CS 6190: Probabilistic Machine Learning Spring 2022

Homework 2

Handed out: 22 Feb, 2022
Due: 11:59pm, 15 March, 2022

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free discuss the homework with the instructor or the TAs.
- Your written solutions should be brief and clear. You need to show your work, not just the final answer, but you do *not* need to write it in gory detail. Your assignment should be **no more than 10 pages**. Every extra page will cost a point.
- Handwritten solutions will not be accepted.
- The homework is due by **midnight of the due date**. Please submit the homework on Canvas.

Analytical problems [60 points + 25 bonus]

1. [10 points] Given a Gaussian likelihood, $p(x|\mu,\sigma) = \mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$, following the general definition of Jeffery's prior,

   (a) [5 points] show that given σ fixed, the Jeffery's prior over μ, $\pi_J(\mu) \propto 1$;
   Answer

   $$\log p(x|\mu,\sigma^2) = \log(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2$$
   $$\frac{\partial \log p(x|\mu,\sigma^2)}{\partial \mu} = \frac{x-\mu}{\sigma^2}$$
   $$\frac{\partial^2 \log p(x|\mu,\sigma^2)}{\partial \mu^2} = -\frac{1}{\sigma^2}$$

   Hence $\mathbf{I}(\mu) = \frac{1}{\sigma^2}$ and $\pi_j(\mu) = \frac{1}{\sigma} \propto 1$ as σ is constant.

   (b) [5 points] show that given μ fixed, the Jeffery's prior over σ, $\pi_J(\sigma) \propto \frac{1}{\sigma}$.
   Answer

   $$\log p(x|\mu,\sigma^2) = \log(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2$$
   $$\frac{\partial \log p(x|\mu,\sigma^2)}{\partial \sigma} = \frac{\sigma^2-(x-\mu)^2}{\sigma^3}$$
   $$\frac{\partial^2 \log p(x|\mu,\sigma^2)}{\partial \sigma^2} = -\frac{2}{\sigma^2}$$

   Hence $\mathbf{I}(\mu) = \frac{2}{\sigma^2}$ and $\pi_j(\sigma) = \frac{\sqrt{2}}{\sigma} \propto \frac{1}{\sigma}$.

2. [5 points] Derive the Jeffery's prior for λ in the Poisson likelihood, $p(x = n) = e^{-\lambda}\frac{\lambda^n}{n!}$.

Answer

$$\log p(x = n) = -\lambda + \frac{\log \lambda}{(n-1)!}$$

$$\frac{\log p(x = n)}{\partial \lambda} = -1 + \frac{1}{\lambda(n-1)!} = \frac{n - \lambda}{\lambda}$$

$$\frac{\log^2 p(x = n)}{\partial \lambda^2} = -\frac{1}{\lambda}$$

Hence $\mathbf{I}(\lambda) = \frac{1}{\lambda}$ and $\pi_j(\lambda) = \frac{1}{\sqrt{\lambda}}$

3. [5 points] Given an infinite sequence of Independently Identically Distributed (IID) random variables, show that they are exchangeable.

Answer

Let Random Variables be $x_1, x_2, ..., x_n$ and $f(x)$ is the PDF for any x_i. They are iid so they follow one single PDF. The joint PDF is written as

$$f_{x_1,x_2,...,x_n}(x_1, x_2, ..., x_n) =^{iid} f(x_1)f(x_2)...f(x_n)$$

As the term on the Right Hand Side can be multiplied in any order, hence $x_1, x_2, ..., x_n$ are exchangeable.

4. [10 points] We discussed Polya's Urn problem as an example of exchangeability. If you do not recall, please look back at the slides we shared in the course website. Now, given two finite sequences $(0, 1, 0, 1)$ and $(1, 1, 0, 0)$, derive their probabilities and show they are the same.

Answer

$$p(0, 1, 0, 1) = \frac{W_0}{B_0 + W_0} \times \frac{B_0}{B_0 + W_0 - 1 + a} \times \frac{W_0 - 1 + a}{B_0 + W_0 - 2 + 2a} \times \frac{B_0 - 1 + a}{B_0 + W_0 - 3 + 3a}$$

$$p(1, 1, 0, 0) = \frac{B_0}{B_0 + W_0} \times \frac{B_0 - 1 + a}{B_0 + W_0 - 1 + a} \times \frac{W_0}{B_0 + W_0 - 2 + 2a} \times \frac{W_0 - 1 + a}{B_0 + W_0 - 3 + 3a}$$

These are equal with numerators in different order.

5. [10 points] For the logistic regression model, we assign a Gaussian prior over the feature weights, $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda\mathbf{I})$. Please derive the Newton-Raphson updates.

Answer

Let $\mathbf{w}$ be the model parameters and $\mathbf{x}$ be the input feature vector, then $\mathbf{y}_n = \sigma(\mathbf{w}^T\mathbf{x})$.

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}) = \left[\prod_{i=1}^{n} y_n^{t_n}(1 - y_n)^{1-t_n}\right] \frac{1}{(2\pi)^{d/2}\lambda^{1/2}} \exp \frac{-\mathbf{w}^T\mathbf{w}}{2\lambda}$$

Taking negative log on both sides and then differentiating it with respect to $\mathbf{w}$, we get

$$\frac{\partial - \log p(\mathbf{w}|\mathbf{t}, \mathbf{X})}{\partial \mathbf{w}} = -\sum_{i=1}^{n}(t_n - y_n)\mathbf{x} + \frac{\mathbf{w}}{\lambda} = -\mathbf{X}^T(\mathbf{t} - \mathbf{y}) + \frac{\mathbf{w}}{\lambda}$$

We get the above result because we know that $\frac{\partial y_n}{\partial \mathbf{w}} = y_n(1 - y_n)\mathbf{x}$

Differentiating again we get

$$\frac{\partial^2 - \log p(\mathbf{w}|\mathbf{t}, \mathbf{X})}{\partial \mathbf{w}^2} = -\sum_{i=1}^{n} y_n(1-y_n)\mathbf{x}_n\mathbf{x}_n^T + \frac{\mathbf{I}}{\lambda} = \mathbf{X}^T\mathbf{R}\mathbf{X} + \frac{\mathbf{I}}{\lambda}$$

Hence the update rule will be

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \left[\mathbf{X}^T\mathbf{R}\mathbf{X} + \frac{\mathbf{I}}{\lambda}\right]^{-1}\left[\mathbf{X}^T(\mathbf{t}-\mathbf{y}) + \frac{\mathbf{w}^{old}}{\lambda}\right]$$

6. [**Bonus**][20 points] For the probit regression model, we assign a Gaussian prior over the feature weights, $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda\mathbf{I})$. Please derive the Newton-Raphson updates.

Answer

In case of probit regression, we know that $y_n = \psi(\mathbf{w}^T\phi(\mathbf{x}_n)) = \int_{-\infty}^{a_n}\mathcal{N}(x|0,1)dx$.

