Report Week 8

Matthijs Neutelings, Dirren van Vlijmen, Olivier Brahma

April 3, 2022

1 Exact computation

For the first exercise, we implement a Boltzmann machine (BM) algorithm for small problems using exact computations. For a BM the basic idea is to treat the Boltzmann-Gibbs distribution of the Ising model as a statistical model (eq. 1 and eq 2). Where $p(s|w, \theta)$ represents the probability of a state s (or pattern) given w and θ. For this exercise, we implemented a BM which has no hidden units and is thus not restricted.

$$p(s|w, \theta) = \frac{1}{Z} \exp\left(\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} s_i s_j + \sum_{i=1}^{n} \theta_i s_i\right) \tag{1}$$

$$Z = \sum_{s} \exp\left(\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} s_i s_j + \sum_{i=1}^{n} \theta_i s_i\right) \tag{2}$$

Given a set of P training patterns $s^\mu = (s_1^\mu, ..., s_n^\mu)$ with $\mu = 1, ..., P$, the likelihood is defined in equation 3. The goal of this exercise is to find both a w and θ, that maximizes this (log) likelihood.

$$L(w, \theta) = \frac{1}{P} \sum_{\mu} logp(s_1^\mu, ..., s_n^\mu | w, \theta) \tag{3}$$

The BM uses a gradient descent like optimization with step-size η, where the ultimate goal is to have a convergence at the point where: $\langle s_i \rangle_c = \langle s_i \rangle$ and $\langle s_i s_j \rangle_c = \langle s_i s_j \rangle$. Thus the difference between the two is used as gradient. The clamped statistics (eq. 4, 5) being computed from the data as follows.

$$\langle s_i \rangle_c = \frac{1}{P} \sum_{\mu} s_i^\mu \tag{4}$$

$$\langle s_i s_j \rangle_c = \frac{1}{P} \sum_{\mu} s_i^\mu s_j^\mu \tag{5}$$

The free statistics (without the small c) are the well known expectation values given the distribution of eq 1.
Which gives the update per iteration as depicted within eqs 6 to 9. It should be noted that learning requires evaluation of the free statistics for each iteration, which can be intractable for bigger models. Further it was empirically found, that without ample data (at least needing ± 1000 samples), convergence was near impossible. To combat that even the toy model became slow, the use of a specific einsum function from the NumPy library improved iteration speed significantly. For the ultimate implementation, we refer to the GitHub repository [1].

$$w_{ij}^{(t)} := w_{ij}^{(t-1)} + \Delta w_{ij} \tag{6}$$

$$\Delta w_{ij} = \eta(\langle s_i s_j \rangle_c - \langle s_i s_j \rangle) \tag{7}$$

$$\theta_i^{(t)} := \theta_i^{(t-1)} + \Delta \theta_i \tag{8}$$

$$\Delta \theta_i = \eta(\langle s_i \rangle_c - \langle s_i \rangle) \tag{9}$$

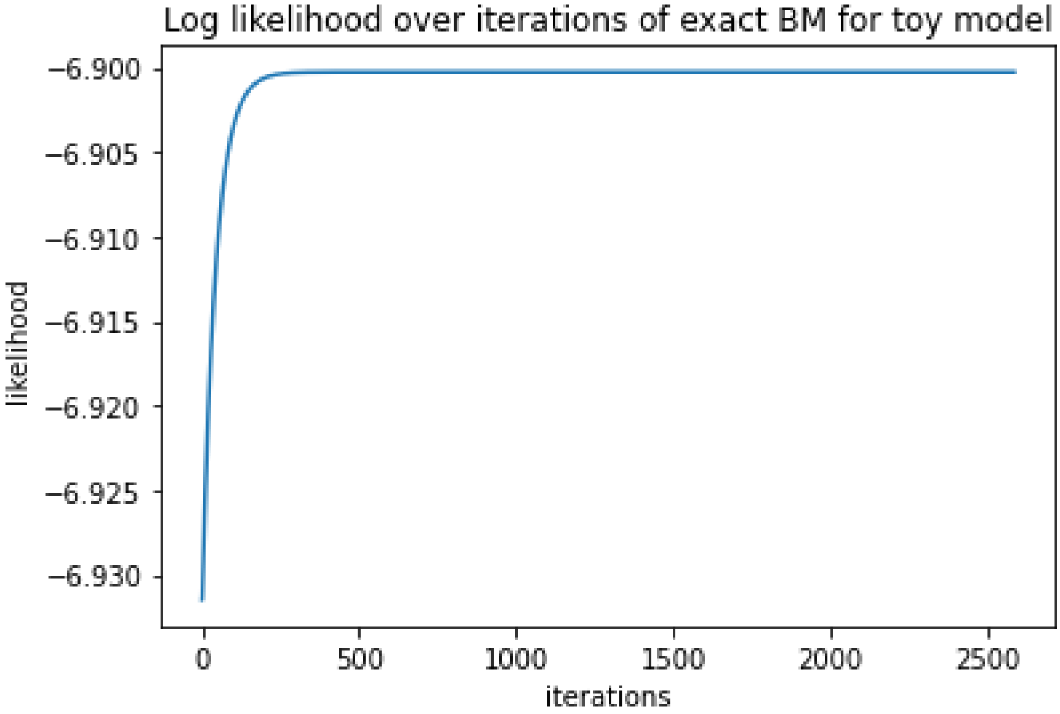


Figure 1: When applied to the toy model with a learning rate of 10^{-3}, the BM algorithm converges after 2500 iterations. The likelihood does stagnate before hitting 500 iterations. Indicating that the convergence criterion might be too strict or the optimal learning rate/momentum values of the algorithm should be investigated to limit this stagnation. Note that the above plot already starts quite high, most likely since we opted to start with a weight matrix and bias vector that are zero. Since we can assume that for the toy model the spins do not have a true underlying correlation between them this will be close to the "true" w and θ.

