1003243509 Agia 1
April 8, 2019

ROB313 Intro to Learning from Data - Assignment 5

Objectives

The objective of this assignment is to gain experience implementing machine learning models from the Bayesian framework. Several techniques will be used to approximate the marginal distribution and posterior distribution for the iris dataset with a Bernoulli likelihood and a Gaussian prior (logistics regression model). Specifically, the log marginal distribution will be approximated using the Laplace Approximation. The predictive posterior of the iris test set will be made with two sampling methods, namely, Importance Sampling and MCMC Metropolis-Hastings Sampling. Visualization of the posterior and predictive posterior will be used to justify any design decisions.

Code Structure and Strategies

The code structure is designed to be modular and easily understandable. Each question has a corresponding code section in the script and in the main block, which is boldly outlined with comments. To run the code for each question, the only modifications needed to be made to the file is to set the booleans corresponding to each question to True in the main block. The results of each question are printed neatly on the terminal when run, and relevant plots will be generated and saved in the current directory. The remaining code is structured as follows:

Q1: **laplace_approx(**data, rate**)**

This function computes the Laplace Approximation to the log marginal likelihood of the iris dataset. Initially, the MAP solution is found using gradient descent with the provided learning rate to optimize the MAP (Likelihood * Prior) of the combined training and validation sets. The hessian is then computed at the MAP solution, and used calculate the log marginal likelihood. This process is repeated for a set of different prior variances.

Q2: **importance_sampling(**data, map_mean, sampling_range, visual=False**)**

This function implements importance sampling around multivariate Gaussian proposal distribution centered at the map_solution obtained in Q1. Parameters for the sample size and proposal variance are selected by minimizing the negative log-likelihood of the validation set. The optimal parameters are then used to acquire the predictive posterior of the test set (using the combined training and validation set). If the *visual* parameter of the function is set to True, a plot of the approximated posterior for each component of the weight vector will be generated. The proposal distribution will also be generated for each plot to display the overlap between the posterior and the selected proposal distribution.

Q3: **metropolis_hastings(**data, map_mean, visual=False**), generate_hastings_sample(**inputs**)**

The *metropolis_hastings()* function implements metropolis-hastings sampling of a conditional multivariate Gaussian proposal distribution centered around a mean equal to the previously sampled weight vector. The initial sample is drawn from a gaussian centered at the MAP solution acquired in Q1. The variance of the proposal distribution is selected by minimizing the negative log-likelihood of the validation set, and then used to compute results on the test set (using the combined training and validation sets). The metropolis-hastings sampling algorithm is implemented in the

generate_hastings_sample() function, which returns 100 samples in accordance to the specified burn-in and thinning requirements.

The code is well commented and organized to make the code easily legible and to reduce the time spent searching for critical/error prone sections. Utility methods for frequently used functions (e.g. log_likelihood(), sigmoid(), likelihood_grad()) were written.

Q1 – Computing the Log Marginal Likelihood using the Laplace Approximation

The Laplace Approximation was used to approximate the log marginal likelihood of the combined training and validation sets for the iris dataset. The MAP solution was computed by optimizing the MAP distribution using gradient descent ($\eta = 0.001$), and the hessian was computed at the MAP solution. The gradient descent process was terminated when the maximum component of the gradient reached 10^{-9}. The log marginal distribution was then computed using the following equation:

$$\log(Pr(\boldsymbol{y}|\boldsymbol{X})) = \log(Pr(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}^*)) + \log(\Pr(\boldsymbol{w}^*)) + \frac{M}{2}\log(2\pi) - \frac{1}{2}\log(\det -\boldsymbol{H}))$$

The log marginal distribution was computed three times for prior variances of $\sigma^2 = [0.5, 1, 2]$. *Table 1* displays the results for each prior variance.