Differentiating and double differentiating y_n with respect to $\mathbf{w}$ we get

$$\frac{\partial y_n}{\partial \mathbf{w}} = \frac{\exp\frac{-a_n^2}{2}}{\sqrt{2\pi}}\phi(\mathbf{x}_n)^T$$

$$\frac{\partial^2 y_n}{\partial \mathbf{w}^2} = -\frac{\exp\frac{-a_n^2}{2}}{\sqrt{2\pi}}\phi(\mathbf{x}_n)\mathbf{w}^T\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T$$

These results will be useful later but first let's write the posterior distribution.

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}) = \left(\prod_{i=1}^{n} y_n^{t_n}(1-y_n)^{1-t_n}\right)\frac{1}{(2\pi)^{d/2}\lambda^{1/2}}\exp\frac{-\mathbf{w}^T\mathbf{w}}{2\lambda}$$

$$\log p(\mathbf{w}|\mathbf{t}, \mathbf{X}) = \left(\sum_{i=1}^{n} t_n\log(y_n) + (1-t_n)\log(1-y_n)\right) - \frac{d}{2}\log(2\pi) - \frac{\log\lambda}{2} - \frac{\mathbf{w}^T\mathbf{w}}{2\lambda}$$

$$\frac{\partial(-\log(\mathbf{w}|\mathbf{t}, \mathbf{X}))}{d\mathbf{w}} = -\sum_{i=1}^{n}\frac{t_n - y_n}{y_n(1-y_n)}\frac{\exp -a^2/2}{\sqrt{2\pi}}\phi(\mathbf{x}_n)^T + \frac{\mathbf{w}^T}{\lambda}$$

$$\frac{\partial^2(-\log(\mathbf{w}|\mathbf{t}, \mathbf{X}))}{d\mathbf{w}^2} = \frac{\mathbf{I}}{\lambda} + \sum_{i=1}^{n}\frac{t_n - y_n}{y_n(1-y_n)}\frac{\exp -a^2/2}{\sqrt{2\pi}}\phi(\mathbf{x}_n)^T\mathbf{w}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T +$$

$$\left(\frac{1-t_n}{(1-y_n)^2} + \frac{t_n}{y_n^2}\right)\left(\frac{\exp -a^2/2}{\sqrt{2\pi}}\right)^2\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T = \Phi^T\mathbf{R}\Phi + \frac{\mathbf{I}}{\lambda}$$

The Newton-Raphson update equation is given as:

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \left[\Phi^T\mathbf{R}\Phi + \frac{\mathbf{I}}{\lambda}\right]^{-1}\left[\mathbf{X}^T\left(\frac{t_n - y_n}{y_n(1-y_n)}\frac{\exp -a^2/2}{\sqrt{2\pi}}\right) + \frac{\mathbf{w}^{old}}{\lambda}\right]$$

7. [10 points] What are the link functions of the following models?

   (a) [5 points] Logistic regression

   Answer

$$\psi(y) = \psi(\sigma(\mathbf{w}^T\mathbf{x}))$$

   For logistic regression, the link function is $\psi = \sigma^{-1}$

$$y = \frac{1}{1+e^{-x}}$$

$$e^{-x} = \frac{1}{y} - 1$$

$$x = \log(\frac{y}{1-y})$$

$$\sigma^{-1}(x) = \log(\frac{x}{1-x})$$

(b) [5 points] Poisson regression: $p(x = n) = e^{-\lambda}\frac{\lambda^n}{n!}$ where $\lambda = \mathbf{w}^\top \boldsymbol{\phi}$.
Answer We know that $\mathbb{E}(x) = \lambda$

$$\lambda = \exp \mathbf{w}^T \phi = f(\mathbf{w}^T \phi)$$

$$\log \lambda = \mathbf{w}^T \phi = f^{-1}(\lambda)$$

$$\psi(\lambda) = f^{-1}(f(\mathbf{w}^T \phi)) = \mathbf{w}^T \phi$$

The link function ψ is as follows:

$$\psi(\lambda) = \log(\lambda)$$

8. [10 points] As we discussed in the class, the probit regression model is equivalent to given each feature vector $\boldsymbol{\phi}$, sampling a latent variable z from $\mathcal{N}(z|\mathbf{w}^\top \boldsymbol{\phi}, 1)$, and then sampling the binary label t from the step distribution, $p(t|z) = \mathbb{1}(t = 0)\mathbb{1}(z < 0) + \mathbb{1}(t = 1)\mathbb{1}(z \geq 0)$ where $\mathbb{1}(\cdot)$ is the indicator function. Show that if we marginalize out z, we recover the original likelihood of the probit regression.
Answer

$$p(t = 1|z) = \mathbf{I}(t = 1)\mathbf{I}(z \geq 0)$$

$$p(t = 1) = \int_0^\infty \frac{1}{\sqrt{2\pi}} \exp - \frac{(z - \mathbf{w}^T \phi)^2}{2} dz$$

Let $z = y + \mathbf{w}^T \phi$, then

$$p(t = 1) = \int_{-\mathbf{w}^T \phi}^\infty \frac{1}{\sqrt{2\pi}} \exp - \frac{y^2}{2} dy$$

$$p(t = 1) = \int_{-\mathbf{w}^T \phi}^\infty \mathcal{N}(y|0, 1) dy$$

$$p(t = 1) = \psi(\mathbf{w}^t \phi)$$

In the above equation we used symmetric nature of gaussian distribution. Similarly for $p(t = 0)$,

$$p(t = 0) = \int_{-\infty}^{-\mathbf{w}^T \phi} \frac{1}{\sqrt{2\pi}} \exp - \frac{y^2}{2} dy$$

Using the symmetric nature of gaussian distribution.

$$p(t=0) = 1 - \psi(\mathbf{w}^T\phi)$$

Hence by marginalising z, we can write

$$p(t|\mathbf{w},\phi) = \prod_i \psi(\mathbf{w}^T\phi)^{t_i}(1-\psi(\mathbf{w}^T\phi)^{t_i})^{1-t_i}$$

9. [**Bonus**][5 points] For polynomial regression (1d feature vector), show that given N training points, you can always choose the highest order M for the polynomial terms such that your model results in 0 training error (*e.g.,* mean squared error or mean absolute error). Please give the corresponding regression function as well.

Practice [40 points + 45 Bonus]

1. [15 Points] Let us generate a simulation dataset for fun. We consider a linear regression model $y(\mathbf{x},\mathbf{w}) = w_0 + w_1 x$. We set the ground-truth $w_0 = -0.3$ and $w_1 = 0.5$. We generate 20 samples $[x_1,\ldots,x_{20}]$ from the uniform distribution in $[-1,1]$. For each sample x_n, we obtain an sample y_n by first calculating $w_0 + w_1 x_n$ with the ground-truth values of w_0 and w_n, and then adding a Gaussian noise with zero mean, standard deviation 0.2. Now let us verify what we have discussed in the class. We use a Bayesian linear regression model. The prior of $\mathbf{w}$ is $\mathcal{N}(\mathbf{w}|\mathbf{0},\alpha\mathbf{I})$, and the likelihood for each sample is $p(y_n|\mathbf{x}_n,\mathbf{w}) = \mathcal{N}(y_n|w_0 + w_1 x, \beta^{-1}\mathbf{I})$. Here we set $\alpha = 2$ and $\beta = 25$.