1.1 Momentum

Another trick we applied to speed up convergence was the implementation of momentum β (eq. 10, 11) as also mentioned within Kappen and Rodrıguez (1997) [2]. Using the gradients of the previous iteration Δw_{ij}^p and $\Delta \theta_i^p$ to keep some inertia, the algorithm did converge notably faster than without it. We decided to set $\beta = 0.01$ for this exercise through trial and error.

$$w_{ij} := w_{ij} + \Delta w_{ij} + \beta \Delta w_{ij}^p \tag{10}$$

$$\theta_i := \theta_i + \Delta \theta_i + \beta \Delta \theta_i^p \tag{11}$$

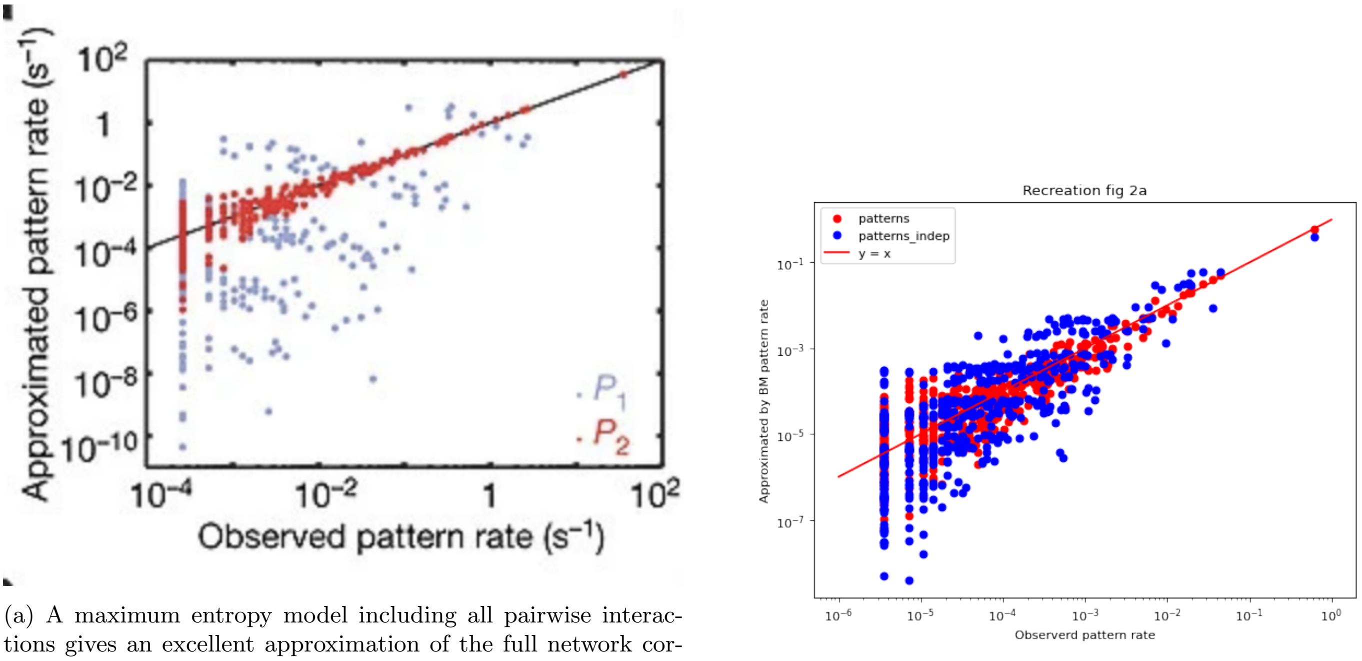
1.2 BM toy model results

The data consists of a "toy problem", which generates random data with 10 spins. The computation for a small model like this can be done exactly. The data is sampled using 1000 randomized vectors containing values $[-1, 1]$ with equal probability.

We demonstrate the convergence of the BM learning rule in figure 1. The convergence criterion is set to 1×10^{-13} for the mean gradients of both $\langle s_i \rangle$ and $\langle s_i s_j \rangle$. The figure shows, that as the number of learning iterations increases, so does the likelihood. This increase stagnates pretty fast and after 2500 iterations has converged.

1.3 Salamander Retina Data

The algorithm is tested on the salamander retina data [Schneidman et al,. 2006] [3]. This dataset shows that weak correlations between pairs of neurons coexist with strongly collective behaviour in the responses of ten or more neurons for salamander retina. It applies to these models, since the maximum entropy models are equivalent to Ising models, and predict that larger networks are completely dominated by correlation effects. Unlike the "toy problem", this dataset is disproportionately distributed with values $[0, 1]$. Here, the 0 values are replaced by -1 values similarly to the generated data. The data consists of 160 neurons with 297 repeated experiments each with 953 time-points. Resulting in a dimension of 160 x 283041 for the data. For training the BM only one of the repeats was used for a



(a) A maximum entropy model including all pairwise interactions gives an excellent approximation of the full network correlation structure [Schneidman et al., 2006] [3]. The rate of occurrence of each firing pattern predicted from the maximum entropy model P2 that takes into account all pairwise correlations is plotted against the measured rate (red dots). The rates of commonly occurring patterns are predicted with better than 10% accuracy, and the scatter between predictions and observations is confined largely to rare events for which the measurement of rates is itself uncertain. For comparison, the independent model P1 is also plotted (grey dots). The black line shows equality.

(b) Recreation of correlated samples against the independent samples. Source can be found in the referenced code. Notice that our implementation is not percentage based, as the maximum approximation rate equals 1.

Figure 2: Figure matching with previous work

dimension of 10 x 953. The goal of this exercise was to reproduce figure 2a.

1.4 BM Salamander results

Using the exact algorithm again, we now apply it to the Salamander Retina data, while attempting to reproduce the figure from Schneidman et al. We noticed that the performance is very dependent on which 10 neurons we selected, likely due to the relatively lower amount of ones compared to the toy model. We decided to reduce the convergence criterion, because of this to 1×10^{-5}. This should somewhat prevent the model from overfitting the data.

The results (fig. 2b) show the recreation of correlated samples against the independent samples. The independent samples are created by re-initializing w as 0 for each step within the update process. The correlated samples show a similar pattern as the reference figure. With the approximation and the observation values staying close to one another. The independent model shows some differences from the reference. Firstly, the model did not show the clear spike patterns at several intervals, that the reference did show. We noticed that, by setting the threshold to a lower value, the spike patterns do reveal themselves more clearly (fig. 7) only much more horizontal, to explore the reasoning for this we feel to be outside of the scope of this report, since we should focus on the dependent model. Secondly, the model seems to perform better than the reference independent model. We were unable to determine how the original authors exactly created their plot, and explicitly which kind of convergence threshold they used.

2 Metropolis Hastings

The exact method requires us to calculate Z, which in turn requires us to evaluate $p*$ for every possible state. Since there are 2^n possible states, the exact algorithm has exponential complexity, and is therefore not suited for larger values of n. Thus, for large values of n, we have to find a different way to compute the free statistics. One approach would be to sample from the distribution with the current w and θ values, and compute the statistics from this data. Given enough samples, this should be an accurate estimate of the free statistics.

We will use Metropolis Hastings sampling. The proposal distribution that we use implements single spin flips.

Remember the equation for determining the acceptance probability:

$$a = \frac{p^*(x')q(x^r|x')}{p^*(x^r)q(x'|x^r)}$$

Because of our choice for q, we have that $q(x^r|x') = q(x'|x^r)$. Thus, it disappears from the acceptance equation, and we have

$$a = \frac{p^*(x')}{p^*(x^r)}.$$

Each round, we compute the statistics for our current w and θ values by sampling from the distribution. The number of samples that we use per round is a hyperparameter, where we have to make a trade-off between speed and accuracy. Furthermore, since we implement single spin flips, two samples that are drawn immediately after each other will be the same on at least $n-1$ spins. Thus, the samples will be heavily correlated. Therefore, we can choose to implement a number of sequential spin flips between each of our samples. The more flips between samples, the less correlated they will be. The number of flips between samples is another parameter.

