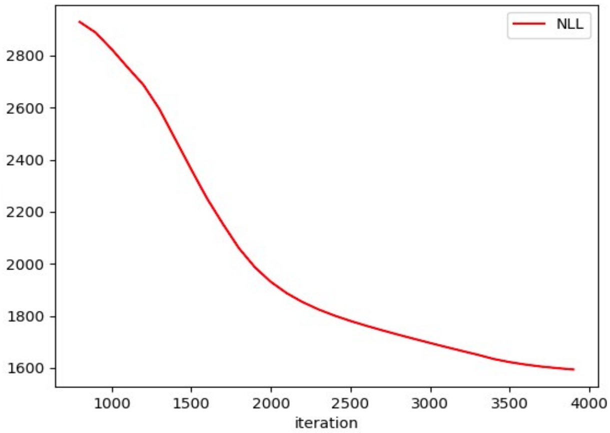
- **Alphas**
  : 0.55829144, 0.6204959
- **Unnormalized Gaussians**
  : 13.870943, 20.844496
- **Cosine**
  : 12.624651
- **Sum of probabilities**
  : 1.8947967

These are the last states of them. As a result, B has much better performance in the training than A. Because of the vanishing blue cluster of A, the cosine became very large. On the other hand, the rate of changes for the parameters in B was moderate by virtue of the constrained optimization.

A (lambda = 1)

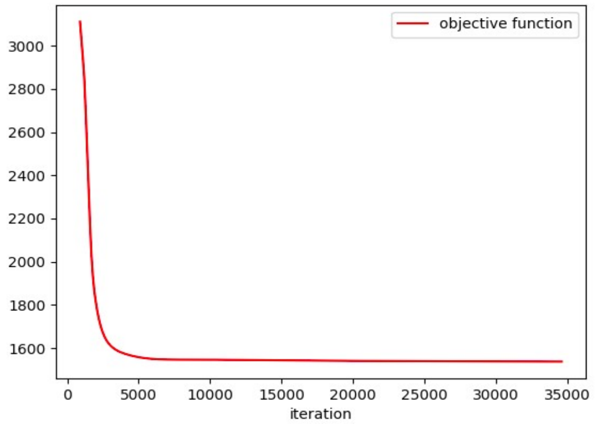


[Fig 2-11. Objective function]

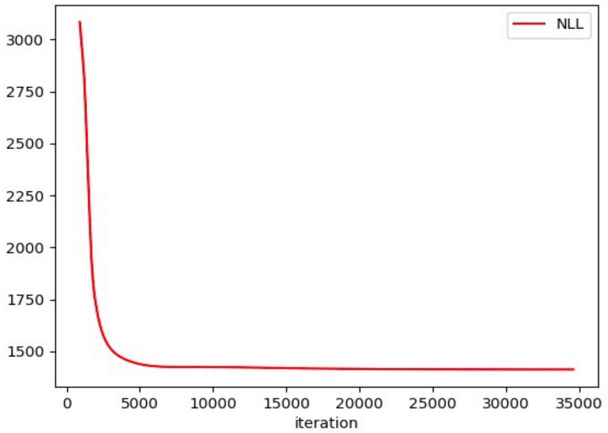


[Fig 2-12. Negative log-likelihood]

B (lambda = 139)



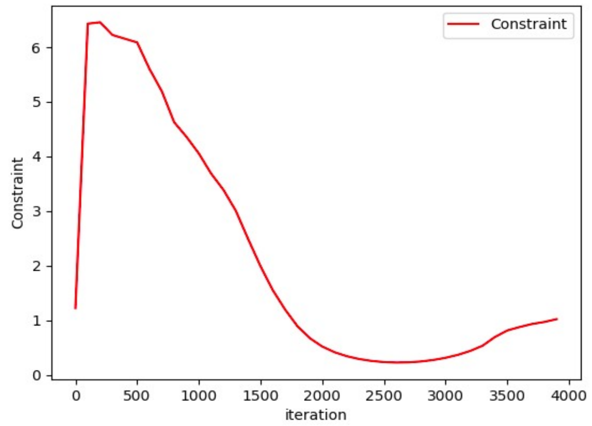
[Fig 2-13. Objective function]



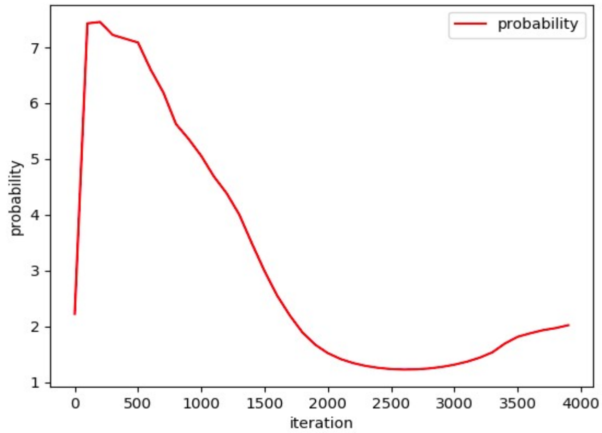
[Fig 2-14. Negative log-likelihood]

Although their values decreased all, the shapes of them were totally different. The value of A has been decreased continuously. On the other hand, the initial value of B has been decreased sharply and it became gentle.