   (a) [3 points] Draw the heat-map of the prior $p(\mathbf{w})$ in the region $w_0 \in [-1,1]$ and $w_1 \in [-1,1]$, where you represent the values of $p(\mathbf{w})$ for different choices of $\mathbf{w}$ with different colors. The darker some given color (*e.g.,* red), the larger the value; the darker some the other given color (*e.g.,* blue), the smaller the value. Most colors should be in between. Then sample 20 instances of $\mathbf{w}$ from $p(\mathbf{w})$. For each w, draw a line $y = w_0 + w_1 x$ in the region $x, y \in [-1,1]$. Ensure these 20 lines are in the same plot. What do you observe?

   Answer

   Code details can be found here.

   Figure 1 and Figure 2 are the plots for the prior and the curves drawn from the prior. From these Figures we can see that all plots are likely and this is confirmed by the haphazard plots of the curves that are drawn from the sample.

   (b) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given $(\mathbf{x}_1, y_1)$. Now draw the heat map of the distribution. Also draw the ground-truth of w_0 and w_1 in the heat map. Then from the posterior distribution, sample 20 instances of $\mathbf{w}$, for each of which draw a line $y = w_0 + w_1 x$ in the region $x, y \in [-1,1]$. Ensure these 20 lines are in the same plot. Also draw (x_1, y_1) as a circle in that plot. What do you observe? Why?

   Answer

   Figure 3 and Figure 4 shows the posterior and the curves from the sample after (x_1, y_1) is seen by the model.

   From Figures 3 and 4, we can see that after a sample is drawn the choices are reduced significantly. There are still some odd curves but it's much better now.

   (c) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$. Then draw the plots as the above. What do you observe now?

   Answer

   Figure 5 and Figure 6 shows the posterior and the curves from the sample after (x_1, y_1) and (x_2, y_2) is seen by the model.

   From Figures 5 and 6, we can see that after two samples are drawn the choices are reduced significantly. This is also visible from the heatmap.

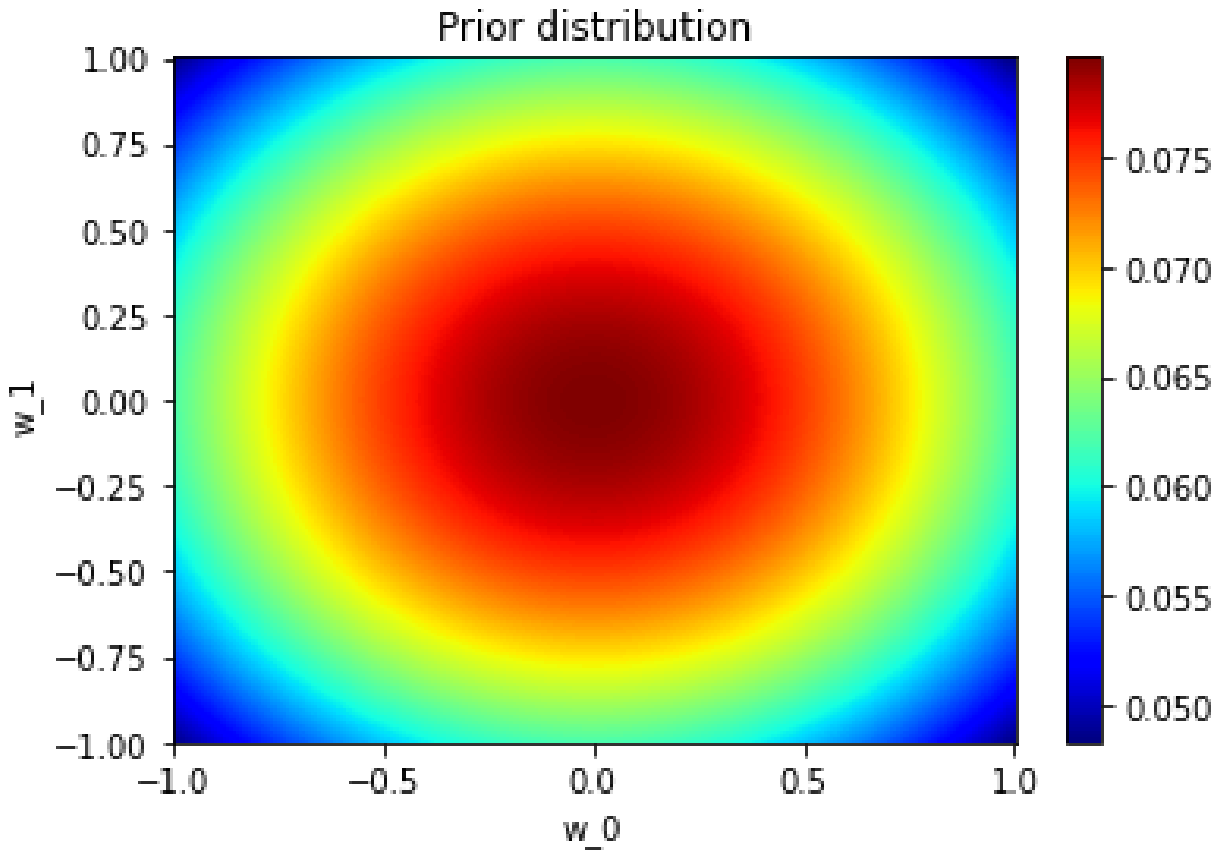

Figure 1: Heatmap of the Prior

(d) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_5, y_5)\}$. Then draw the plots as the above. What do you observe now?

Answer

Figure 7 and Figure 8 shows the posterior and the curves from the sample after (x_1, y_1) to (x_5, y_5) are seen by the model.

From Figures 7 and 8, we can see that after 5 samples are drawn the choices are reduced significantly. This is also visible from the heatmap.

(e) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given all the 20 data points. Then draw the plots as the above. What do you observe now?

Answer

Figure 9 and Figure 10 shows the posterior and the curves from the sample after all 20 points are seen by the model.

From Figures 9 and 10, we can see that after 20 samples are drawn the choices are reduced significantly. This is also visible from the heatmap.

As more data-points are seen by the model, the distribution on $\mathbf{w}$ changes such that the variance decreases with the mean approaching the true $\mathbf{w}$. Thus, all those $\mathbf{w}$ which can fit the seen data-points become more likely.