2.1 MH results

We compared the MH gradients with the exact gradients of the toy data model in order to find the optimal number of samples needed. We distinguish between 'correlated' and 'uncorrelated' samples, which means having 0 or 10 (the length of the pattern) sequential spin flips between samples respectively. At each step, we calculated both gradients of the current w and θ values, and then updated w and θ according to the exact gradients. Results can be found in the Appendix (figure 9). We see that more samples generally lead to less variance in the MH gradients (note that the y-axis differs between graphs). Also, having uncorrelated samples is better than having correlated ones, although it takes longer because only 1 in 10 spin flips end up making it in the samples. We ultimately opted for the 250 uncorrelated samples. Uncorrelated since it created better gradients and we thus feel it to be unwise to not add it. 250 since this was the lowest value (with speed in mind) that we felt to have an okay-ish variance at convergence and followed the gradients comparably to the higher values before convergence.

2.2 Convergence criterion

Looking at the images appendix, we see that we cannot simply wait until the gradient is close to zero, because the randomness of MH sampling causes the gradients to fluctuate around 0. Thus, we propose looking at a larger number of recent samples, and say that we converge when their mean is sufficiently close to zero, since this would indicate oscillations of the gradient and thus convergence of w and θ. The number of samples to look at and the convergence threshold are hyperparameters. Another option would be to wait until the absolute values of the gradients are (semi) consistently below a threshold, but this would be difficult because the threshold would need to depend on the number of samples used for the MH sampling (since more samples gives lower fluctuation). We apply the MH sampling method on the full salamander data in section 4.

3 Mean Field approximation

A second method to approximate the free statistics without the need for a calculation of Z is the mean field and the linear response approximation. For this method the mean spins need to be approximated. These are calculated via fixed point iteration using the following formula.

$$m_i = tanh\Big(\sum_j w_{ij}m_j + \theta_i\Big) \tag{12}$$

It should be noted that in order to make this method work the initial value of this fixed point iteration should be within the possible values of m, specifically -1 and 1. These values thus already gives us the first needed free statistics, since $\langle s_i \rangle_c = m_i$. The second free statistics we need are calculated using the linear response approximation. Which for this context becomes:

$$\langle s_i s_j \rangle = \chi_{ij} + m_i m_j \tag{13}$$

Where χ_{ij} is defined as follows:

$$\chi_{ij} = A^{-1}_{ij} = (\frac{\delta_{ij}}{1-m_i^2} - w_{ij})^{-1} \tag{14}$$

For this approximation it was empirically found that the convergence method that was proposed for the exact computation can also be used for this approximation. This is most likely due to the fact that the computed gradient via this approximation is quite stable (compared with the Metropolis Hastings sampling method) and thus the change of the parameters is stable enough that it will after a while adhere to this criterion. Of course different convergence criterion's are possible, with the easiest one being the one used for the MH method. But given that the exact convergence is possible, and this one is less dependent on a bit of luck with the gradient a few iterations back being of a similar absolute value, it is felt better to use the normal convergence criterion. Further it could be questioned whether the criterion from MH would be applicable, given that the gradient fluctuates less, it might be impossible to “catch” this behaviour, or at least much harder. Which further makes the case for the use of the exact convergence criterion.

3.1 Partition approximation

The mentioned sampling methods will be used to approximately find the w, and θ that maximizes the likelihood of the data. In order to show that the process thus converges we need to calculate the likelihood per iteration and ultimately generate a plot of this likelihood versus the iteration. But, as mentioned, that would mean we have to calculate the partition value, which becomes intractable for bigger data (like the Salamander data).

Therefore we need an approximation for this value. For this we will use an approximation that is based on the mean field approximation as (also) presented within the paper written by Kappen and Rodriquez [2].

$$-F = \log Z' = \sum_i \log\left(2\cosh\left(\theta_i + W_i\right)\right) - \sum_i W_i m_i + \frac{1}{2}\sum_{i,j} w_{ij} m_i m_j \tag{15}$$

Given the usual definition of the mean field equations, as denoted by eq 12, and the definition given of m_i within the paper:

$$m_i = \tanh\left(W_i + \theta_i\right) \tag{16}$$

It is assumed that W follows the following definition:

$$W_i = \sum_j w_{ij} m_j \tag{17}$$

Which is felt to be okay, since this definition can also be seen as the interaction that neuron i has with the other neurons. Which is mentioned to be the approximate idea behind W in the paper.

4 Results for Sampling methods

Firstly the implementation of the approximation is checked for a randomly chosen smaller sub part of the salamander data. This was also done for the Metropolis Hastings method, with the goal of comparing these likelihoods with the likelihood from the exact method. In order to see whether the approximations are sufficiently correct. The plot generated can be seen within figure 3