Table 1 – Number of iterations to MAP and Log Marginal Likelihood for specified prior variances

Prior Variance (σ^2)	**Iterations to MAP Solution**	**Log Marginal Likelihood**
0.5	8213	-74.82352052
1	13856	-74.82352052
2	21298	-74.81351683

The model complexity analysis is essentially the Bayesian interpretation of Occam's Razor. We can infer model complexity from the log marginal likelihoods displayed above. For a given dataset, the most complex model is typically the model with the smallest marginal likelihood, and hence, the model with the largest negative log marginal likelihood. Based on these results, the models with prior variances of 0.5 and 1 have equivalent negative marginal log-likelihoods that are larger than that of the model with prior variance 2. Thus, they must be similar in complexity and more complex than the model with prior variance of 2. The model with the smallest negative log marginal likelihood (i.e. largest marginal likelihood) has a prior variance of 2, and hence, is the least complex model.

Q2 – Importance Sampling with A Multivariate Gaussian Proposal Distribution

In this part, the predictive posterior class on each element of the test set was computed from an importance sampling approach. The following equation was used to compute the predictive posterior for a given datapoint $\boldsymbol{x}^*$:

$$\Pr(y^*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}^*) \approx \sum_{i=1}^{S} \Pr(y^*|\boldsymbol{w}^{(i)}, \boldsymbol{x}^*) \left[\frac{r(\boldsymbol{w}^{(i)})}{\sum_{j=1}^{S} r(\boldsymbol{w}^{(j)})}\right]$$

where $r(\boldsymbol{w}) = \frac{(\Pr(\boldsymbol{y}|\boldsymbol{w},\boldsymbol{X}) * \Pr(\boldsymbol{w}))}{q(\boldsymbol{w})}$ and $\boldsymbol{w} \sim q(\boldsymbol{w})$ is our proposal distribution. For the proposal distribution, a multivariate Gaussian with variance σ_p^2 centered at the MAP solution (i.e. $\boldsymbol{\mu}_p = \boldsymbol{w}_{MAP}$ for prior variance of 1) was used. The optimal sample size and proposal distribution variance parameters were selected by computing negative log-likelihood on the validation set, and choosing the parameter pair $\left(Sample\ Size_{optimal}, \sigma_{p,optimal}^2\right)$ that minimized the validation negative log-likelihood. Sample sizes on the range $[5, 10, 20, 50, 100, 500]$ were tried, and proposal variances of $[1, 2, 5, 10]$ were tried. The optimal parameters were then used to compute the predictive posterior on the test set. The results are shown in *Table 2*.

Table 2 – Predictive Posterior Results of Importance Sampling on Iris Test Set

Proposal Variance σ_p^2	**Sample Size**	**Valid (-) Log-Likelihood**	**Valid Accuracy Ratio**	**Test (-) Log-Likelihood**	**Test Accuracy Ratio**
2	100	14.685	0.710	7.122	0.733

The results show considerate accuracy on both the validation set and the test set. The difference in the log-likelihood between the validation set and the test set are due to the number of datapoints in each set. The validation set has 31 datapoints, while the test set has 15 datapoints; which explains why the validation log-likelihood is approximately double the test log-likelihood with similar accuracy ratios.

The accuracy of the proposal distribution can be visualized by an overlapping with the approximate posterior. The posterior can be approximated by sampling $\boldsymbol{w}^{(i)} \sim q(\boldsymbol{w})$, and plotting the results of $\frac{r(\boldsymbol{w}^{(i)})}{\sum_{j=1}^{S} r(\boldsymbol{w}^{(j)})}$ for $i = 1, \ldots, 5000$ (i.e. 5000 samples for effective visualization). Technically, this does approximate the posterior due to the following approximate equality:

$$\Pr(y^*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}^*) = \int \Pr\left(y^*\middle|\boldsymbol{w}^{(i)}, \boldsymbol{x}^*\right) \Pr(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) d\boldsymbol{w} \approx \sum_{i=1}^{S} \Pr(y^*|\boldsymbol{w}^{(i)}, \boldsymbol{x}^*) \left[\frac{r\left(\boldsymbol{w}^{(i)}\right)}{\sum_{j=1}^{S} r(\boldsymbol{w}^{(j)})}\right]$$

Notice that the Monte Carlo sampler simply approximates the predictive posterior integral. Due to symmetry in the equations, it is clear that the posterior, $\Pr(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})$, is approximated by the $\frac{r(\boldsymbol{w}^{(i)})}{\sum_{j=1}^{S} r(\boldsymbol{w}^{(j)})}$ term in the Monte Carlo sampler equation. *Figures1-5* show the overlap between posterior and proposal distribution for each component of the weight vector $\boldsymbol{w} \in \mathbb{R}^{\mathbf{5}}$.