2. [25 points] We will implement Logistic regression and Probit regression for a binary classification task — bank-note authentication. Please download the data "bank-note.zip" from Canvas. The features and labels are listed in the file "bank-note/data-desc.txt". The training data are stored in the file

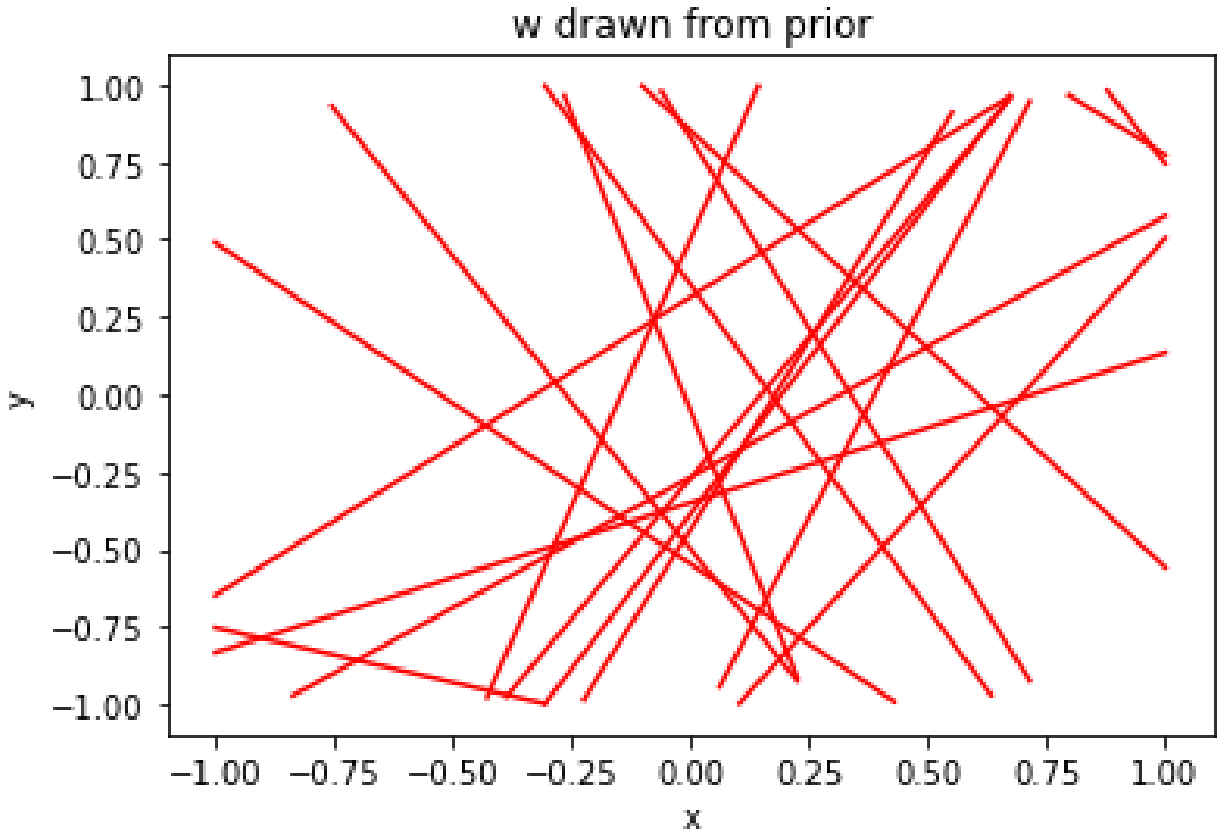

Figure 2: Curves drawn from the samples in prior

"bank-note/train.csv", consisting of 872 examples. The test data are stored in "bank-note/test.csv", and comprise of 500 examples. In both the training and testing datasets, feature values and labels are separated by commas. To ensure numerical stability and avoid overfitting, we assign the feature weights a standard normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

(a) [15 points] Implement Newton-Raphson scheme to find the MAP estimation of the feature weights in the logistic regression model. Set the maximum number of iterations to 100 and the tolerance level to be $1e-5$, i.e., when the norm of difference between the weight vectors after one update is below the tolerance level, we consider it converges and stop updating the weights any more. Initially, you can set all the weights to be zero. Report the prediction accuracy on the test data. Now set the initial weights values be to be randomly generated, say, from the standard Gaussian, run and test your algorithm. What do you observe? Why?

Answer

Code details can be found here.

Setting all weights to zero, I got the algorithm to converge in 10 iterations and got the final accuracy of 95.6%. The learned weights are $[-1.50015252, -0.90812559, -0.98061604, -0.44317884]$.

Setting all weights to random, the algorithm ran for all 100 iterations and got the final accuracy of 63.6%. The learned weights are $[-976.93770373, -3299.21784144, 1432.59566538, 612.89583119]$.

I think the results for random are bad because the initial values were too large. Reducing this helps and I tried that by simply dividing it by 100.

(b) [10 points] Implement MAP estimation algorithm for Probit regression model. You can calculate the gradient and feed it to any optimization algorithm, say, L-BFGS. Set the maximum number