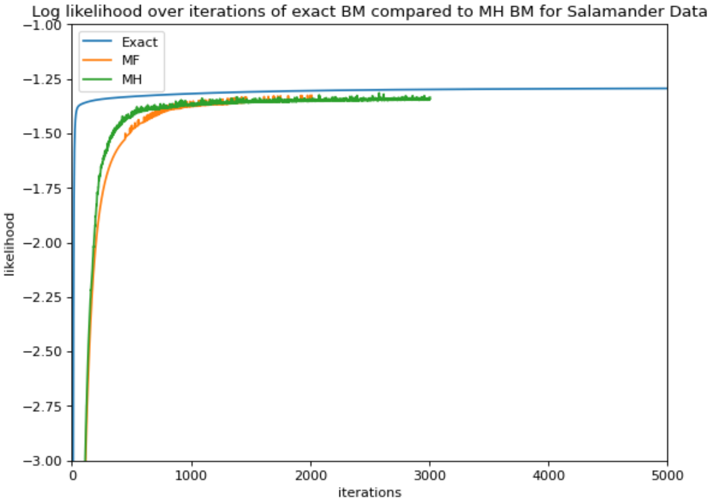
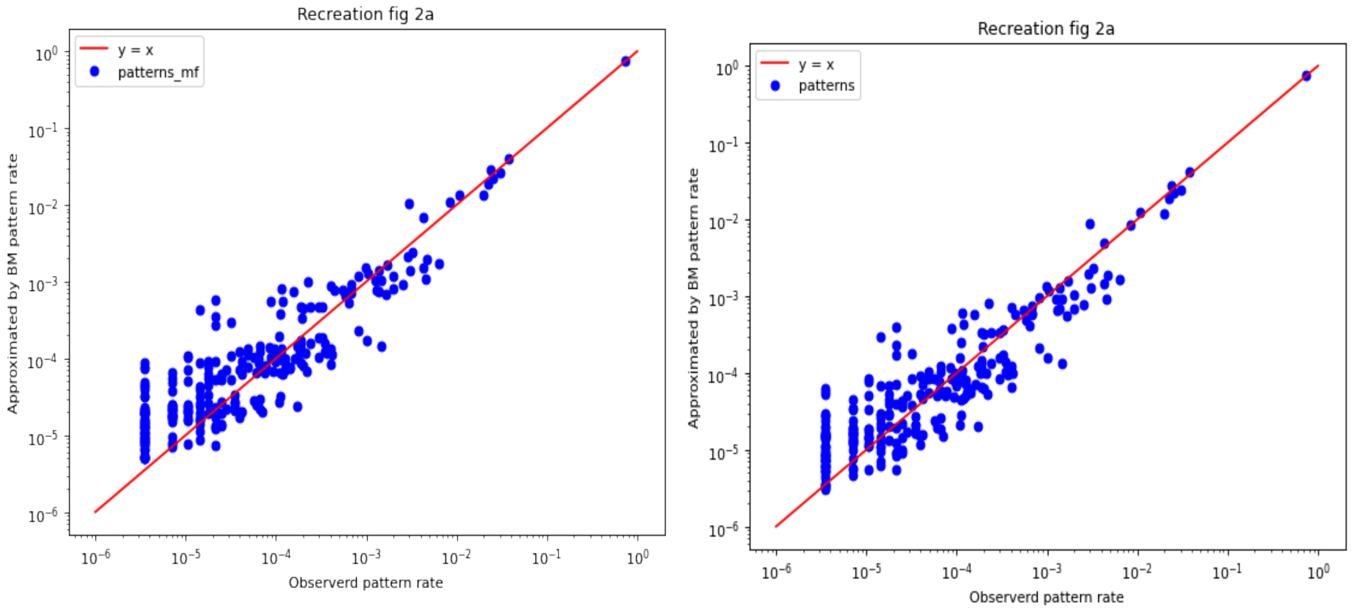


Figure 3: Figure containing the likelihood per iteration for the exact, Metropolis Hastings and Mean Field/Linear Response Boltzmann machine. All likelihoods are calculated exactly, the approximation from sec 3.1 is **not** used. For MH: 250 uncorrelated samples (tried flips are now equal to 160 between samples) were used and convergence was determined on the last 50 gradients for a threshold equal to the exact threshold of 10^{-5} times the used learning rate of 10^{-3} (also the threshold and learning rate for the MF method)

The figures shows that the likelihood for all methods converge, with a small difference in ultimate maximal value between the exact method and the approximations. But we feel this difference to be within the range of differences that can be contributed to the fact that it are approximations. Further we can note that the approximations are a bit slower within their convergence. They “lag” behind at the start. Where the exact method is almost vertical, and the approximations are somewhat angled. And the angle between the likelihood going up and starting to converge is much closer to 90 degrees for the exact method. Again we feel that all of this can be contributed to the fact that these are approximations, that will both have a harder time to start going into the right direction (which determines the "verticalness" at the start) and a harder time fine tuning at the last part. Both are due to the loss of information used within the calculation of the gradient and thus not pushing the parameters fully into the correct direction, or not as hard. The last thing of note is that the likelihood seems to oscillate a little bit for both of them. This is due to the instability of the gradient, making the parameters oscillate a little bit and thus the likelihood as well. We see that, especially at the end, that the MH method oscillates a bit more and in both directions. Where the MF method oscillates a little less and only in the upwards direction. Which seems to somewhat support the decision to not use the different convergence criterion for the MF method.

The biggest difference between the two methods is the fact that the MF method is much faster than the MH method. Where the MF method needed around 40 seconds. The MH method needed multiple minutes.

To showcase for both methods that they converged on a w and theta that indeed produce good probabilities. we reproduced the figure 2 from [3] also for the Metropolis Hastings and Mean Field/Linear Response. This resulted in the following pictures.



(a) Approximated rate (by Boltzmann machine) vs the observed rate within the data for the Mean Field/ Linear Response approximation

(b) Approximated rate (by Boltzmann machine) vs the observed rate within the data for the Metropolis Hastings approximation

Which both show very good similarity with the plot that was generated by the exact method. The two methods reproduce very similar results. The only seen difference is that the MH method puts the patterns that are observed very few times correctly a little bit lower than the MF method. Overall it is similarly seen that the MF method puts some patterns a little too high compared to the MH method. This has most likely to do with the fact that the MH method really tries to sample according to the wanted distribution, but the mean field uses the mean value of the spins. Which will be a little bit away from -1 for this data. And thus the very rarely observed patterns, that will mostly consist of patterns with a lot of ones, get a slightly too high value. Where the MH method could just almost never accept a spin flip, and thus respect the rarity of those patterns better.

4.1 Full Salamander Data

The second part of the results for both approximations is to use the implementations for the full salamander data. For this part also a plot of the likelihood is made per iteration for both methods. Of course, given the now very big nature of the data, we need to use the approximation of the partition value (as explained within section 3.1), for both methods this approximation is used. The generated plot can be found as figure 5

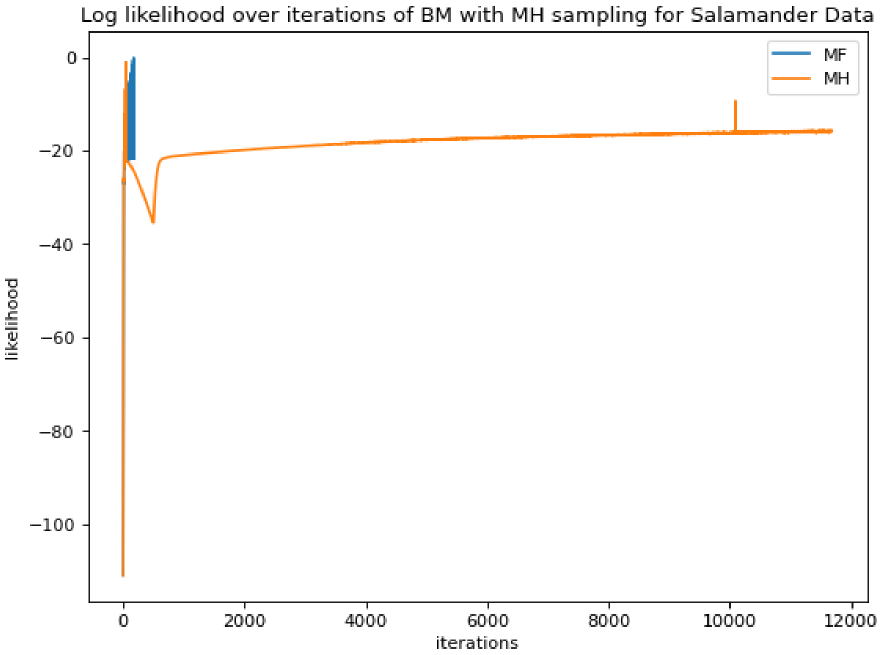


Figure 5: The log-likelihood (approximated as explained within section 3.1) per iteration for both the Metropolis Hastings (MH) and Mean Field/ Linear Response (MF) method. All hyperparameters (convergence threshold, learning rate etc.) are the same as within the 10 neuron runs.