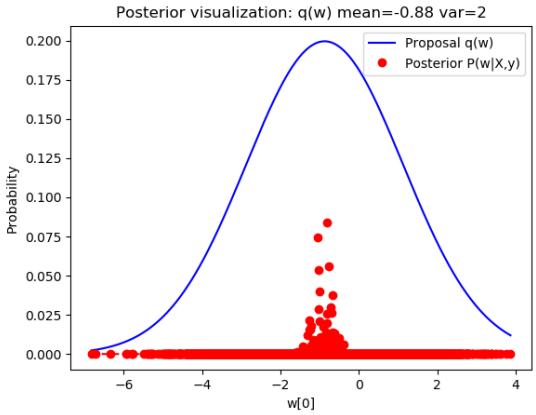

Figure 1 – Posterior and Proposal for $w^{(1)}$

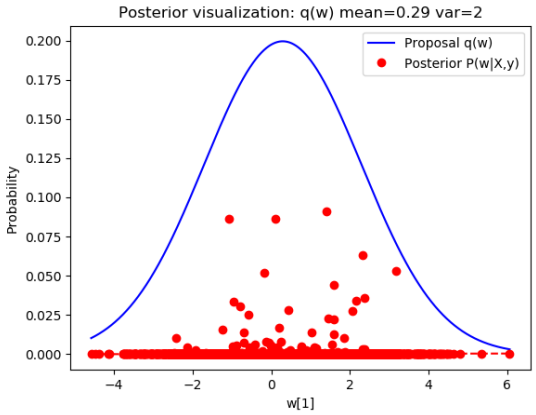

Figure 2 – Posterior and Proposal for $w^{(2)}$

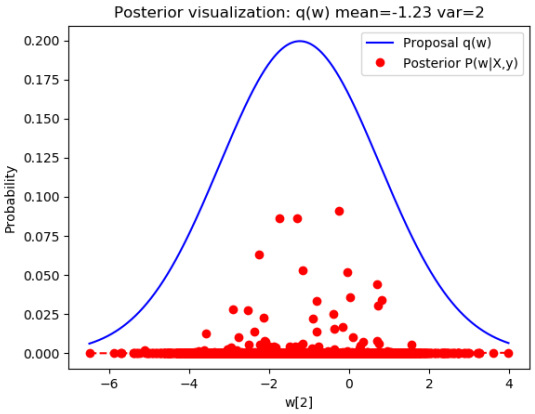

Figure 3 – Posterior and Proposal for $w^{(3)}$

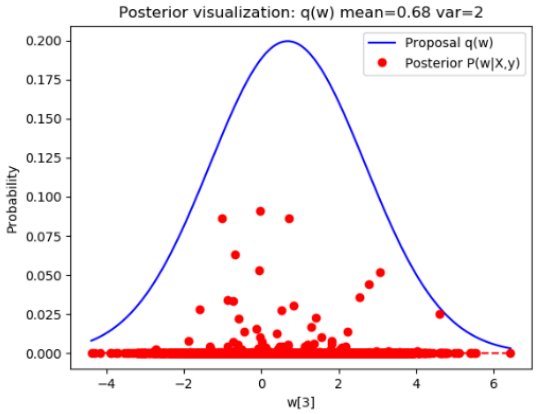

Figure 4 – Posterior and Proposal for $w^{(4)}$

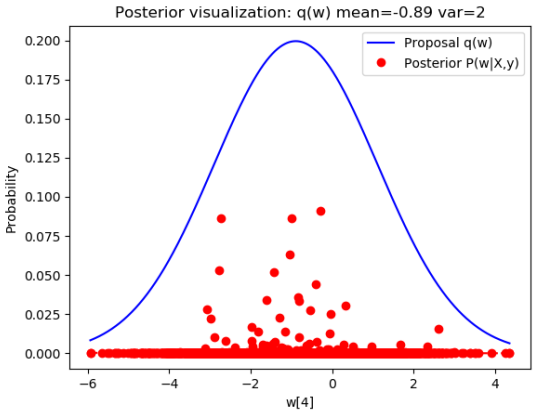

Figure 5 – Posterior and Proposal for $w^{(5)}$

These figures illustrate the overlap between the approximate posterior distribution and the proposal distribution for each component of the sampled weight vector. As desired, there is significant overlap between the posterior and the proposal in areas of high probability mass, which yields the reasonably accurate results shown in *Table 2*.

