Table of Contents

1. Introduction

In machine learning, linear regression is used for finding relationship between a target variable and one or more predictors. In this report, we compare and contrast two such methods of linear regression, namely the simple linear regression model and the Bayesian linear regression model.

Section 2 discusses the basic terminologies and background required for understanding linear regression while sections 3 and 4 take up simple linear regression and Bayesian linear regression respectively. Finally, section 5 summarizes the differences between the aforementioned two models.

2. Basic Terminologies and Background

2.1 Definitions and Terminologies

The following terminologies are used throughout the rest of the paper.

1. d-dimensional column vectors will be represented with lowercase bold typeface characters like **x**, **w** etc.
2. Matrices will be denoted by uppercase bold typeface characters like **A**, **B** etc.
3. The data is denoted by *D* and comprises of n points, [($\mathbf{x_1}$, y_1), ($\mathbf{x_2}$, y_2), …, ($\mathbf{x_n}$, y_n)]. Here, **x** represents the predictor vector while y is the target variable.
4. The predicted value is denoted by t and the parameter space is denoted by Θ.

2.2 Gaussian Distribution

In probability theory, the gaussian distribution is a very common continuous probability distribution. A univariate gaussian distribution is defined by 2 parameters: mean (μ) and precision (p). Its closed form is given by:

$$p(x) = \frac{\sqrt{p}}{\sqrt{2\pi}} exp(\frac{p}{-2}(x - \mu)^2)$$

In shorthand, we write p(x) with mean μ and precision p as:

$$N(x|\mu, p^{-1})$$

A multivariate gaussian distribution is a generalization of the univariate gaussian distribution to higher dimensions. A gaussian distribution in d

variables is defined by 2 parameters: a mean (**m**) of size d and a precision matrix (**P**) of size *d*d*. For independent dimensions, its closed form is given by:

$$p(x) = \frac{1}{2\pi^{\frac{d}{2}}|P^{-1}|^{\frac{1}{2}}} exp(-\frac{1}{2}(x - m)^T P(x - m))$$

In shorthand, we write p(x) as:

$$N(x|m, P^{-1})$$

Now, we simplify the above expression.

We know $p(x) \propto exp(-\frac{1}{2}\underbrace{(x-m)^T P(x-m)}_{K})$

Simplifying K → $K = (x-m)^T P(x-m)$

$= (x^T - m^T) P(x-m)$

$= x^T Px - x^T Pm - m^T Px + m^T Pm$

Since each term in the above expression is a scalar,

$x^T Pm = (x^T Pm)^T = m^T Px$

$\Rightarrow K = x^T Px - 2x^T Pm + m^T Pm$

Quadratic in x — Linear in x — Constant in x

Substituting this value of k and removing the constant term from proportionality, we get,

$$p(x) \propto exp(-\frac{1}{2}(x^T Px - 2x^T Pm)) \quad (2.1)$$

3. Simple Linear Regression

In linear regression, we try to predict a target function in **x** by modelling it to be linear in weights **w**. By using square error as the loss function and minimizing it, our target function comes out to be:

$$t = w^T x$$

Here, **w** is the parameter vector to our learning model. The model needs to learn **w** for it to predict values of y given **x**.

For this, we assume that for each given datapoint $\mathbf{x}_i$, y_i was picked randomly from the distribution:

$$Y = w^T x + \in$$

Where, $\in$ is the gaussian noise, with mean 0 and precision a.