There are a few things to be noted regarding this plot. A general note that should be made is that the approximation for the likelihood is deemed to be a bit conservative, and sometimes a bit unstable. For us this means that we will mostly be looking at the relative values, and not absolute values. First we will concern ourselves with the yellow metropolis Hastings line. This shows, after some initial weird behaviour, a slow but steady convergence of the likelihood. For us this shows that the convergence method proposed does its work. Since at the very point that the very slow rise of the line seems to stall the method is stopped by this criterion. This is also supported by the fact that even at this large y-scale, we still see the oscillations happening at the end. Which would rule out the exact convergence criterion.
For this plot the line of the Mean Field approximation seems to be very weird. Basically becoming a blob. But if we were to zoom in, as is done in figure 10 within the appendix, we see that the likelihood is jumping up and down, having a few iterations at some baseline, and then a jump of one iteration to somewhere close to zero. The most important things is that this baseline the likelihood is at the most of the time converges. And given that the exact convergence criterion is used, having one iteration at this converged baseline makes the method stop. Thus outside of this weird jumping behaviour, that we attribute to the approximation of the likelihood, the method seems to do its indented job.

If we compare the two methods we see that the MF method reaches a lower likelihood than the MH method, we mostly comes from the slow but steady increase of the MH method after ± 1500 iterations, since before that point the likelihood is very similar to that of the MH method. This shortcoming of the MF method is (more than) equalled by the shortcoming of the MH method: time. The results are shown for an MH method that samples 250 uncorrelated patterns. The method ultimately took ± 12000 iterations and took ± 2 hours on a 7th generation i7 intel CPU. The MF method only needed ± 400 iterations and took ± 40 seconds.

5 Directly solving the fixed point equations

The last exploration to find the w and θ that maximize the likelihood is to directly solve the equations from the mean field and linear response approximation. A very important note for this is that this can only be done for a Boltzmann machine without hidden units. So **not** for a restricted Boltzmann machine. The idea is that we state the clamped statistics to be equal to the free statistics. Basically setting the gradient of the Boltzmann machine directly to zero. Ultimately this leads to the following equations:

$$w_{ij} = \frac{\delta_{ij}}{1-m_i^2} - (C^{-1})_{ij} \tag{18}$$

$$\theta_i = \tanh^{-1}(m_i) - \sum_{j=1} w_{ij} m_j \tag{19}$$

Where $m_i = \langle s_i \rangle$ and is calculated through the fixed point equation from the MF approximation. Further C is defined as:

$$C_{ij} = \langle s_i s_j \rangle_c - \langle s_i \rangle_c \langle s_j \rangle_c \tag{20}$$

This method will in theory work perfectly. But there are two remarks to be made. Firstly the C matrix needs to be inverted, which might not work in some cases. And these cases include the salamander data. Therefore a small ϵ will need to be added to its diagonal.

A second remark concerns the diagonal term that is added to w. If any of the m_i is completely equal to 1 or -1 this will lead to a division by zero. Given the very large amount of -1's in the salamander data this could definitely happen. Therefore a small value of 10^{-8} is added to all components of m. This value was found by trial and error.

5.1 Likelihoods

This method is used to find w and θ for some possible values of ϵ that are added to the diagonal of C. Given that for this method the full salamander data is used, this also means that we use the approximation for the partition value in order to calculate the likelihood. This alters the results slightly, since the values themselves will no longer absolutely mean anything. We will only be looking at the relative values. Ultimately the following plot was made:

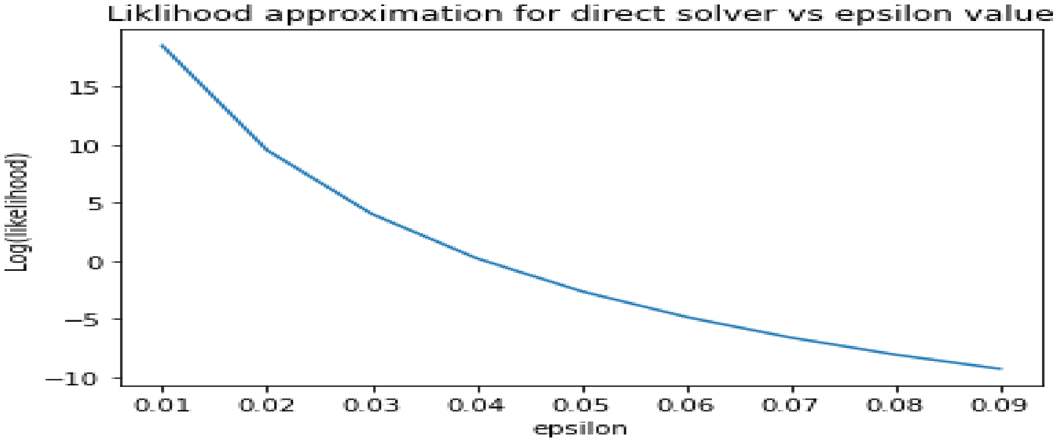


Figure 6: Likelihood values calculated via the direct method

There are some clear things to mention for this figure. The first and most clear is the fact that for some values the likelihood becomes bigger 1 (the log-likelihood higher than 0). This should never happen of course, but we feel this is most likely due to the weird approximation of the likelihood combined with the "perfectness" of the w and θ resulting far too high values. For some reason, as is seen with the spikes in the plot of the likelihoods of the full salamander data, the Z value is sometimes far too low. The Z seems to be some lower bound value, that sometimes is a bit conservative. This also has effect on the choice of epsilon values. For lower values, the Z value becomes zero. Resulting in an infinite likelihood. This happened for values lower then $8 * 10^{-3}$. And for values higher then $9 * 10^{-2}$ the likelihood became far lower and thus far less interesting. For a value of zero we indeed did not get a result, as the inverting of the C matrix became impossible.
Expectedly we see that a lower value of epsilon results in a higher likelihood. Most likely since a lower value of epsilon is a smaller deviation of the true data.

