1a) $P(D|\theta^{MLE}) = \prod_{n=1}^{N} P(Y=y_n \mid X=x_n, \theta^{MLE})$

$$= \exp\left(\operatorname{Log} \prod_{n=1}^{N} P(Y=y_n \mid X=x_n, \theta^{MLE})\right)$$

$$= \exp\left(\operatorname{Log} \prod_{n=1}^{N} \left(\frac{1}{1+\exp(-(wx_n^T+b))}\right)^{[y_n=1]} \left(\frac{\exp(-(wx_n^T+b))}{1+\exp(-(wx_n^T+b))}\right)^{[y_n=0]}\right)$$

$$= \exp\left(\sum_{n=1}^{N} \operatorname{Log}\left(\frac{1}{1+\exp(-(wx_n^T+b))}\right)^{[y_n=1]}\left(\frac{\exp(-(wx_n^T+b))}{1+\exp(-(wx_n^T+b))}\right)^{[y_n=0]}\right)$$

where $w = [w_{1_{mle}}, w_{2_{mle}}]$ and $b = [b_{mle}]$ $\{\theta^{MLE} = [w_{1_{mle}}, w_{2_{mle}}, b_{mle}]\}$

Similarly, $P(D|\theta_s)$ can be expressed as

$$\exp\left(\sum_{n=1}^{N} \operatorname{Log}\left(\frac{1}{1+\exp(-(w'x_n^T+b'))}\right)^{[y_n=1]}\left(\frac{\exp(-(w'x_n^T+b'))}{1+\exp(-(w'x_n^T+b'))}\right)^{[y_n=0]}\right)$$

where $w' = [w_{1_s}, w_{2_s}]$ and $b = [b_s]$ $\{\theta_s = [w_{1_s}, w_{2_s}, b_s]\}$

It is clear that when N is large, computing $P(D|\theta^{MLE})$ and $P(D|\theta_s)$ as a product of probabilities ($\prod_{n=1}^{N} P(Y=y_n|X=x_n,\theta^{MLE})$ and ~~θ_s~~ $\prod_{n=1}^{N} P(Y=y_n|X=x_n,\theta_s)$ respectively) will underflow. This is because the value of each $P(Y=y_n|X=x_n,\theta)$ lies between 0 and 1 (as it is a probability) and product of a large number of $P(Y=y_n|X=x_n,\theta)$ terms will result in a very small value which due to constraints on the smallest value that can be represented in a float or double data type defaults to zero (underflow). Therefore we have issues such as division by zero (when $P(D|\theta_s)$ underflows).

In order to overcome this and compute $P(D|\theta^{MLE})$ and $P(D|\theta_s)$ in a numerically stable manner, the two probabilities can be expressed as above.

1c) Implicit function representation of decision boundary:

$$w_1 x_1 + w_2 x_2 + b = 0$$

$$\Rightarrow w_2 x_2 = -w_1 x_1 - b$$

$$\Rightarrow x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2} \Rightarrow x_2 = a x_1 + c \text{ where } a = \frac{-w_1}{w_2} \; \& \; c = \frac{-b}{w_2}$$

2c) Our function:

$$\max_{w,b} \text{Log}\, \mathcal{N}(w, \mu_0, \Sigma_0) + \text{Log} \prod_{n=1}^{N} P(Y = y_n | X = x_n, \theta)$$

$$= \max_{w,b} \text{Log}\left(\frac{1}{\sqrt{|2\pi\Sigma_0|}} \exp\left(-\frac{1}{2}(w-\mu_0)^T \Sigma_0^{-1}(w-\mu_0)\right)\right) + \sum_{n=1}^{N} \text{Log}\, P(Y = y_n | X = x_n, \theta)$$

$$= \max_{w,b} \left\{-\frac{1}{2}\text{Log}|2\pi\Sigma_0| - \frac{1}{2}(w-\mu_0)^T \Sigma_0^{-1}(w-\mu_0) + \sum_{n=1}^{N} \text{Log}\left(\frac{1}{1+\exp(-y_n(w x_n^T + b))}\right)\right\}$$