Therefore, each y_i can be modelled as an independent random variable with a gaussian distribution like:

$$p(x_i, w) = N(w^T x, a^{-1})$$

3.1 MLE Estimate for **w**

In simple linear regression, we follow the maximum likelihood approach for estimating **w**, and therefore choose the **w** that maximizes p(D|**w**) i.e.

$$w_{MLE} = argmax_w(p(D|w))$$

Now,

$$p(D|\omega) = p(y_1, y_2, y_3 \dots, y_n | x_1, x_2 \dots, x_n, \omega)$$

Since each y_i is an independent observation,

$$p(D|\omega) = \prod_{i=1}^{n} p(y_i | x_1, x_2 \dots, x_n, \omega)$$

Also, y_i is conditionally independent of all x_j, $i \neq j$

$$\therefore P(D|\omega) = \prod_{i=1}^{n} p(y_i | x_i, \omega)$$

Using eqn 3.1, $P(D|\omega) = \prod_{i=1}^{n} \frac{\sqrt{a}}{\sqrt{2\pi}} \exp\left(-\frac{a}{2}(y_i - \omega^T x_i)^2\right)$

$$\Rightarrow P(D|\omega) = \frac{a^{n/2}}{(2\pi)^{n/2}} \exp\left(-\frac{a}{2}\sum_{i=1}^{n}(y_i - \omega^T x_i)^2\right)$$

↳ Since this is a scalar, $\omega^T x_i = x_i^T \omega$

Let $A = \begin{bmatrix} -- x_1^T -- \\ -- x_2^T -- \\ \vdots \end{bmatrix}$ & $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix}$

Simplifying, we get: $p(D|\omega) = \frac{a^{n/2}}{(2\pi)^{n/2}} \exp\left[-\frac{a}{2}\underbrace{(y - A\omega)^T(y - A\omega)}_{K}\right]$ — 3.2

Now, $K = (y - A\omega)^T(y - A\omega) = y^T y - 2\omega^T A^T y + \omega^T A^T A\omega$

$$\Rightarrow P(D|\omega) \propto \exp\left[-\frac{1}{2}(\omega^T A^T A\omega a - 2\omega^T A^T y a)\right]$$

Comparing the eqn. with 2.1, $P = aA^TA$ & $P m = A^T y a$

$$\Rightarrow m = P^{-1}A^T y a$$

$$\Rightarrow P(D|\omega) = N(D|(A^TA)^{-1}A^T y, (a A^T A)^{-1}) \quad \text{— 3.3}$$

Since 3.3 is a gaussian distribution, its maximum occurs at the mean. Therefore, our MLE estimate for **w** is:

$$w = (A^T A)^{-1} A^T y \tag{3.4}$$

Here, A is called the design matrix. Consequently, our target variable t is predicted by:

$$t = ((A^T A)^{-1} A^T y)^T x \tag{3.5}$$

4. Bayesian Linear Regression

In the Bayesian viewpoint, we formulate linear regression using probability distributions rather than point estimates. Like simple linear regression, the output y is assumed to be drawn from the probability distribution:

$$y = N\left(w^T x,\ a^{-1}\right) \tag{4.1}$$

The aim of Bayesian Linear Regression is not to find the single "best" value of the model parameters, but rather to determine the posterior distribution for the model parameters. Therefore, not only is the response generated from a probability distribution, but the model parameters are assumed to come from a distribution as well. Here, we assume that **w** comes from the distribution:

$$w = N(0,\ (bI)^{-1}) \tag{4.2}$$

Where, ***I*** is the identity matrix. The above distribution is the *prior* for **w**.

According to Bayes' Theorem,

$$Posterior = \frac{Likelihood*Prior}{Normalization}$$

Since normalization is a constant factor,

$$Posterior \propto Likelihood * Prior \tag{4.3}$$

4.1 MAP estimate of **w**

In Bayesian linear regression, we follow the *maximum aposteriori* approach for estimating **w**, and therefore choose the **w** that maximizes p(**w**|D) i.e.

$$w_{MAP} = argmax_w(p(w|D))$$

According to eqn. 4.3,

$$p(w|D) \propto p(D|w) \times p(w)$$

Substituting the values of $p(D|w)$ from 3.2 & $p(w)$ from 4.2,

$$p(w|D) \propto \exp\left[\underbrace{-\frac{a}{2}(y-Aw)^T(y-Aw)-\frac{b}{2}w^Tw}_{K}\right]$$

$$K \propto a(y-Aw)^T(y-Aw)-bw^Tw$$
$$= ay^Ty - 2aw^TA^Ty + w^T(A^TAa+bI)w$$

$$\Rightarrow p(w|D) \propto \exp\left[-\frac{1}{2}\left[w^T(aA^TA+bI)w-2aw^TA^Ty\right]\right]$$