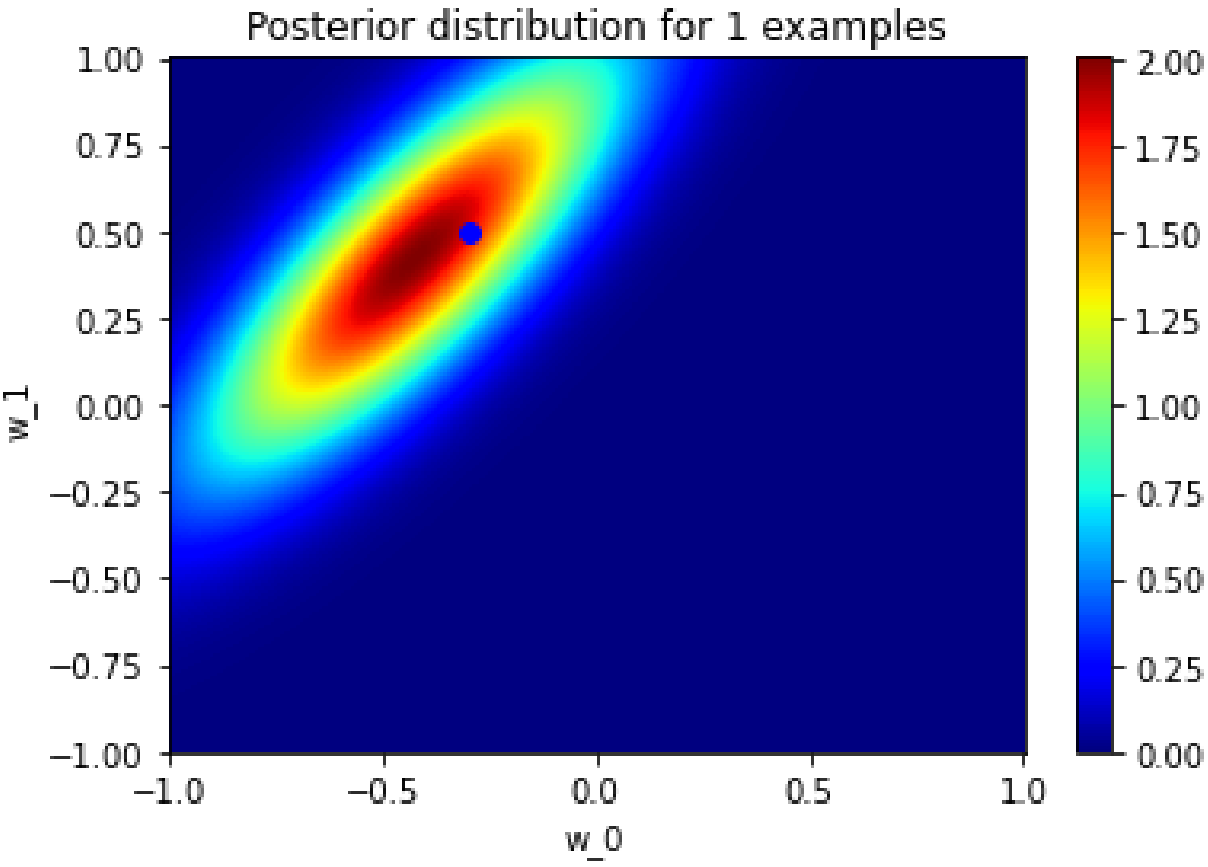

Figure 3: Heatmap of the Posterior after 1 sample

of iterations to 100 and the tolerance level to $1e-5$. Initially, you can set all the weights to zero. Report the prediction accuracy on the test data. Compared with logistic regression, which one is better? Now set the initial weights values be to be randomly generated, say, from the standard Gaussian, run and test your algorithm. What do you observe? Can you guess why?

Answer

To implement LBFGS, I used minimize routine.

Setting all weights to zero, I got the final accuracy of 95.6%. The learned weights are $[-1.50015252, -0.90812559, -0.98061604, -0.44317884]$. These are exactly the same as ones with logistic regression.

Setting all weights to random, the algorithm ran for all 100 iterations and got the final accuracy of 96.6%. The learned weights are $[-1.50015252 - 0.90812559 - 0.98061604 - 0.44317884]$. Compared to logistic regression, now we got convergence and learned the weights that gave an accuracy of 96.6%.

LBFGS helps because the algorithm makes use of the Hessian to compute the update direction.

(c) [**Bonus**][15 points]. Implement Newton-Raphson scheme to find the MAP estimation for Probit regression. Report the prediction accuracy

Answer

I've used the code written before to do this. Newton-Raphson uses both derivatives and here are the results.

Initialising with zeros, it converged in 10 iterations and I got the final accuracy of 95.6%. The

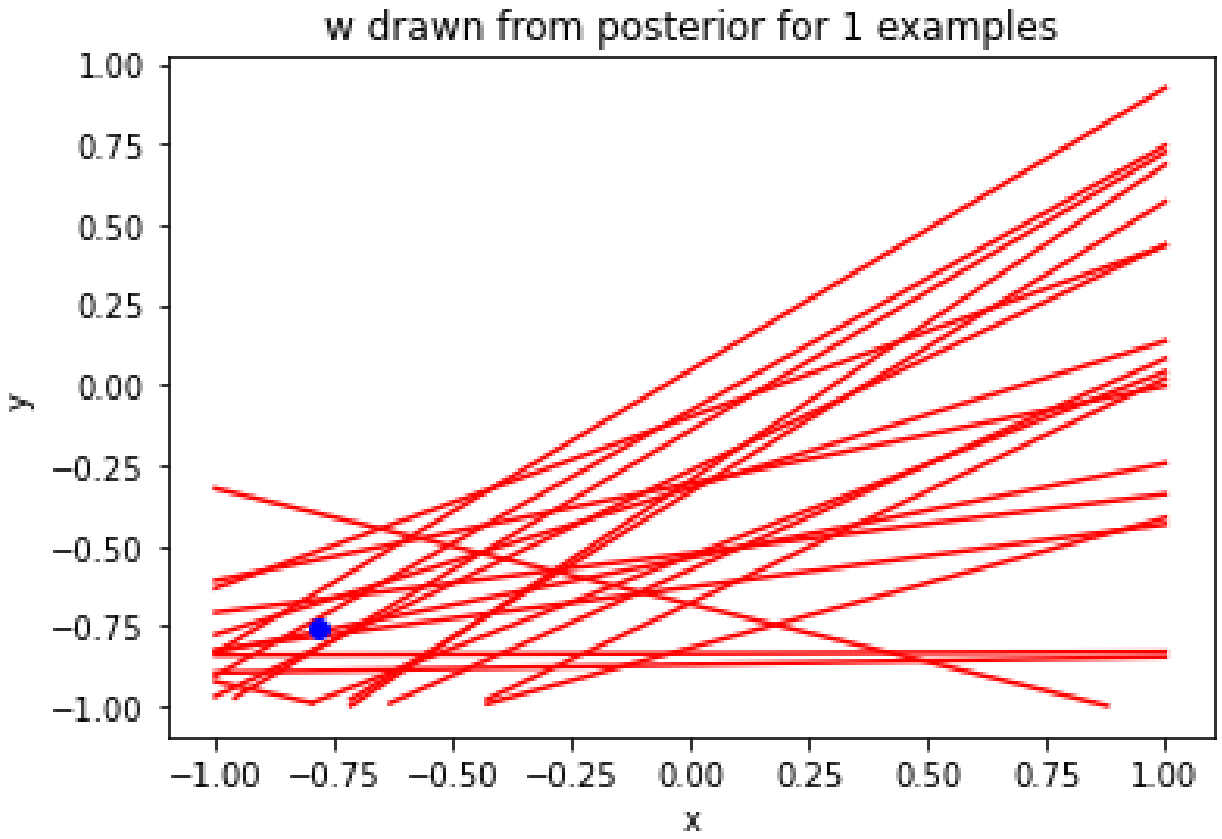

Figure 4: Curves drawn from the samples in posterior after 1 sample

learned weights are $[-1.50015252, -0.90812559, -0.98061604, -0.44317884]$. This is exactly the same as the cases before.

3. [**Bonus**][30 points] We will implement a multi-class logistic regression model for car evaluation task. The dataset is from UCI repository(`https://archive.ics.uci.edu/ml/datasets/car+evaluation`). Please download the processed dataset (car.zip) from Canvas. In this task, we have 6 car attributes, and the label is the evaluation of the car. The attribute and label values are listed in the file "data-desc.txt". All the attributes are categorical. Please convert each categorical attribute into binary features. For example, for "safety: low, med, high", we convert it into three binary features: "safety" is "low" or not, "safety" is "med" or not, and "safety" is "high" or not. The training data are stored in the file "train.csv", consisting of $1,000$ examples. The test data are stored in "test.csv", and comprise 728 examples. In both training and test datasets, attribute values are separated by commas; the file "data-desc.txt" lists the attribute names in each column. To ensure numerical stability and avoid overfitting, we assign the feature weights a standard normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

   (a) [15 points] Implement MAP estimation algorithm for multi-class logistic regression model. To do so, you can calculate the gradient and feed it to some optimization package, say, L-BFGS. Report the prediction accuracy on the test data.