$-\frac{1}{2}\text{Log}|2\pi\Sigma_0|$ is a constant so it can be ignored for optimization.

$$= \max_{w,b} -\left(\frac{1}{2}(w-\mu_0)^T \Sigma_0^{-1}(w-\mu_0) + \sum_{n=1}^{N} \text{Log}(1+\exp(-y_n(w x_n^T + b)))\right)$$

$$= \min_{w,b} \left(\frac{1}{2}(w-\mu_0)^T \Sigma_0^{-1}(w-\mu_0) + \sum_{n=1}^{N} \text{Log}(1+\exp(-y_n(w x_n^T + b)))\right)$$

$\mu_0 = 0$ and $\Sigma_0 = 100 I$

$$\Rightarrow \min_{w,b} \left(\frac{1}{2(100)} w^T w + \sum_{n=1}^{N} \text{Log}(1+\exp(-y_n(w x_n^T + b)))\right)$$

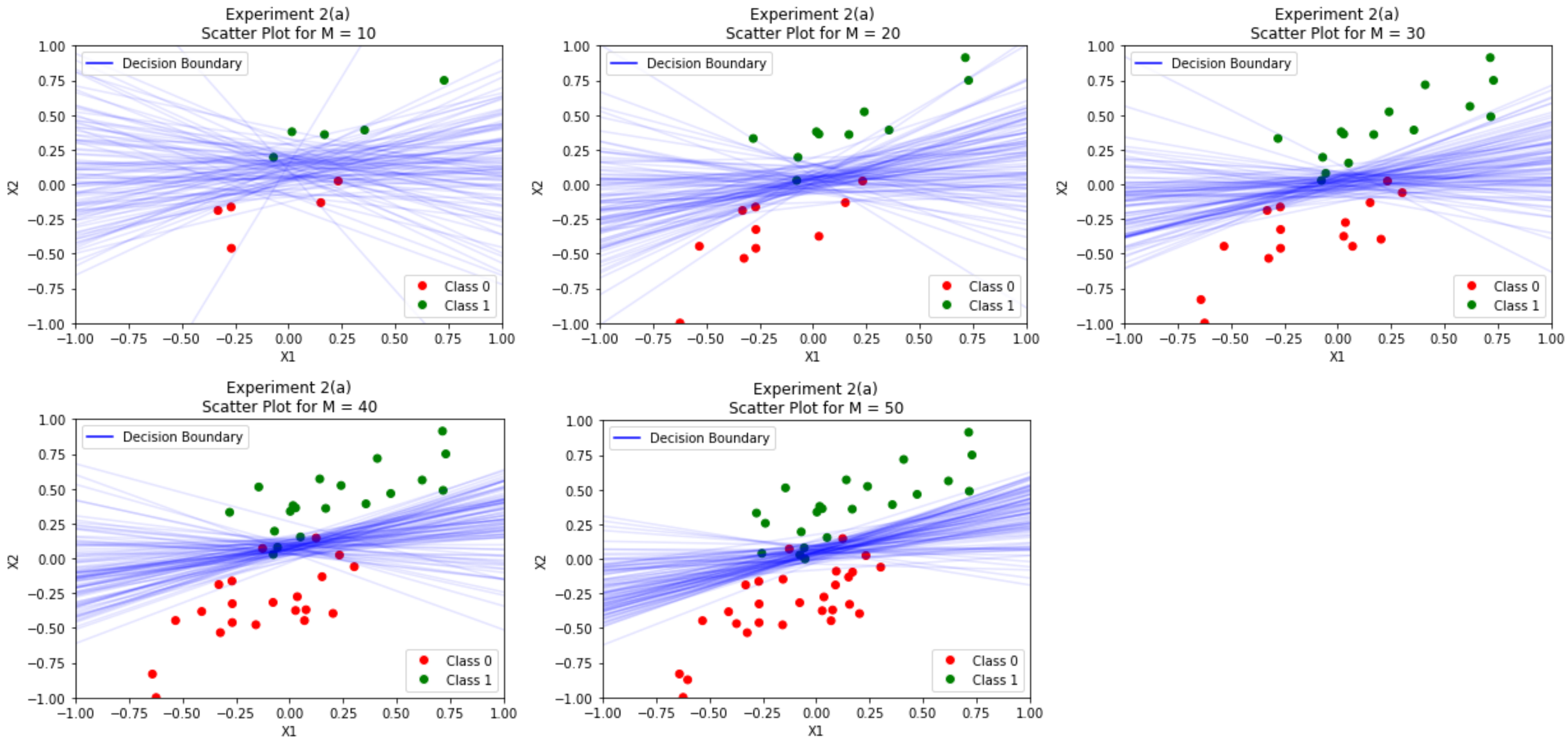
$$= \min_{w,b} \frac{1}{2} w^T w + 100 \sum_{n=1}^{N} \text{Log}(1+\exp(-y_n(w x_n^T + b)))$$