6 Summary

This report implements different techniques for applying Boltzmann machine learning. We apply an exact model, Metropolis-Hastings Sampling, mean-field + linear statistics and a direct method to both a generated toy model and the Salamander Retina data. The exact model did work, but the convergence was gradual. Implementing momentum did seem to significantly speed up the convergence. We do note that this method is impossible for larger datasets, as the current implementation only uses a limited amount of spins. The MH algorithm did work but also suffered from a lack of performance (not as bad as the exact method). The MF+LR algorithm did show better convergence speed, with a small cost to the likelihood. Finally, the direct solving method was of course much faster, but required the tuning of ϵ to make it work at all. Further all methods returned weird results due to the conservative approximation of Z. The final source code for this project is available in the References [1].

References

[1] M. Neutelings, D. van Vlijmen, and O. Brahma, *Advanced-machine-learning: week 8*, https://github.com/Dirrenvv99/Advanced-Machine-Learning/tree/main/Week%208.

[2] H. J. Kappen and F. Rodrıguez, "Mean field approach to learning in boltzmann machines", Pattern Recognition Letters **18**, 1317–1322 (1997).

[3] E. Schneidman, M. Berry II, R. Segev, and W. Bialek, "Weak pairwise correlations imply strongly correlated network states in a neural population", Nature **440**, 1007–12 (2006).

7 Appendix: Extra Plots

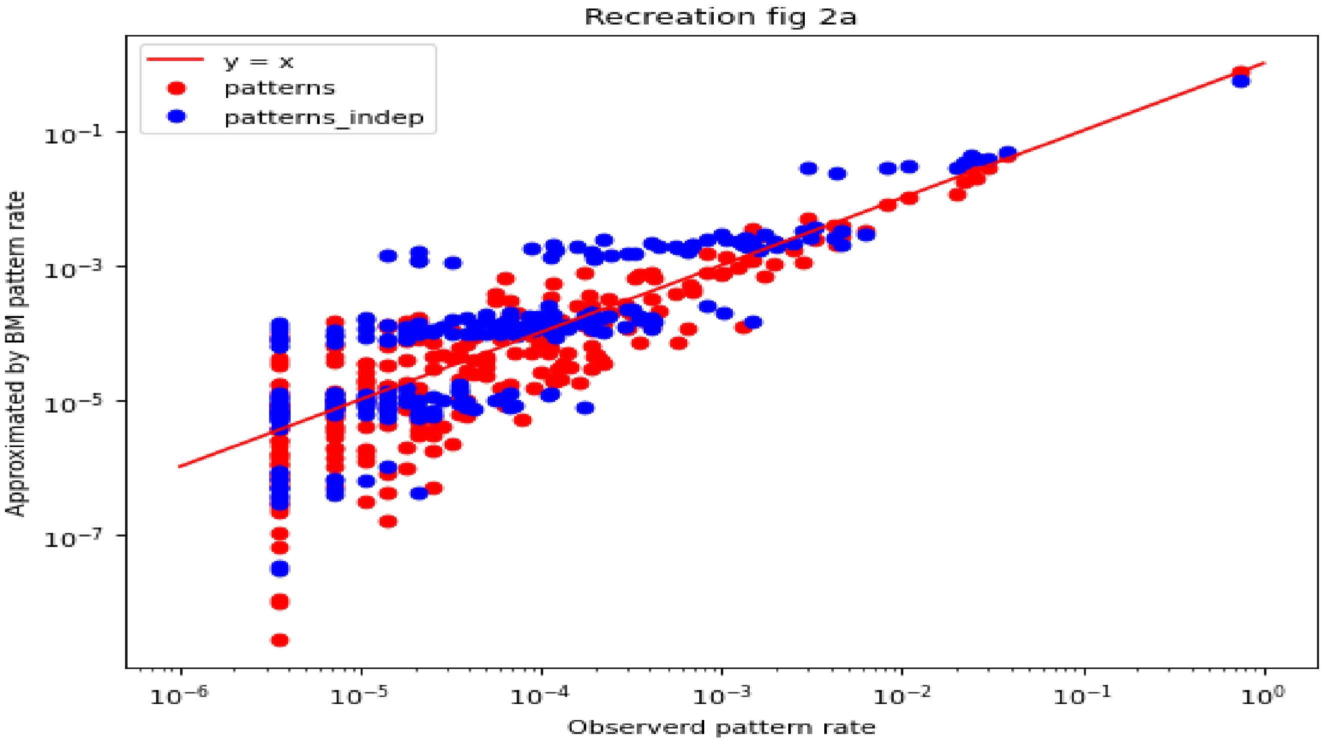
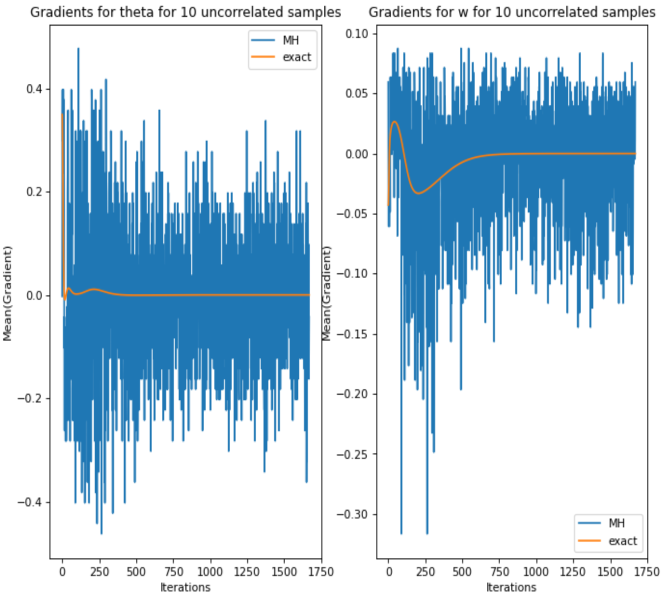
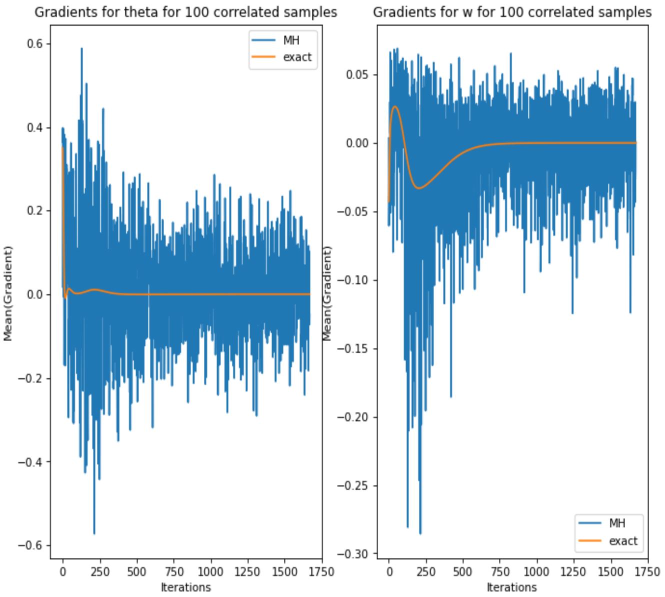