   (b) [15 points] Let us use an "ugly" trick to convert the multi-class classification problem into a binary classification problem. Let us train four logistic regression models, where each model predicts one particular label, i.e., "unacc" or not, "acc" or not, "good" or not, and "vgood" or not. Then for each test example, we run the models to get four logistic scores, i.e., the probability that each label is one. We choose the label with the highest score as the final

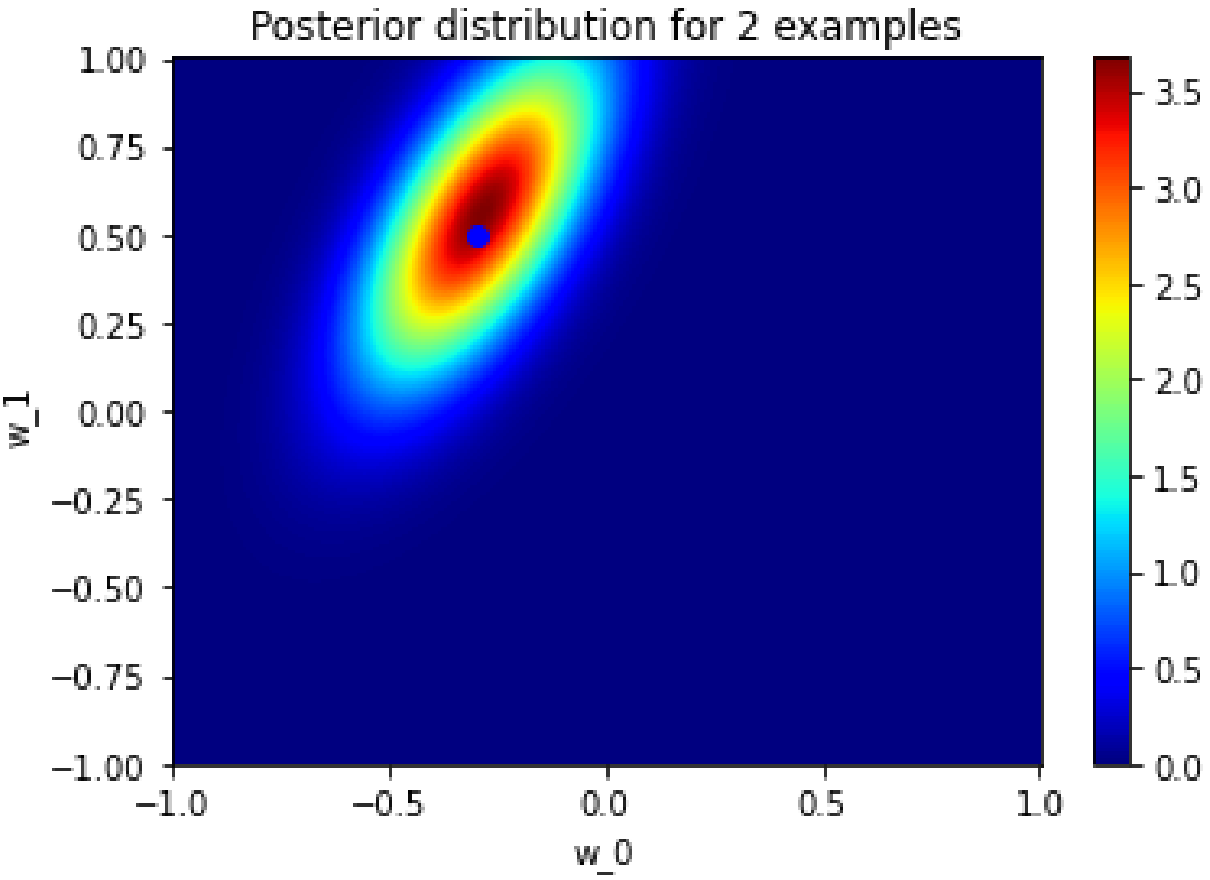

Figure 5: Heatmap of the Posterior after 2 sample

prediction. Report the prediction accuracy on the test data. As compared with multi-class logistic regression ,which one is better?