Scikit-learn's logistic regression cost function:

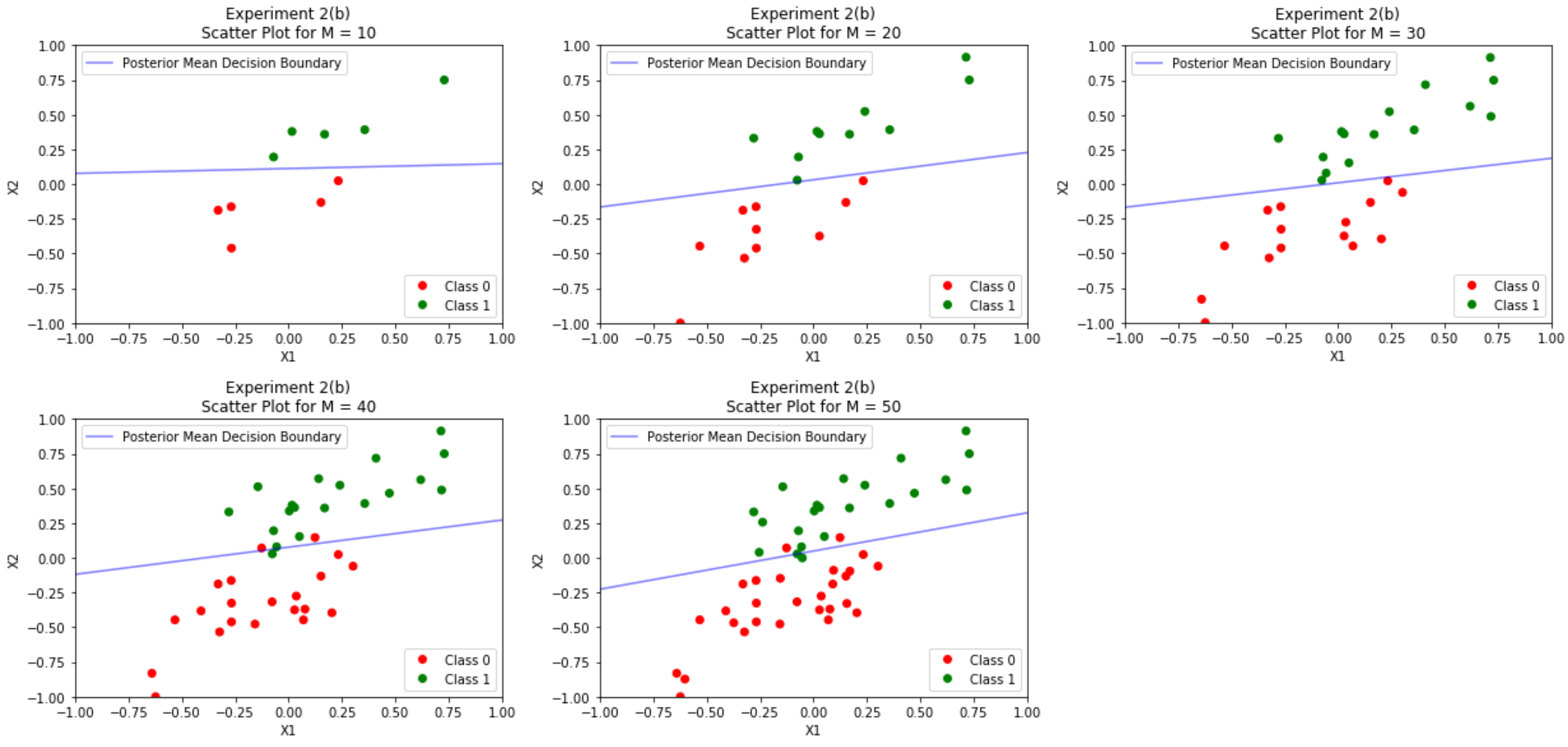
$$\min_{w,b} \left(\frac{1}{2} w^T w + C \sum_{n=1}^{N} \text{Log}(1+\exp(-y_n(w x_n^T + b)))\right)$$