Comparing with 2.1, we get:

$$P_1 = aA^TA+bI \quad \& \quad m_1 = (A^TA+\frac{b}{a}I)^{-1}A^Ty$$

$$\& \quad p(w|D) = N(w|m_1, P_1^{-1}) \quad \text{---} \quad 4.4$$

Since, p(**w**|D) is a gaussian distribution, its maximum lies at its mean. Therefore, our MAP estimate for **w** is:

$$w = (A^TA + \frac{b}{a}I)^{-1}A^Ty \tag{4.5}$$

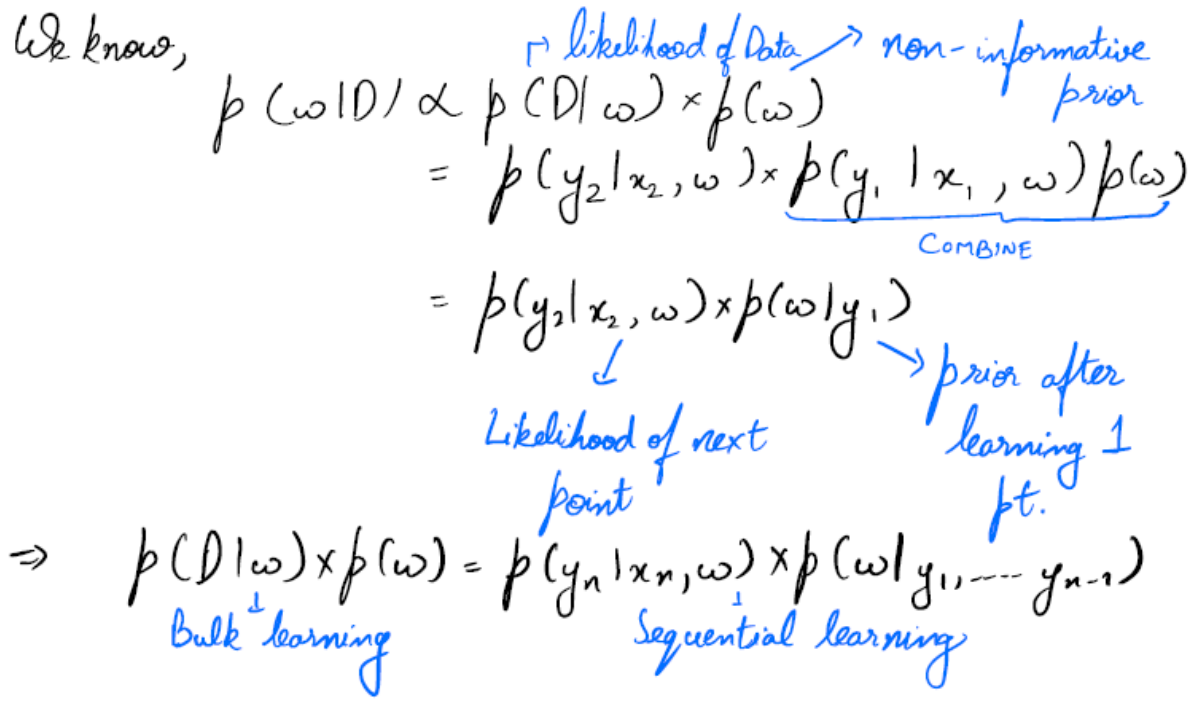
Here, $\frac{b}{a}$ is called the regularization constant. This estimate is equivalent to simple linear regression with regularization and prevents the model from overfitting. However, in simple linear regression, the estimate for the target variable is a single point value given by:

$$t = w^Tx$$

Whereas, in Bayesian Linear Regression we are interested in finding the predictive posterior distribution of t, P(t|**x**, D).