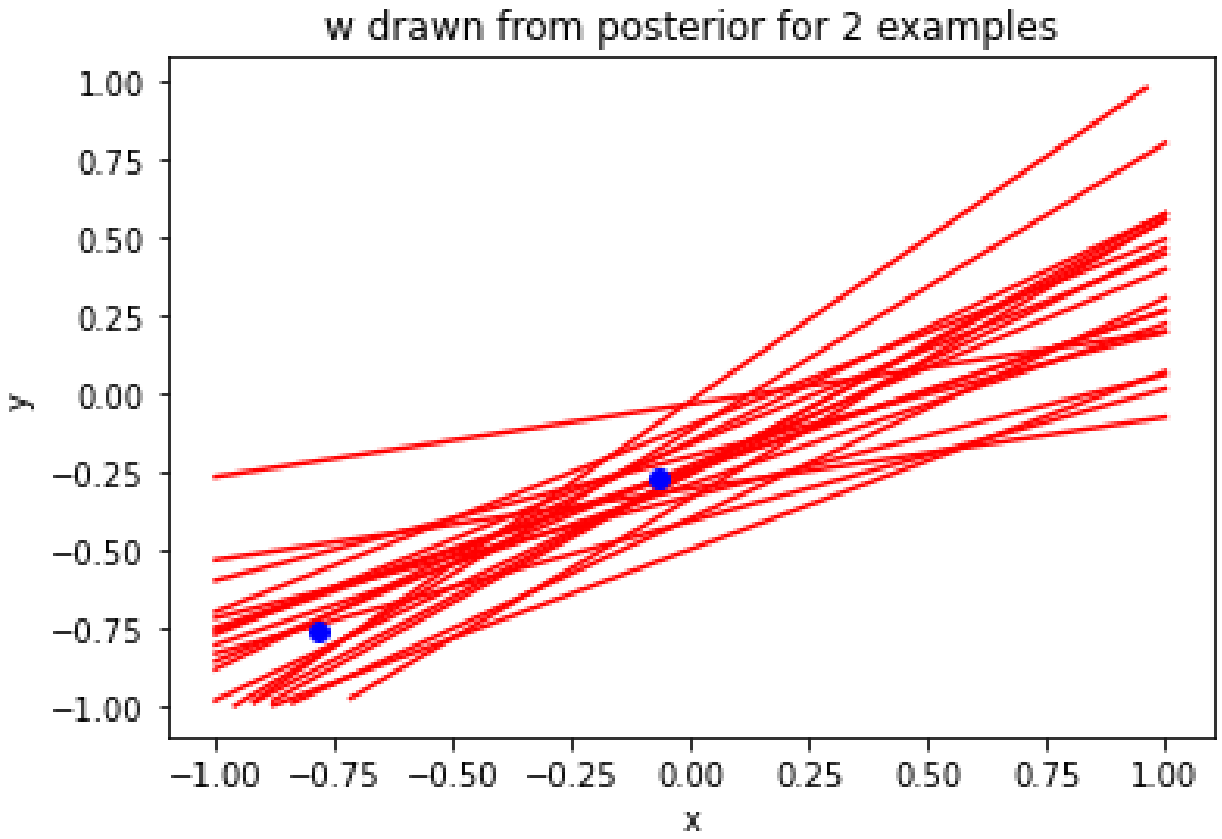

Figure 6: Curves drawn from the samples in posterior after 2 sample

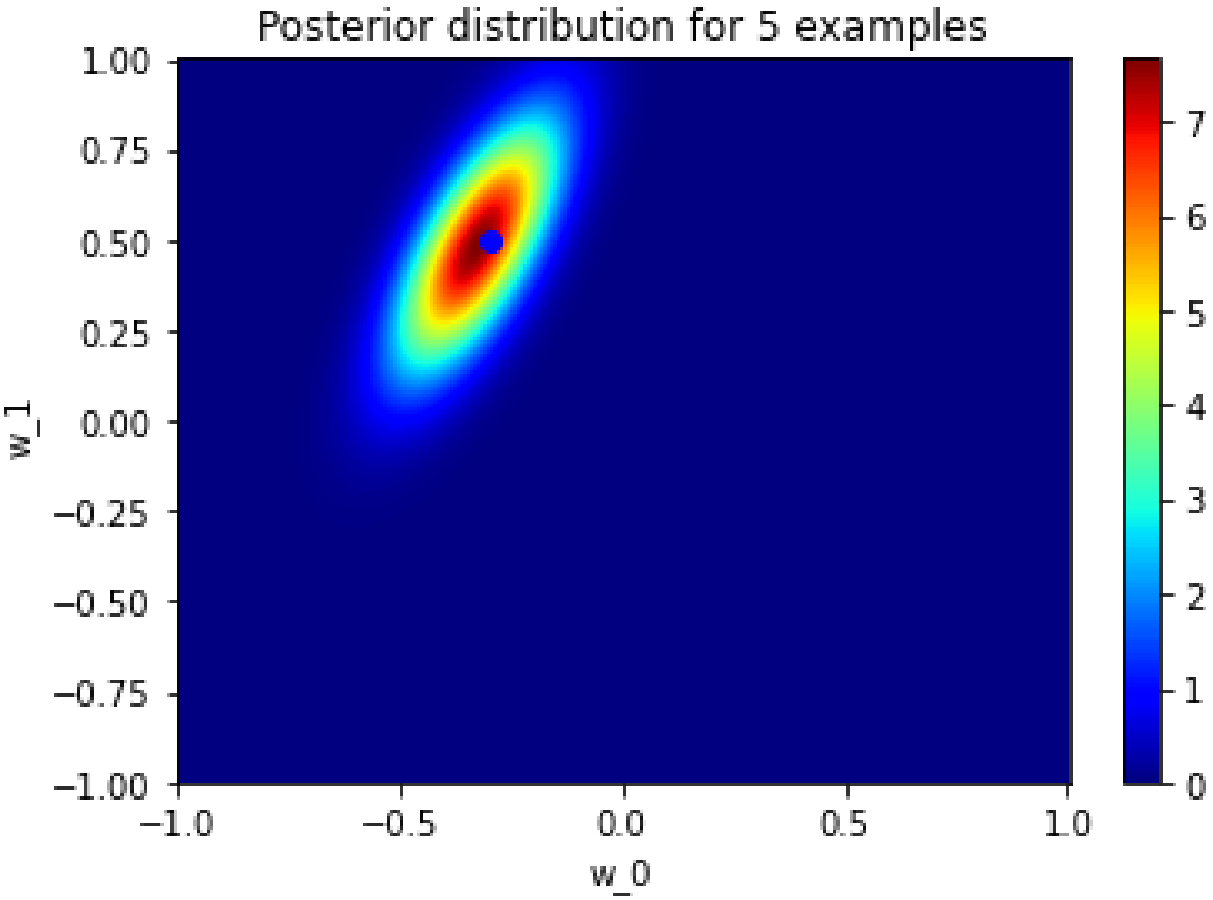

Figure 7: Heatmap of the Posterior after 5 sample

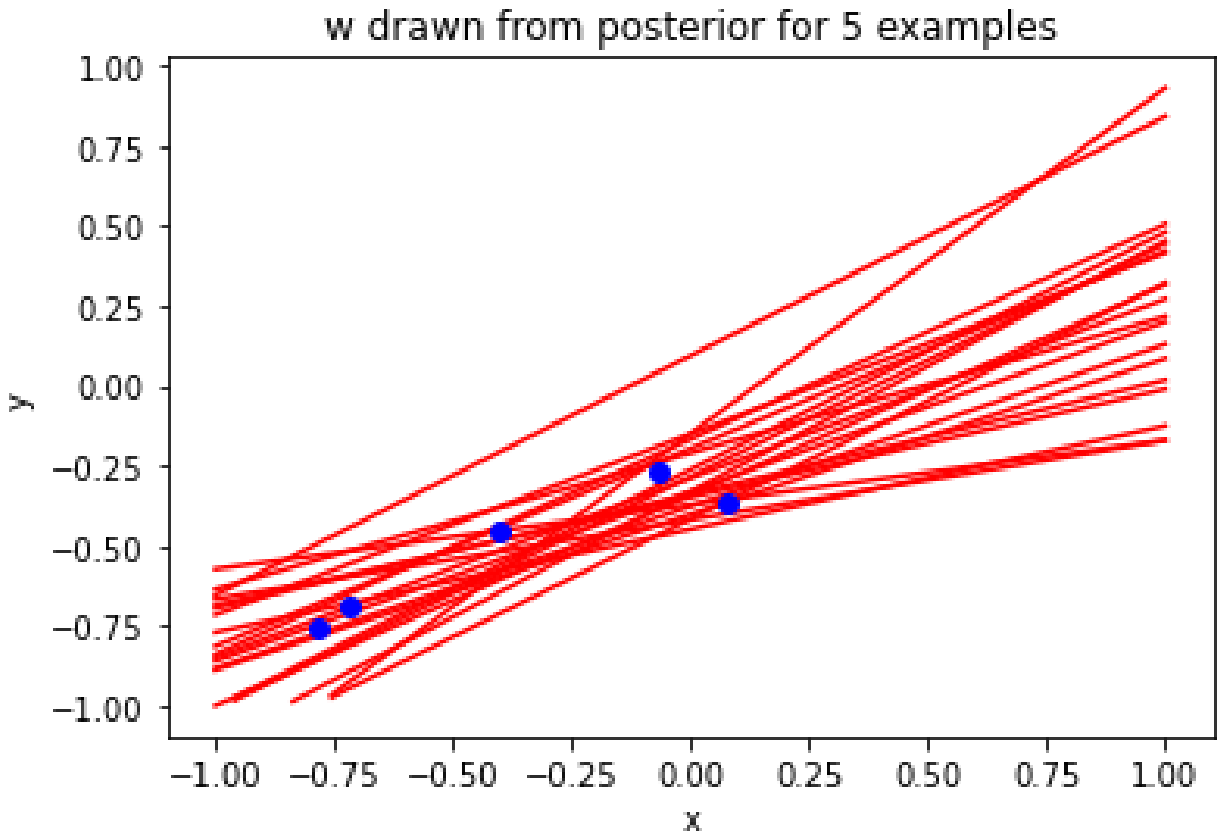