$\therefore$ $C = 100$ for $\mu_0 = [0\ 0\ 0]$ and $\Sigma_0 = 100 I$

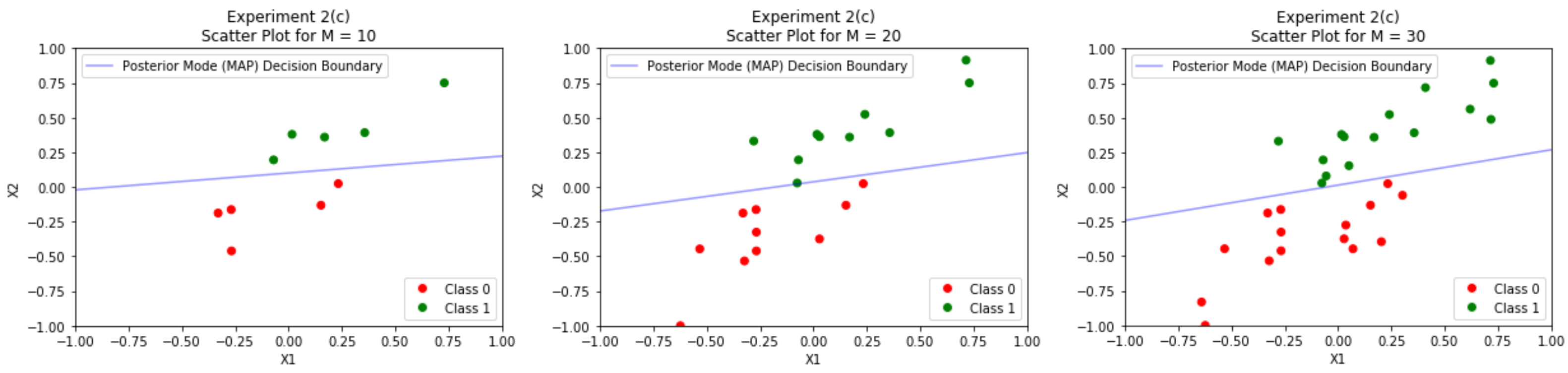
2a) Scatter plots overlaid with decision boundaries of ThetaRS for different values of M



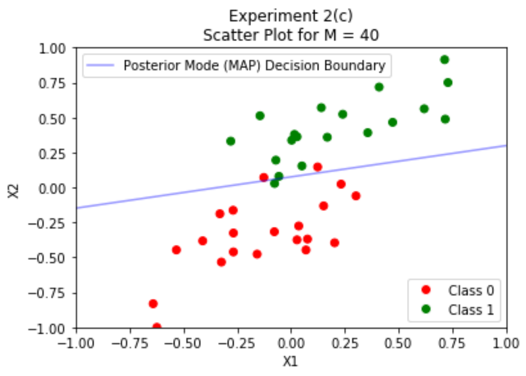
2b) Scatter plots overlaid with decision boundary of ThetaMean (posterior mean) for different values of M