4.1.1 Bulk vs Sequential Learning: Updating Priors

In Bayesian linear regression too, like simple linear regression, sequential learning and bulk learning yield the same results. This can be shown by the following proof.



Therefore, we can treat sequential learning Bayesian Linear Regression as learning one point, where the prior for **w** (which is now an informative prior) provides the rest of the information for calculating the posterior distribution of **w**.

4.1.2 Effect of Priors on Bayesian Linear Regression

Figure 1 demonstrates sequential learning in Bayesian Linear Regression for approximating a straight-line function, y = -0.3 + 0.5x. The first column gives the likelihood distribution of weights given the next observation ($P(y_i|x_i,\mathbf{w})$), and the second column represents the prior distribution of w ($P(\mathbf{w}|y_1, \ldots, y_{i-1})$). Finally, for plotting the last column, w_0 and w_1 are picked from the posterior distribution of **w** (obtained by multiplying the likelihood and the prior distribution in that row) and the corresponding probable target functions are plotted on the xy-plane. Note that the target functions are probable because w_0 and w_1 are sampled from a gaussian distribution.

We can see, as the bayesian model learns, the prior becomes more and more informative as it confines itself to closer to the actual point with each new data point. Row 4 shows the result of learning with 20 data points, where the posterior distribution has become very compact.

In the limit of infinite points, the posterior distribution would become a delta function centered on the true parameter values. At this point, the prior for **w** is a single value, and its distribution's precision becomes ∞. This leads the target function for bayesian linear regression to be exactly similar to that of simple linear regression.