Figure 8: Curves drawn from the samples in posterior after 5 sample

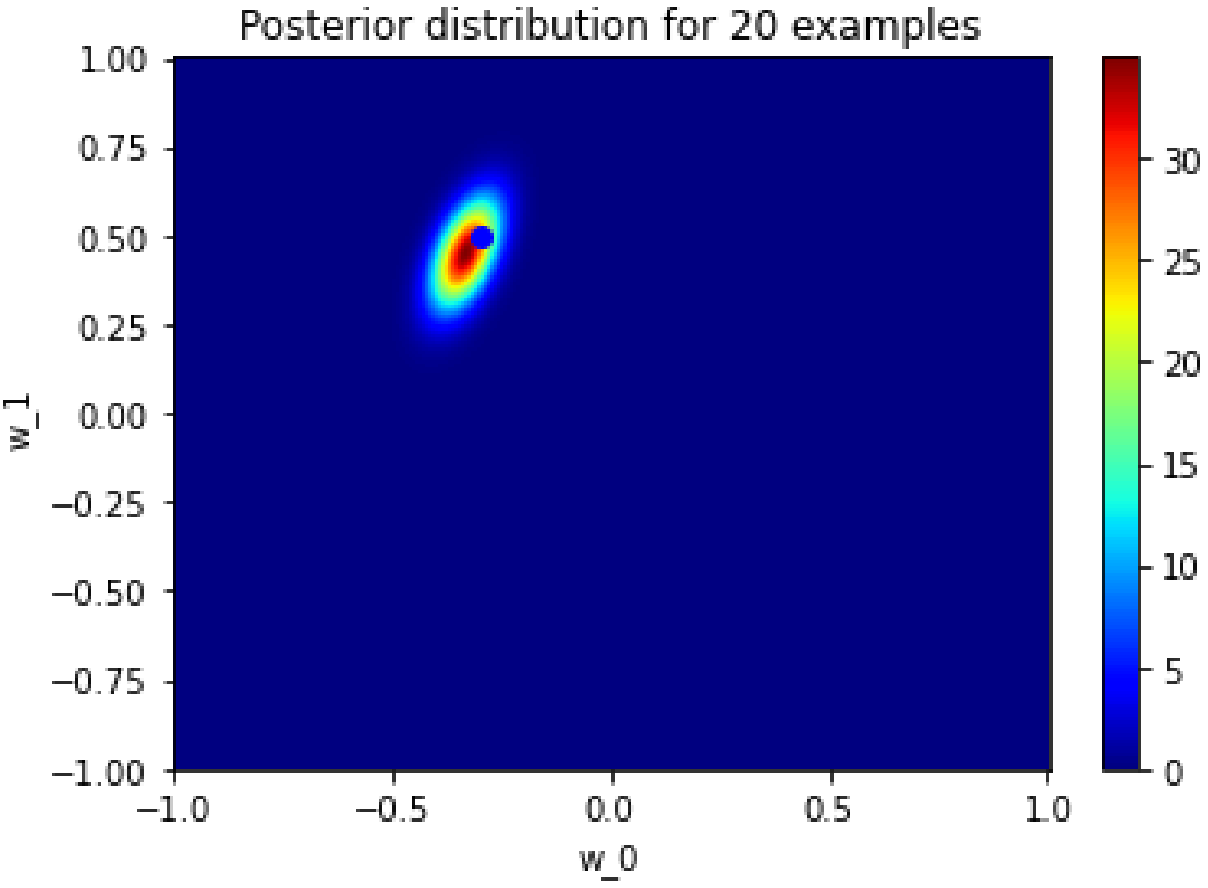

Figure 9: Heatmap of the Posterior after 20 sample

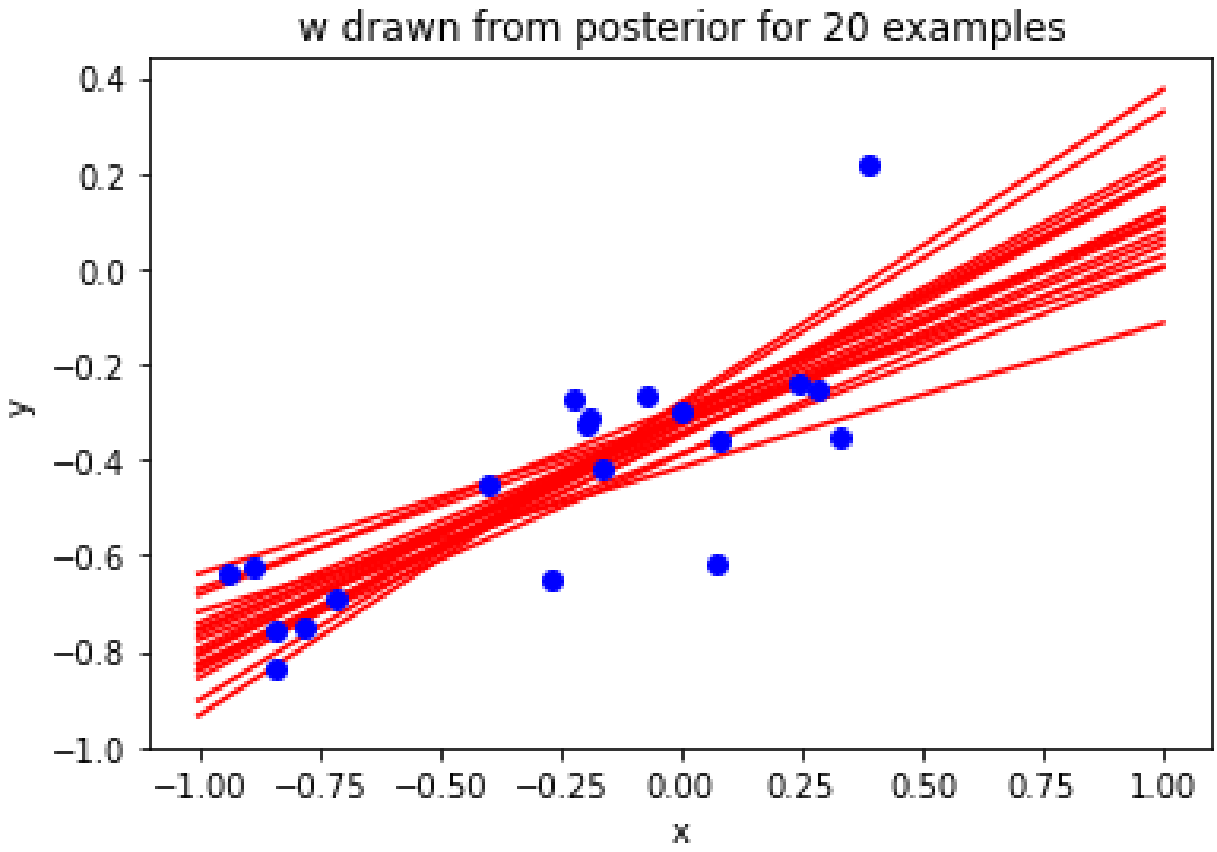

Figure 10: Curves drawn from the samples in posterior after 20 sample