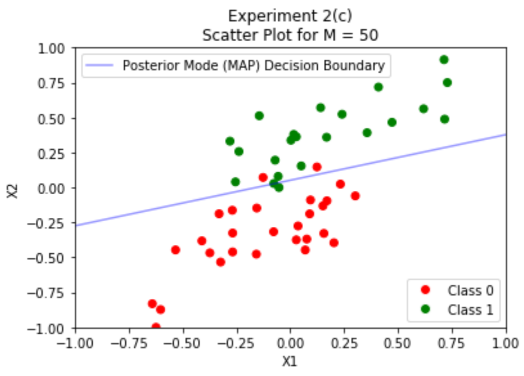


2c) Scatter plots overlaid with decision boundary of ThetaMAP for different values of M

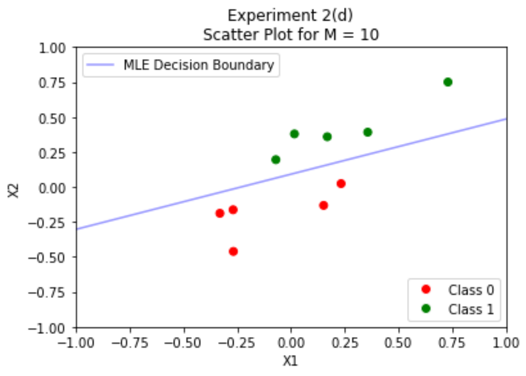


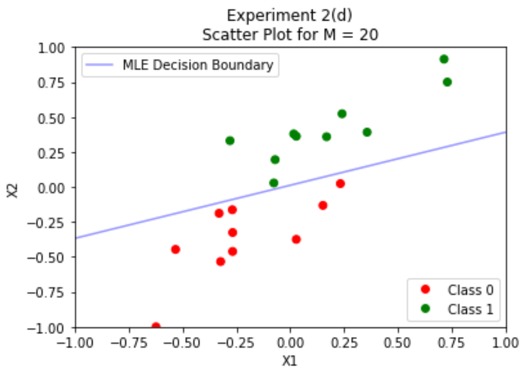
ThetaMAP was computed using scikit-learn's logistic regression code by setting C = 100 (with convergence tolerance of 1e-2). Formula of C in terms of Mu and Sigma is provided in the handwritten section of the report.

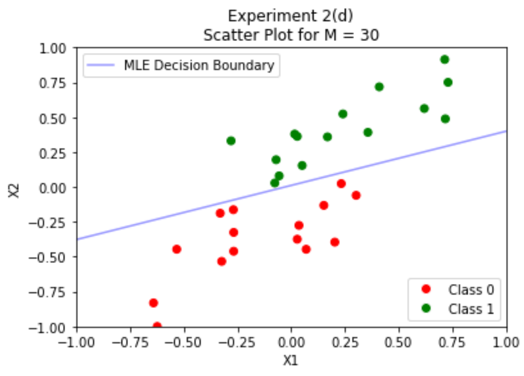


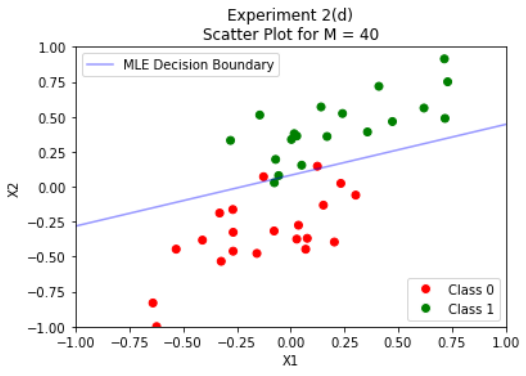


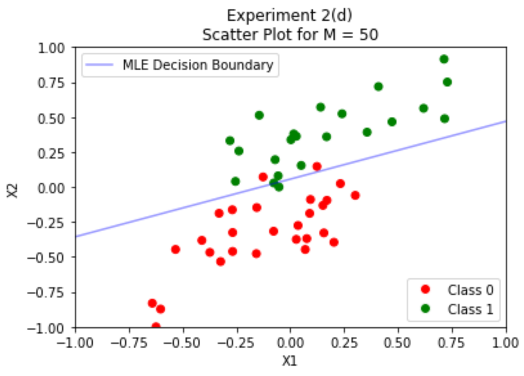
2d) Scatter plots overlaid with decision boundary of ThetaMLE for different values of M. ThetaMLE was computed using scikit-learn's logistic regression code by setting C = 1e+33 to remove influence of prior (with a convergence tolerance of 1e-2).