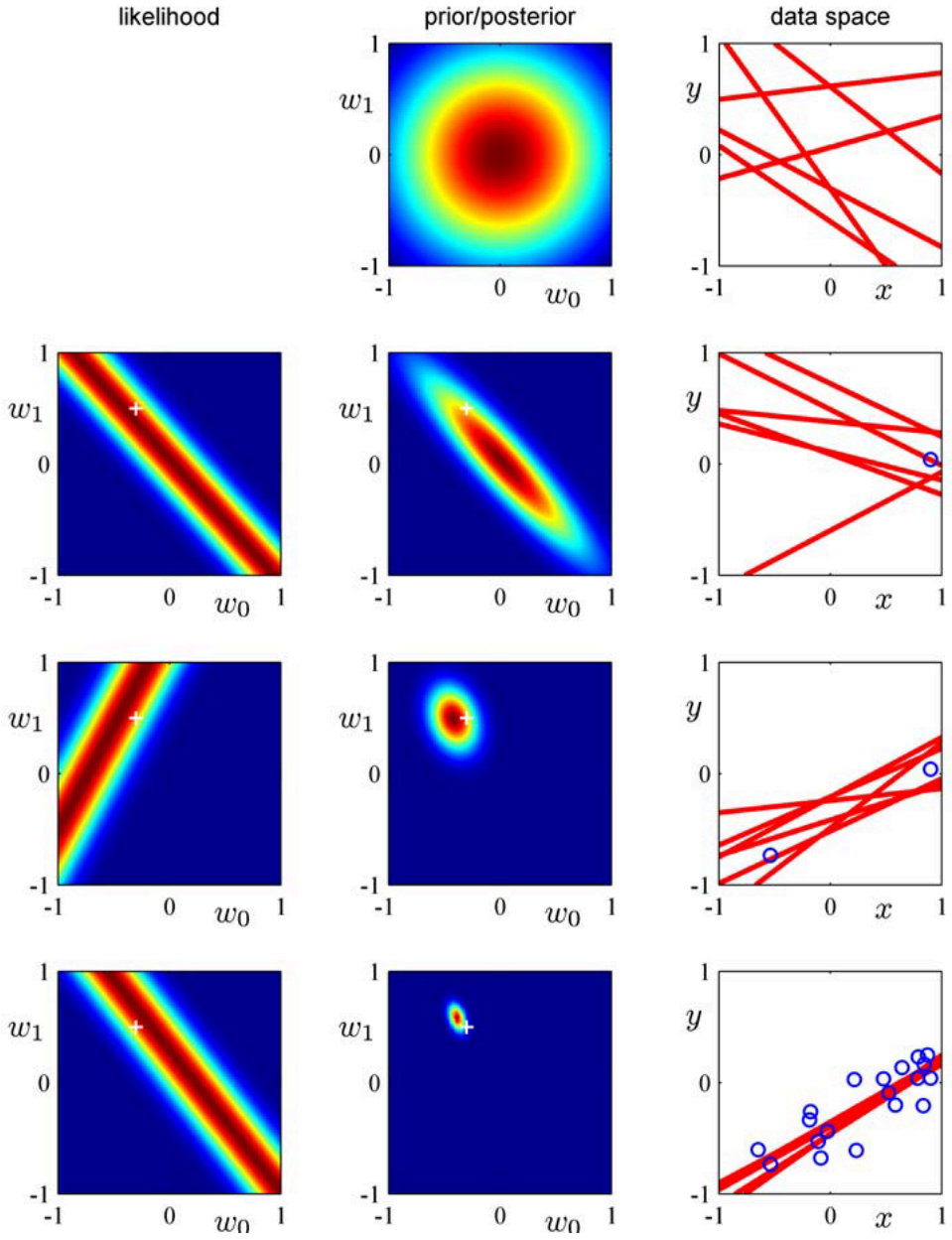


Figure 1: Illustration of sequential Bayesian learning for a simple linear model of the form $y(x, w) = w_0 + w_1 x$.

4.2 Posterior Distribution for Prediction

As stated earlier, in Bayesian linear regression, the target output is not a single point but a distribution. This requires us to calculate the predictive distribution of the target variable ($P(t|\mathbf{x}, D)$).

$$P(t|x,D) = \int p(t|x,D,\omega)\, p(\omega|x,D)\, d\omega$$

$$= \int \mathcal{N}(t|\omega^T x, a^{-1})\, \mathcal{N}(\omega|m_1, P_1^{-1})\, d\omega$$

$$\propto \int \exp\left[\underbrace{-\frac{a}{2}(t-\omega^T x)^2 - \frac{1}{2}(\omega-m_1)^T P_1 (\omega-m_1)}_{-2k}\right] d\omega$$

$$k = \omega^T \underbrace{(a x x^T + P_1)}_{P_2}\omega - 2\omega^T \underbrace{(x y a + P_1 m_1)}_{P_2 m_2} + a t^2$$

$$\Rightarrow k = \omega^T P_2 \omega - 2\omega^T P_2 m_2 + a t^2 + m_2^T P_2 m_2 - m_2^T P_2 m_2$$

$$\Rightarrow k = (\omega - m_2)^T P_2 (\omega - m_2) - m_2^T P_2 m_2 + a t^2$$

↳ Gaussian in ω

Finally,

$$p(t|x,D) \propto \int \mathcal{N}(\omega|m_2, P_2^{-1}) \exp\left[-\frac{1}{2}(a t^2 - m_2^T P m_2)\right] d\omega$$

Const in ω.

$$\propto \exp\left[-\frac{1}{2}\underbrace{(a t^2 - m_2^T P_2 m_2)}_{k}\right]$$

Simplifying k as done in previous proofs, we get →

$$p(t|x,D) \propto \exp\left[-\frac{1}{2}\left[\begin{matrix}(a-a^2x^TP_2^{-1}x)t^2 \\ -2(ax^TP_2P_1^{1}m_1)t\end{matrix}\right]\right]$$

$\therefore P(t|x,D)$ is gaussian in t, with

$$P_3 = a(1-ax^TP_2^{-1}x) \quad \& \quad m_3 = \frac{1}{P_3}ax^TP_2^{-1}P_1m_1$$