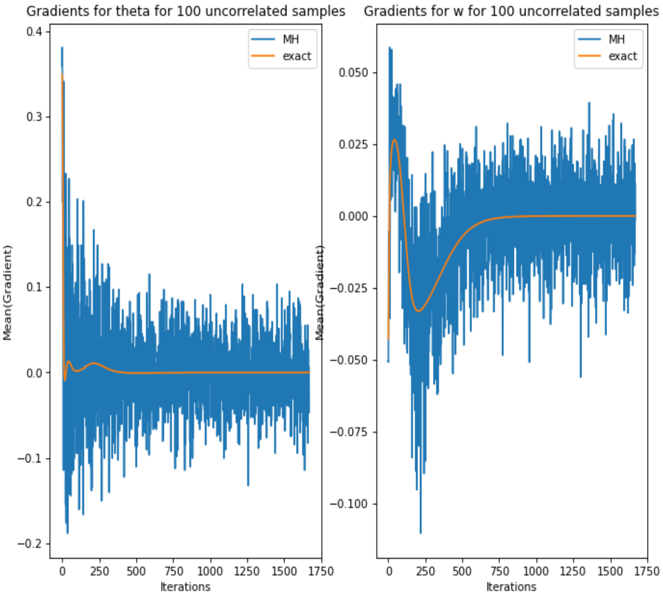
Figure 7: Recreation of the reference figure with a lower threshold. This figure more resembles the reference figure as can be seen by the clear spike patterns.


Gradients for theta for 10 uncorrelated samples
Gradients for w for 10 uncorrelated samples
MH
exact
Mean(Gradient)
Iterations
0.4
0.2
0.0
−0.2
−0.4
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
0 250 500 750 1000 1250 1500 1750

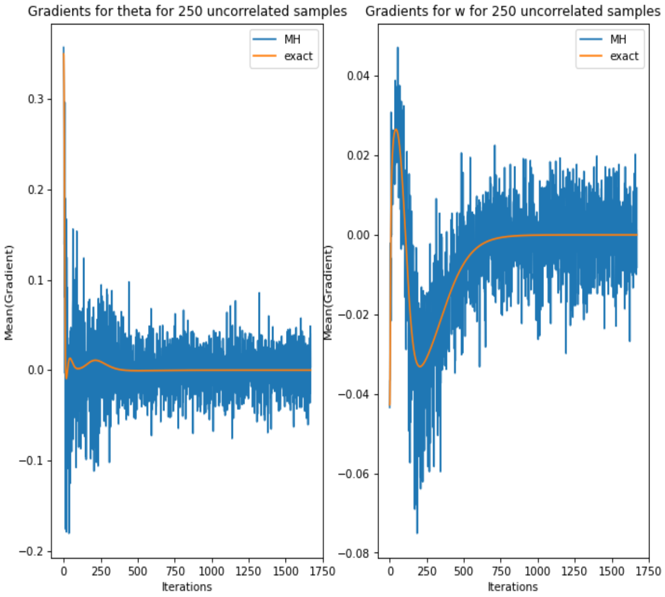
(a)


Gradients for theta for 100 correlated samples
Gradients for w for 100 correlated samples
MH
exact
Mean(Gradient)
Iterations
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
0 250 500 750 1000 1250 1500 1750

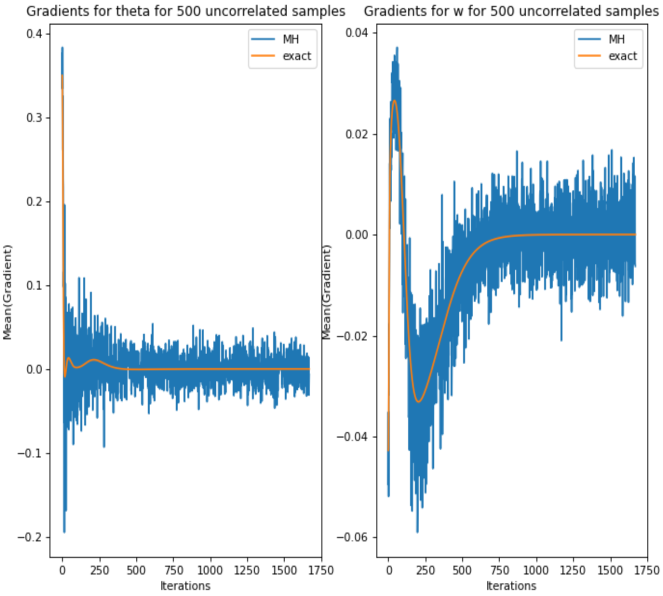
(b)


Gradients for theta for 100 uncorrelated samples
Gradients for w for 100 uncorrelated samples
MH
exact
Mean(Gradient)
Iterations
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
0 250 500 750 1000 1250 1500 1750

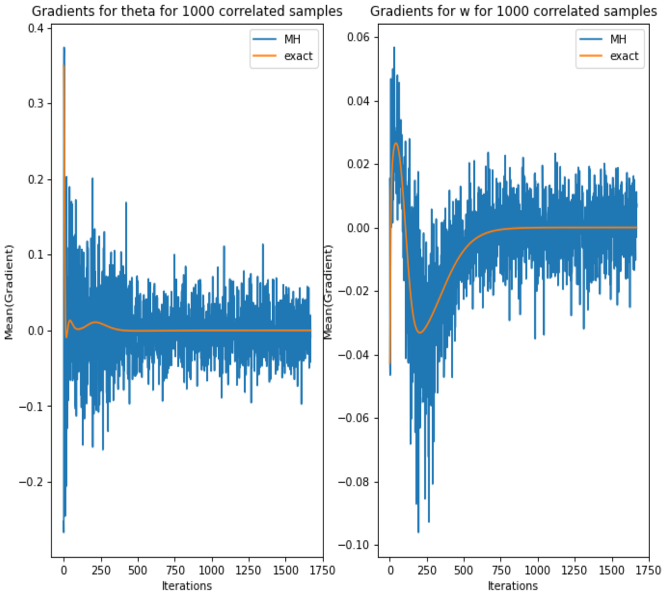
(c)


Gradients for theta for 250 uncorrelated samples
Gradients for w for 250 uncorrelated samples
MH
exact
Mean(Gradient)
Iterations
0.3
0.2
0.1
0.0
−0.1
−0.2
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
0 250 500 750 1000 1250 1500 1750

(d)


Gradients for theta for 500 uncorrelated samples
Gradients for w for 500 uncorrelated samples
MH
exact
Mean(Gradient)
Iterations
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0.04
0.02
0.00
−0.02
−0.04
−0.06
0 250 500 750 1000 1250 1500 1750