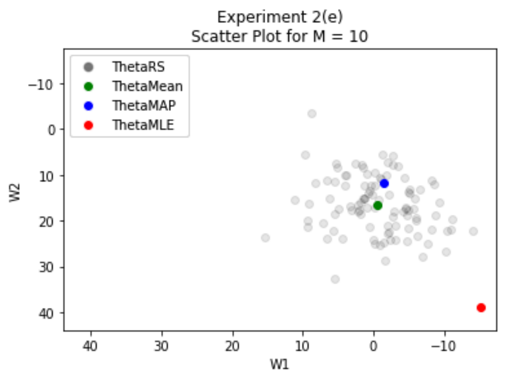


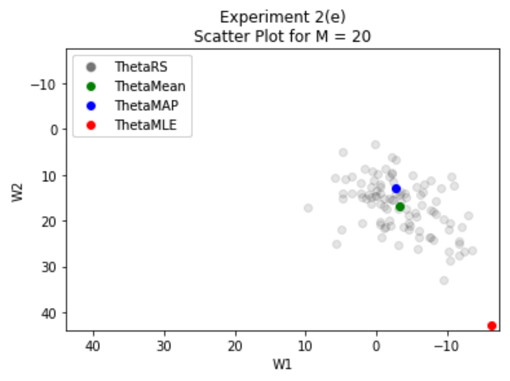


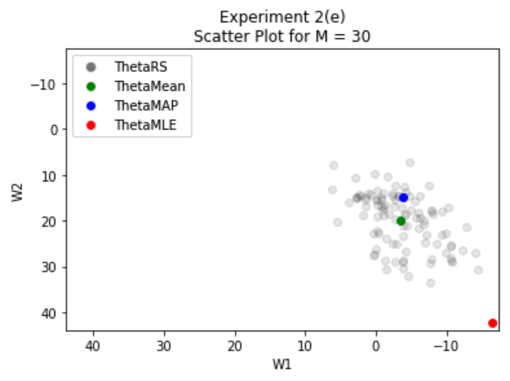


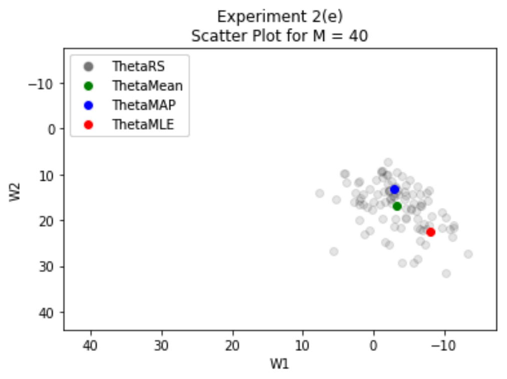


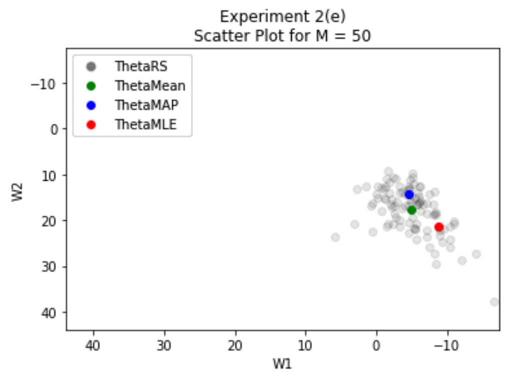
2e) Plots of weight parameters in weight space for different values of M