A (lambda = 1)

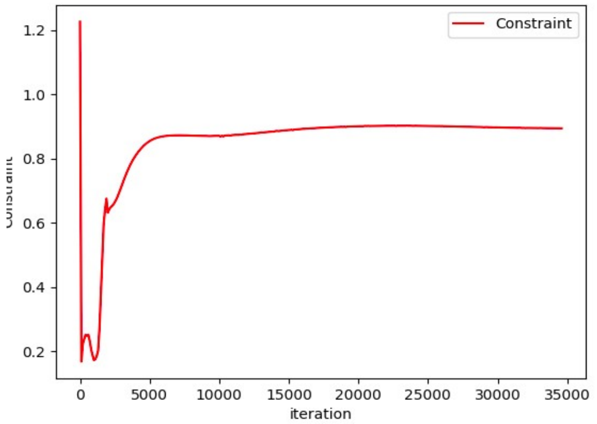


[Fig 2-15. Constraint]

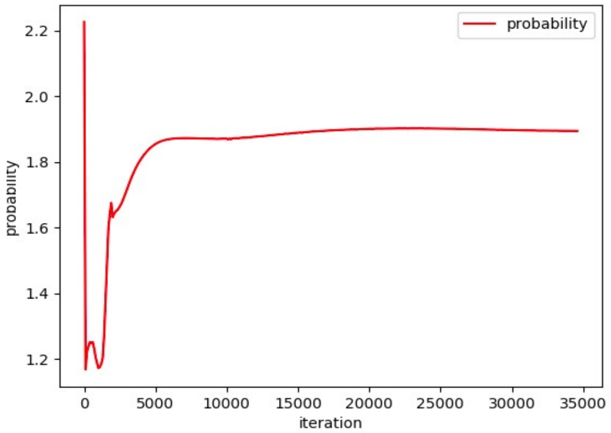


[Fig 2-12. Sum of probabilities]

B (lambda = 139)



[Fig 2-13. Constraint]

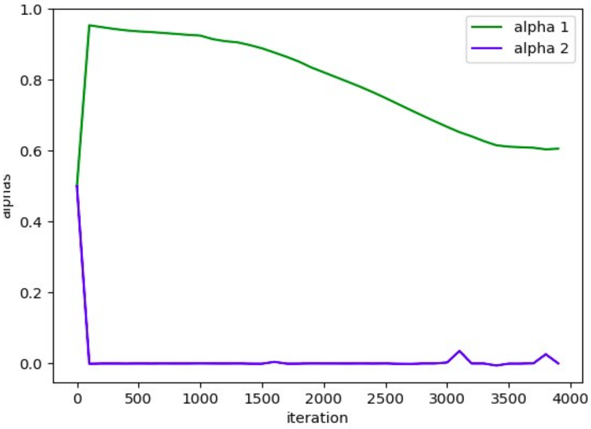


[Fig 2-14. Sum of probabilities]

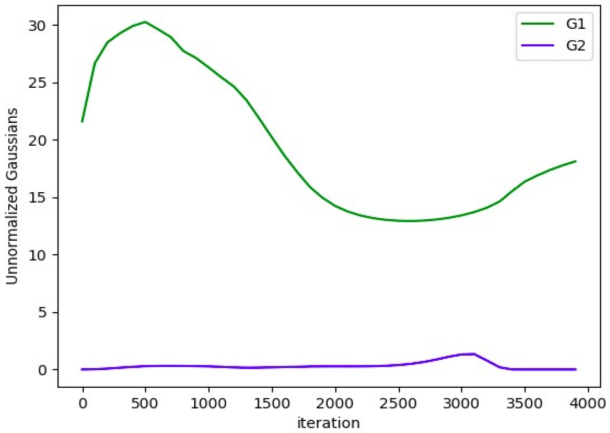
In A, seeing that the rate of change was so big, the training was unstable on the whole.

In B, in the early stages, the training process was very unstable, however, it became stable soon.