On substituting the values of m_1, m_2, P_1, P_2 & P_3, we get →

$$m_3 = m_1^Tx \quad \& \quad \frac{1}{P_3} = \sigma_3^2 = \frac{1}{a}\left(1+x^T(A^TA+\frac{b}{a}I)^{-1}x\right)$$

$$\& \quad p(t|x,D) = N(t|m_3,\sigma_3^2)$$

And hence, our estimate for t has a mean at:

$$t = ((A^TA + \frac{b}{a}I)^{-1}A^Ty)^Tx \qquad (4.6)$$

Note that the equation above corresponds to the MAP estimate of **w**. This implies, that the predicted value has the maximum probability to be at our MAP estimate of **w** (which should intuitively be so). Finally, the variance of the distribution is given by:

$$var = \frac{1}{a}(1 + x^T(A^TA + \frac{b}{a}I)^{-1}x) \qquad (4.7)$$

4.3 Demonstrating the effect of predictive distribution

As an Illustration, Figure 2 fits a Bayesian linear regression model to datasets of various sizes and plots the corresponding posterior distributions for each x. Here, the green curves correspond to the function sin(2πx) from which the data points were generated. Data sets of size 1, 2, 4 and 25 are shown in the four plots by the blue circles. The red curve denotes the mean of the corresponding gaussian predictive distribution and the pink shaded region

spans 1 standard deviation to either side of the mean for a 9-dimensional model.

Note that the predictive uncertainty depends on x and is smallest in the neighbourhood of the data points. Also note that the level of uncertainty decreases as more data points are observed. Therefore, for points that lie far away from a cluster of data-points, the uncertainty rises, and Bayesian linear regression allows us to effectively say “I don’t know” for the prediction of t at such points as the standard deviation is too high.