(e)


Gradients for theta for 1000 correlated samples
Gradients for w for 1000 correlated samples
MH
exact
Mean(Gradient)
Iterations
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
0 250 500 750 1000 1250 1500 1750

(f)

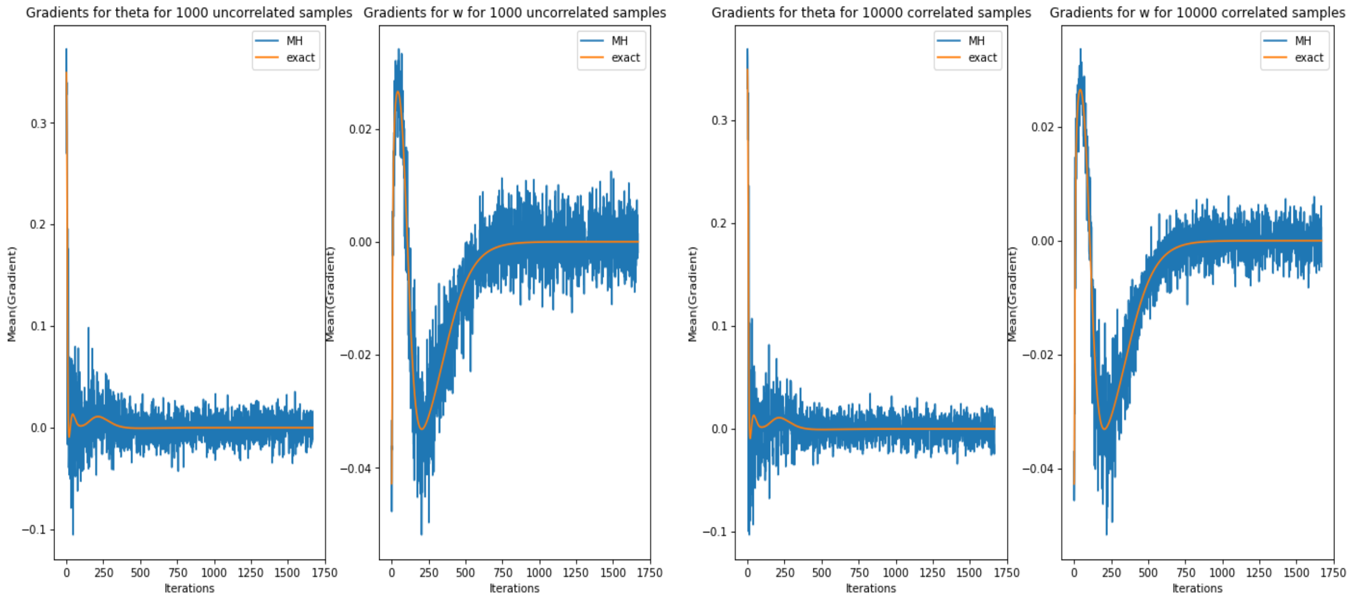


Figure 9: Gradients for exact and MH methods. Each step, gradients are calculated according to current w and θ values, and then w and θ are updated according to the exact gradients. We notice that even though the MH gradients fluctuate a lot, they do center around the exact gradients. We see that more samples lead to smaller fluctuations, and taking uncorrelated samples is better than correlated ones.

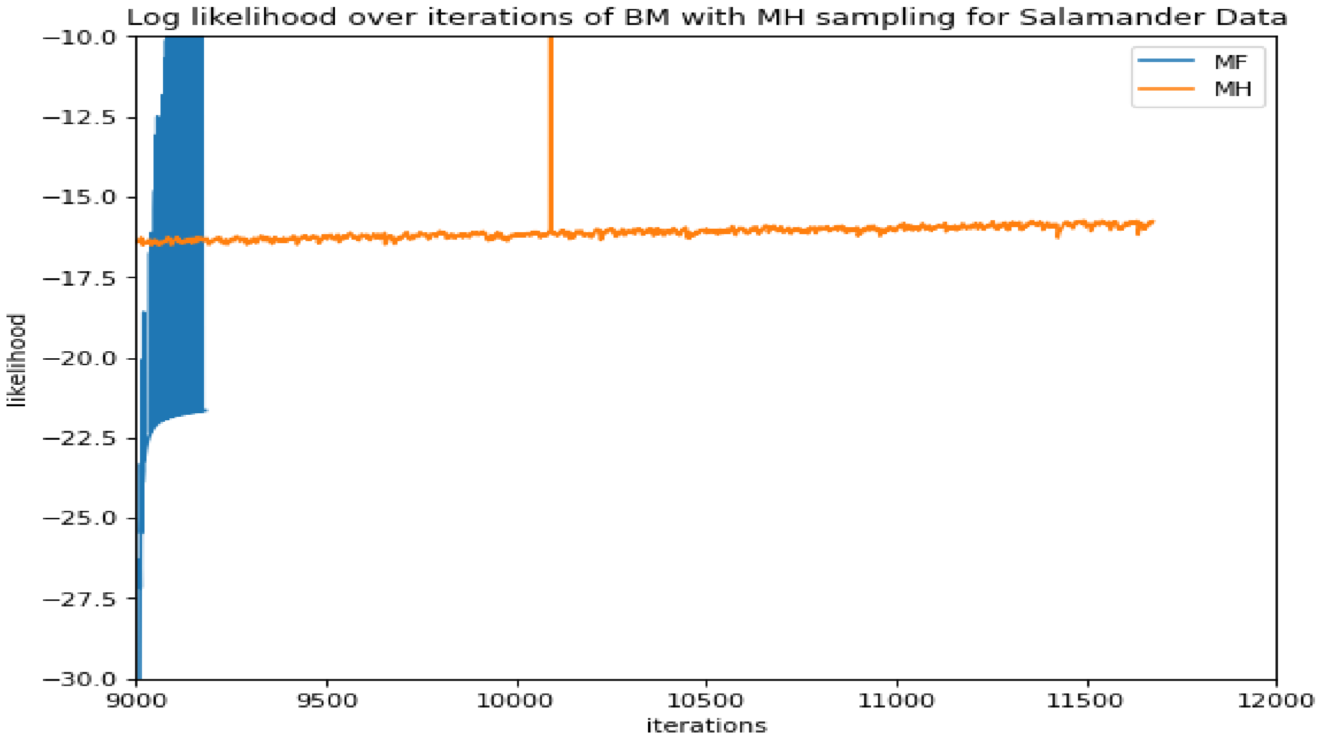


Figure 10: Plot showing the likelihood over iterations for the Metropolis Hastings (MH) and Mean Field/ Linear Response (MF) approximation from iteration 9000 and onwards. The MF line is started at iteration 9000 to make the comparison easier. Thus the correct iteration for this method is the given iteration -9000.