A (lambda = 1)

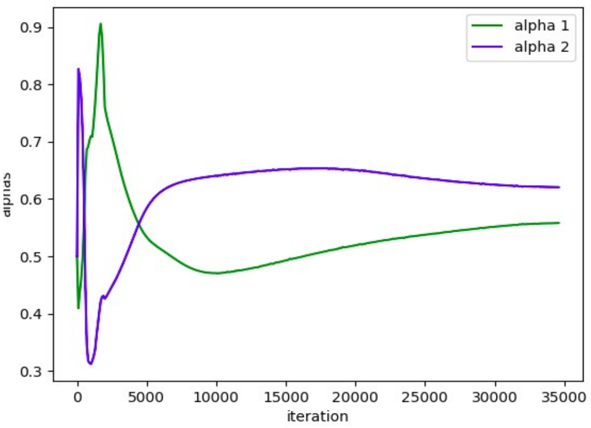


[Fig 2-15. Alphas]

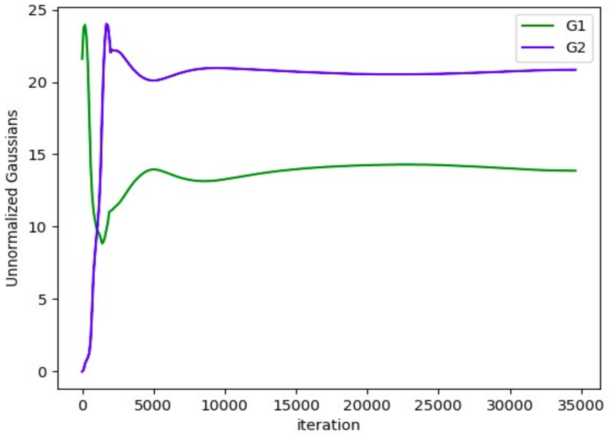


[Fig 2-16. Unnormalized Gaussians]

B (lambda = 139)

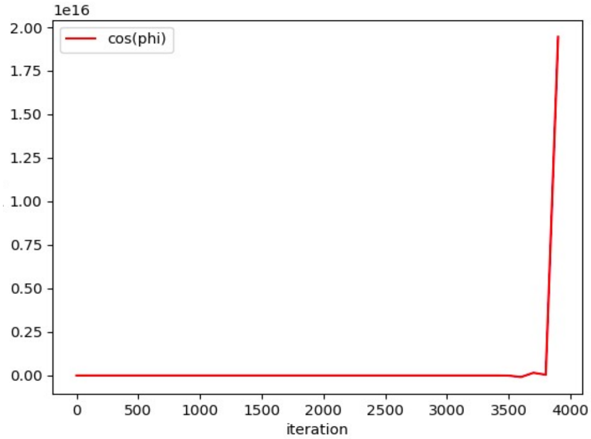
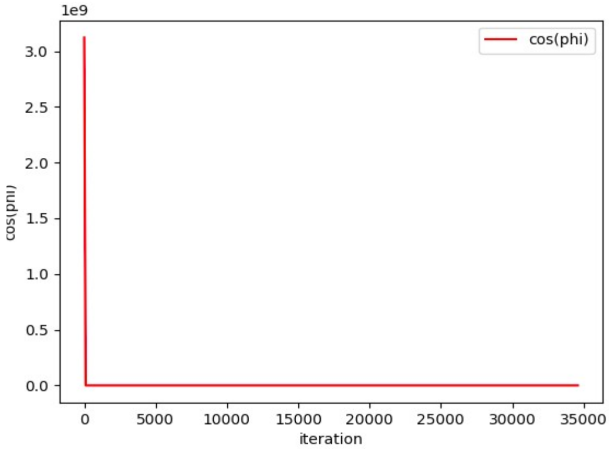


[Fig 2-17. Alphas]



[Fig 2-18. Unnormalized Gaussians]

The alphas and the unnormalized Gaussian of the blue in A keep decreasing to zero. On the other hand, B's were more stable relatively although there was something to be twisted the graphs in the early stages.

A (lambda = 1)	B (lambda = 139)
	
[Fig 2-19. Cosine]	[Fig 2-20. Cosine]

While the cosine of A diverged, B's converged to 16. Therefore, we can see that the training process of B was more stable than A.

Conclusions

We figured out the impact of lambda in this research and I think it was valuable to see the possibility to increase the performance of training.

In the validity research, the experiments of t3 and t5 had bad results. However, we trained them with constrained optimization by adjusting the value of lambda, and the results became better. Owing to the more constrained optimization, we observed the training process that was unstable in the validity research became more stable.

Consequently, in this research, we checked that we can make our training more constrained and it is possible to get better results in cases that the training wasn't performed well with the unconstrained optimization.