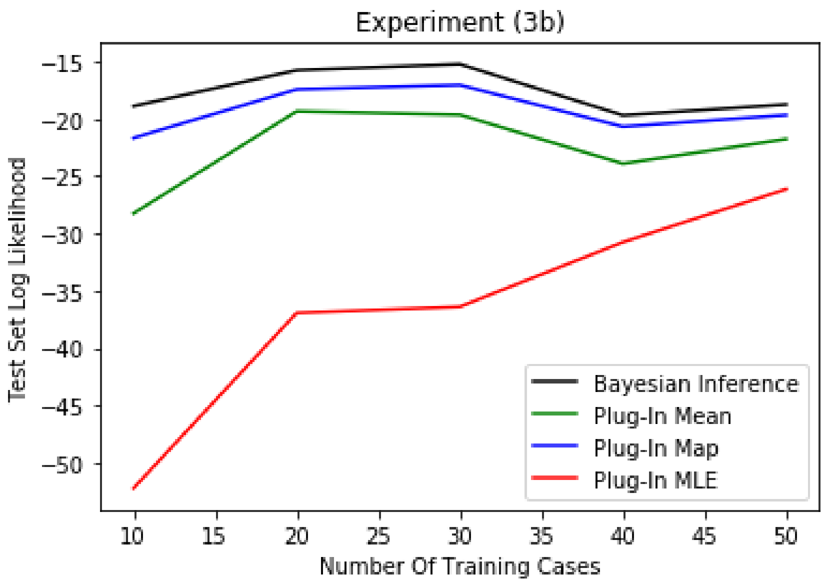
2f) Based on the visualizations, it is clear that thetaMLE tends to overfit the data when M is small. The thetaMLE decision boundary when M=10 has a steeper slope in comparison to the thetaMean decision boundary which is somewhat flat indicating that thetaMean is more conservative and does not overfit. With the gradual increase of M, we see the thetaMLE decision boundary shifting around to a greater extent (than the thetaMean decision boundary) before converging for large M=50. Based on the weight space visualization, we see that thetaMLE performs very poorly in predicting the true parameter values when M is small.

This is because thetaMean is the average of all possible parameter values represented by the parameter posterior which takes into account the prior and uncertainty on parameters. thetaMLE is a point estimate which does not have any information regarding the prior and therefore tends to overfit the data when M is small.

2g) Based on the visualizations, we see that when M is small, the decision boundary has a wider range and the scatter plot has a wider spread because there is more uncertainty on the parameters. As we increase M, the parameter posterior shrinks and concentrates its mass on the true parameter space resulting in a narrower decision boundary and a more concentrated scatter of weights in the weight space. This is because as more training data becomes available, there is lesser uncertainty on the parameters.

2h) According to theory, as the amount of training data M increases, thetaMLE, thetaMAP and thetaMean should converge. This is because parameter posterior is proportional to likelihood*prior. When the prior is flat or a large amount of training data reduces the influence of prior, parameter posterior (thereby thetaMean) and thetaMLE converge. Also, when M is sufficiently large, the concentrated parameter posterior can be approximated by a dirac delta function at the mode which is thetaMAP. From the visualizations, we see that this is indeed true. The decision boundaries for thetaMLE, thetaMAP and thetaMean converge for M=50. The weight space scatter plots help us visualize this effect more dramatically. thetaMean and thetaMAP are relatively close even for small M but thetaMLE is way off and is not even within the range of possible parameter values described by the posterior. As M increases however, we see thetaMLE, thetaMAP and thetaMean converge.

3b) Plot of test set log likelihood for Bayesian Inference, Plug-In Mean, Plug-In MAP and Plug-In MLE is provided below.



3c) The Bayesian Inference method performs best in terms of test set log likelihood. This is not surprising as the Bayesian Inference method does not discard any uncertainty in the parameters but marginalizes them away, whereas Plug-In Mean, MAP and MLE methods approximate the parameters. When the amount of training data is low, performance of Plug-In MLE suffers the most as it is highly sensitive to data and does not have information regarding the prior. Plug-In Mean and MAP fare better than MLE as they both depend on the prior. As the amount of training data increases however, the relative performance of Plug-In MLE, MAP, Mean and Bayesian Inference methods is similar as thetaMLE, thetaMAP, thetaMean and parameter posterior converge to the true parameter values.