Q3 – Metropolis-Hastings MCMC Sampler with Conditional Gaussian Proposal Distribution

A MCMC Metropolis-Hastings sampler was implemented to approximate the predictive posterior for the elements of the iris test set. Similar to Q2, parameter selected for the proposal distribution variance was selected as the variance the minimized the negative log-likelihood on the validation set. A sample size of 100 was used to make predictions, with thinning by sampling every 100 iterations after a burn-in of 1000 iterations. A conditional multivariate Gaussian proposal distribution was used to draw samples. The results are summarized in *Table 3*.

Table 3 – Predictive Posterior Results of Metropolis-Hastings MCMC Sampler on Iris Test Set

Proposal Variance σ_p^2	**Sample Size**	**Valid (-) Log-Likelihood**	**Valid Accuracy Ratio**	**Test (-) Log-Likelihood**	**Test Accuracy Ratio**
2	100	15.011	0.677	7.215	0.733

We can see that the results of this particular run of the MCMC sampler was very similar to the results obtained with the importance sampling implementation. Also, notice that the preferred variance of the proposal distribution is $\sigma_p^2 = 2$, which is the same preferred variance seen for importance sampling,

A histogram of the predictive posterior class-conditional probability samples ($\Pr(y^* = 1|\boldsymbol{x}^*, \boldsymbol{w}^{(i)})$ for $i = [1, 100]$) for flowers 9 and 10 of the iris test set are shown in *Figure 6* and *Figure 7*.

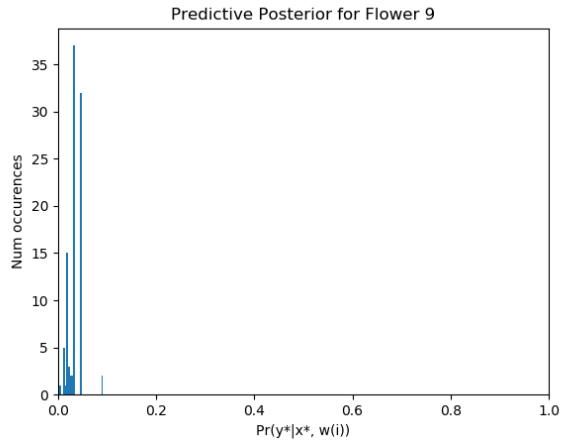

Figure 6 – Flower 9 Predictive Posterior

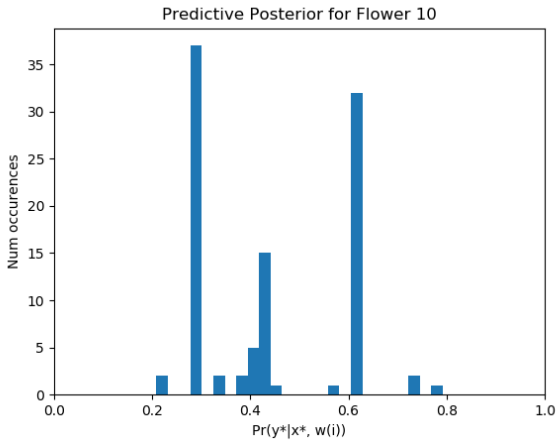

Figure 7 – Flower 10 Predictive Posterior

Both histograms illustrate a distribution of the predictive posterior over 100 MCMC samples. In *Figure 6*, we see that the class-conditional predictive posterior for all 100 samples suggests that flower 9 is most likely not class 1. In *Figure 7*, the results of 100 samples show more of a split decision for the class of flower 10, with a slight skew towards not class 1.