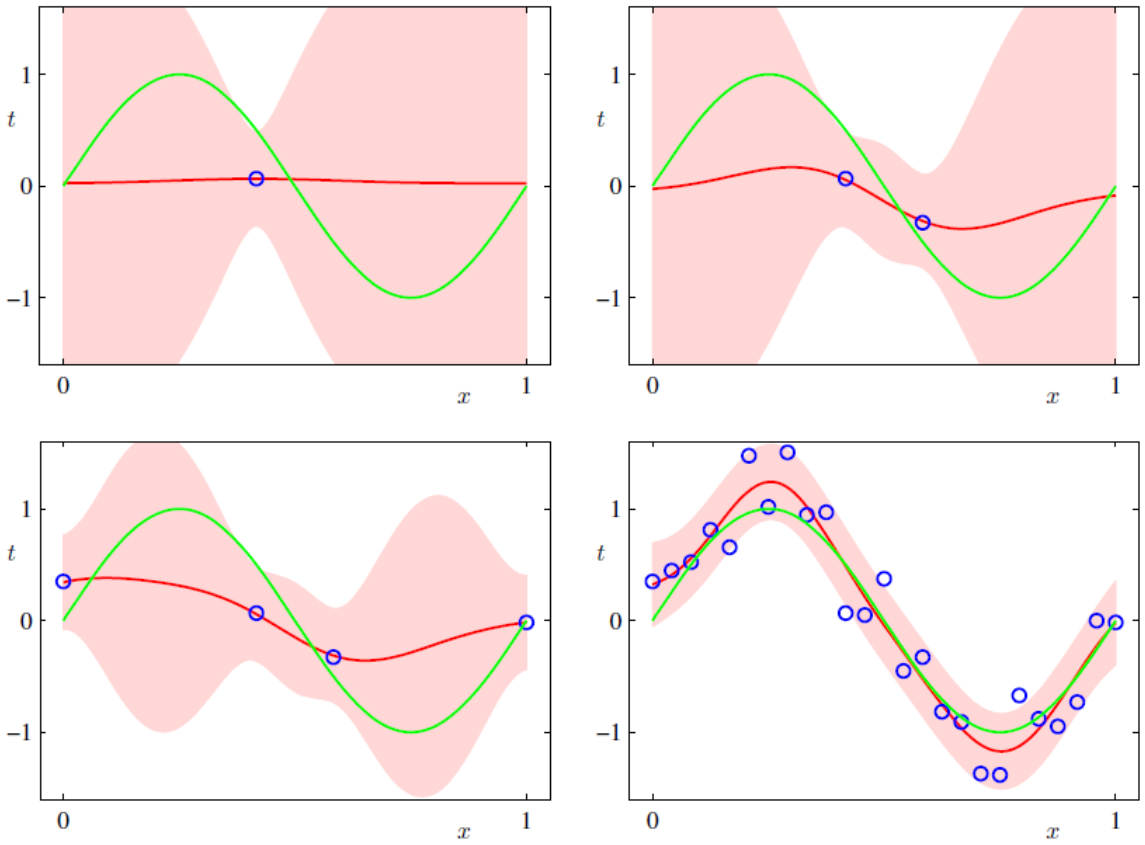


Figure 2: Examples of predictive distribution for a Bayesian linear regression model to approximate sin(2πx).

5. Comparing Simple and Bayesian Linear Regression

Simple LR	Bayesian LR	Reference
• Gives a point estimate of the target variable.	• Gives a probability distribution for the target variable.	**3, 4, 4.3**

• Uses MLE to estimate the values of parameters.	• Uses MAP to estimate the distribution of parameters.	**3.1, 4.1**
• It is prone to overfitting in complex models.	• Prevents overfitting in complex models by implicitly adding a regularization parameter.	**4.1**
• Complexity of the model needs to be limited according to number of datapoints available.	• Due to the availability of priors, the complexity of the model can be decided only by the complexity of the problem being solved. An increase in data washes out the effect of priors.	**4.1.1, 4.1.2**
• It does not report any loss of confidence when extrapolating data points.	• As data points are extrapolated, loss of confidence is given by standard deviation of the predicted distribution.	**3.1, 4.3, 4.4**

6. Experimental Results

For comparing linear and Bayesian regression, we created two regression models which predict the number of calories burnt given the age, height, weight, duration of exercise, heart rate and the body temperature of a person. As mentioned before, simple linear regression gives a single point estimate which can be interpreted as the MLE estimate given the data while the Bayesian regression model gives the distribution (mean and standard deviation) of the target variable. We can use the mean as the MAP estimate.

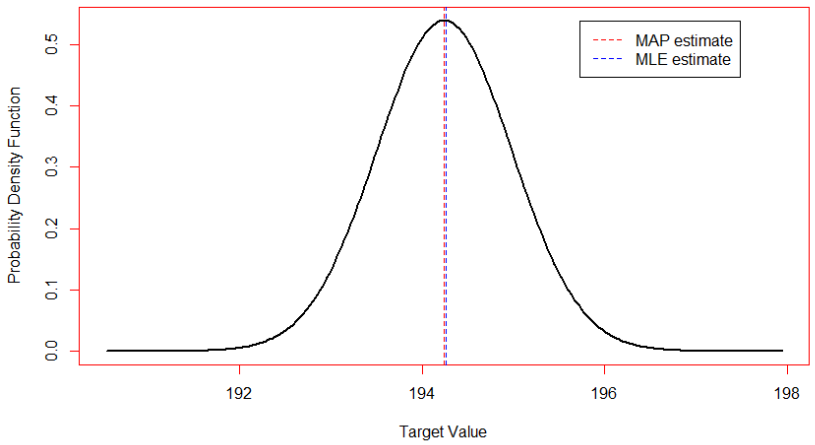


Figure 3: Feeding points that are near the training data while testing leads to low standard deviation in MAP approach.

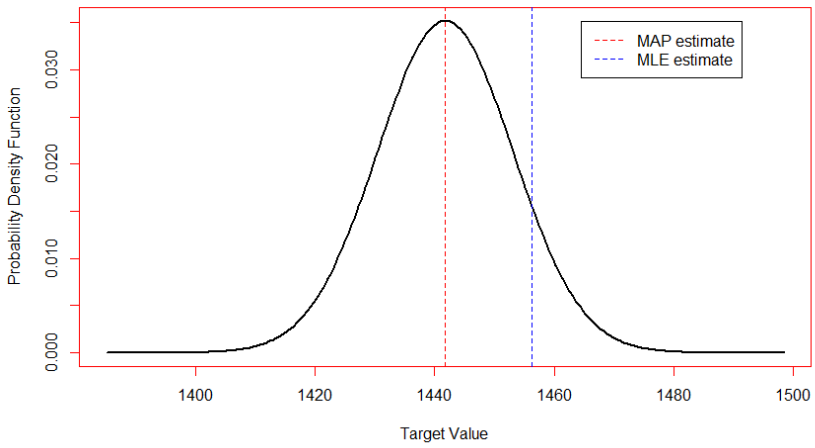


Figure 4: Extrapolation in MAP and MLE when only 10 data points are presented. High standard deviation in MAP indicates that the estimate is probably not a good one. MLE gives an answer without any indication of standard deviation

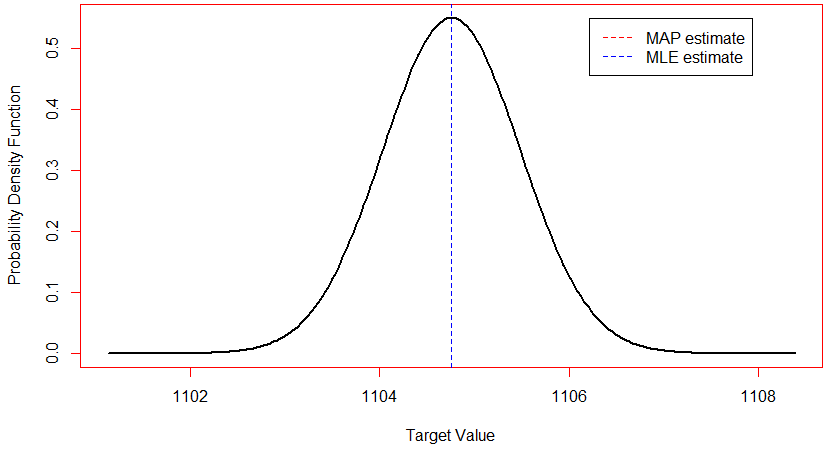


Figure 5: On giving 15,000 training points, MAP and MLE estimates by the two models perfectly coincide due to the priors being washed out.

For demonstrating the predictive uncertainty of the Bayesian regression model, we input test data points which were close enough to those in the training data. The Bayesian model output was almost same as the simple model output and the standard deviation output by the Bayesian model was very low (close to 0.2). But when we input a data point which was far away from the cluster of data points, the standard deviation of the output given by the Bayesian model was quite high while the simple model just gave a single point estimate through extrapolation. The above result illustrates one of the biggest downsides of using simple linear regression. Since the prediction made by the simple model does not report any loss of confidence when extrapolating data points, the estimate has more chances of being wrong.

Bayesian regression model also performs better when the size of the dataset is small because of the availability of priors. Simple linear regression may overfit the data while Bayesian model incorporates regularization and expresses the estimate as a distribution of possible values. As the number of data points increases, the likelihood washes out the priors, and thus, when data was increased to 15000 points, the output of the Bayesian model converged to that of simple linear regression. This property is illustrated in the below table.

7. References

[1] M. Monk, “mathematicalmonk,” Youtube, 2011. [Online]. Available: https://www.youtube.com/user/mathematicalmonk.

[2] W. Koehrsen, “Towards Data Science,” Medium, 14 April 2018. [Online]. Available: https://towardsdatascience.com/introduction-to-bayesian-linear-regression-e66e60791ea7.

[3] C. Bishop, “Linear Models for Regression,” in *Pattern Recognition and Machine Learning*, Springer, 2006, pp. 137-173.

[4] C. B. Do, “The Multivariate Gaussian Distribution,” *Lecture on Machine Learning,* 2008.

[5] N. Goel, Introduction to ML, 2018.

[6] N. Goel, Polynomial Curve Fitting, 2018.