The benefit of using the Bayesian approach is that we can visualize the class-conditional predictive posterior of each flower as a distribution that is dependent on the samples collected. In contrast, since sampling is not used in a frequentist approach, composing a histogram for the class predictions would not make sense. This is because all 100 predictions would result in the same class-conditional probability, failing to provide any additional information on alternate potential classifications. This may not be problematic for a datapoint such as flower 9, where all samples

from the Bayesian approach predict class 0 with extremely high certainty, and thus a frequentist approach would yield a comparable result. But take for instance a datapoint where a frequentist approach yields a class conditional probability of 0.5 (perhaps for flower 10). In this case, the frequentist approach will be unable to make an inference on the flower type, whereas a Bayesian sampling approach may be able to provide a class-conditional predictive posterior distribution that is skewed slightly more to one class. This indicates how the Bayesian approach may provide deeper insight on the data we are analyzing, in comparison to a frequentist approach.

Q4 – Research Paper Report: Practical Aspects of Machine Learning

Here I present a critical review of *"*A Few Useful Thing to Know about Machine Learning" by Pedro Domingos from Department of Computer Science at the University of Washington.

This paper was written to highlight some key details and present general tips that are commonly used by machine learning practitioners to write a successful machine learning algorithms for classification in a timely manner. He begins by decomposing the classifier into three primary components: representation, evaluation, and optimization (i.e. model type, evaluation function, optimization techniques). With these components, the main goal is laid out to be the design of a machine learning model that is best capable of generalizing to new data with low variance and bias. The remainder of the research paper is comprised of sections that describes several common challenges faced when designing a well performing classifier, and various techniques that one may use to remedy them.

The first statement, "Data Alone is Not Enough", makes the claim that a successful machine learning model must embody knowledge or assumptions beyond the data it is trained on to be able to generalize effectively. Ideally, we would like a machine learning model to demonstrate the ability to reason by inductance. In doing so, the model should be able to generate large output knowledge from little input knowledge (i.e. generalize through induction). The authors argument makes a clear distinction between the data being fed into the model, and the embodiment of some inherent knowledge in the model. This is problematic, as it is easy to see that a model without any data has no knowledge. In this regard, the author fails to recognize that the embodiment of knowledge in the model is actually transferred from data it is being fed (e.g. where the knowledge is initially embedded). By leaving this key fact out, the author undermines the importance of well processed, clean data that allows for this knowledge transfer from the data the machine learning model to occur seamlessly.

After briefly discussing the challenge of overfitting and presenting several techniques including cross-validation and regularization, the author moves on to "The Curse of Dimensionality". In this section, a case is made that aside from the computational struggles of high dimensional data, the general issue with working in high dimensions is due to a lack of intuition. Because we live in a three-dimensional world, we fail to thoroughly understand the geometric properties of shapes (e.g. distributions, hyperplanes) in high dimensions. Although the author makes an interesting argument, it may have been useful to justify his claim by focusing on exactly how our spatial intuition goes into design and tuning of machine learning algorithms operating in lower dimensions. This section is concluded by introducing the Blessing of Non-Uniformity – a special phenomenon stating that most examples are not spread throughout the instance space, but are instead concentrated near a

lower-dimensional manifold. And machine learning models may be able to capitalize on this inherently, or through the use of dimensionality reduction methods.

The following section, “Theoretical Guarantees are Not What They Seem”, takes a confusing stance on the practicality of theoretical results shown in modern research papers. The author begins by stating that because inductive reasoning used to make theoretical bounding or asymptotic guarantees, they are to be interpreted more so as educated predictions (i.e. very loose guarantees). After stating that theory is not so useful in practical applications, the author goes on to accredit the modern advancement of machine learning to the interplay between theory and practice, a clear contradiction. He also seems to place the blame on the theory, instead of the misinterpretation of the theory by the users, for incorrect design decisions made based on irrational conclusions.

In the remaining sections, the author presents some well formulated discussions related the importance of feature engineering, more data in contrast to algorithm complexity, and provides some warning about incorrect implications commonly made by machine learning practitioners. Most of the claims made were well supported (with exception of a few), and the author did well include supporting examples where possible. Overall, I believe the paper successfully summarizes the key challenges one may face when designing machine learning classifier in a concise and easily understandable manner.