COMP 6321 Machine Learning Assignment

Qingbo Kang, Concordia University 40058122

1 L1 vs. L2 Regularization

1.1 a

Figure 1 shows the RMSE (root mean squared error) on the training set and the test set, as a function of the regularization parameter λ.

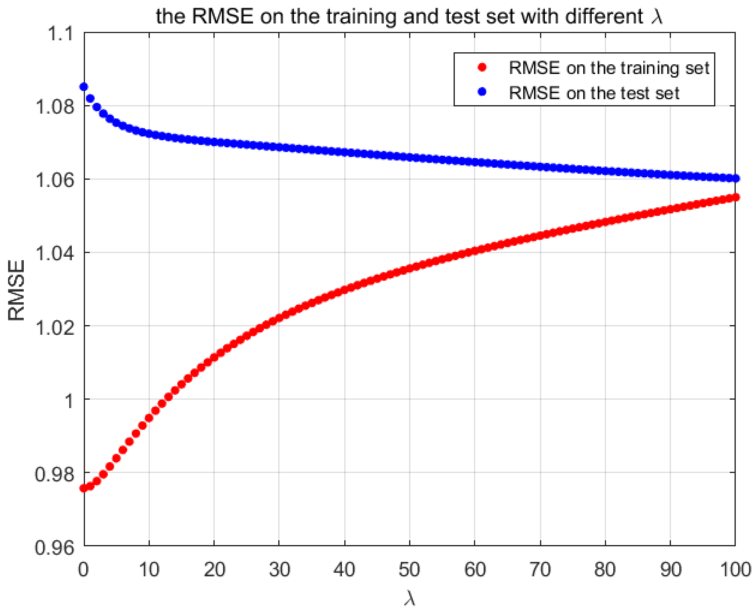


Figure 1: The RMSE on the training and test set (L2)

Figure 2 shows all the weights as a function of λ.

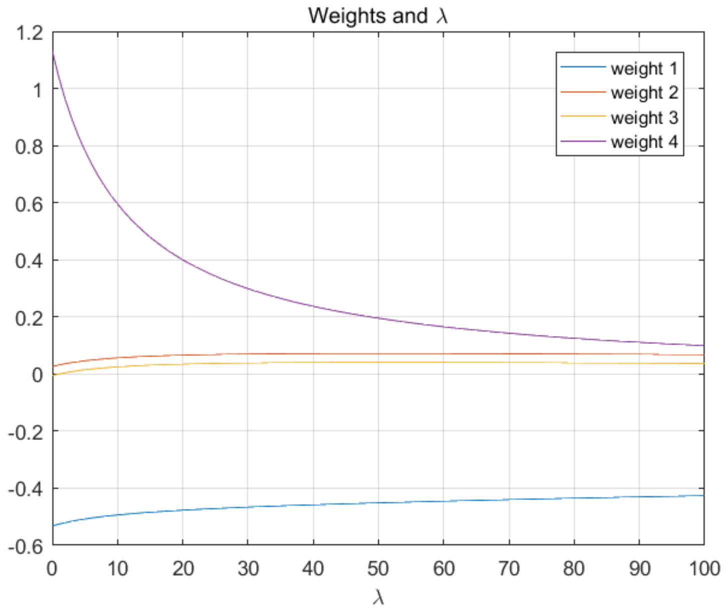


Figure 2: All the weights as a function of λ (L2)

1.2 b

There are 4 weights in this problem:

$$\min_{w_1,w_2,w_3,w_4} \sum_{j=1}^{m} (y_j - w_1x_1 - w_2x_2 - w_3x_3 - w_4x_4)^2$$

such that

w$_1 + w_2 + w_3 + w_4 \leq \eta, w_1 + w_2 + w_3 - w_4 \leq \eta, w_1 + w_2 - w_3 + w_4 \leq \eta,$

w$_1 + w_2 - w_3 - w_4 \leq \eta, w_1 - w_2 + w_3 + w_4 \leq \eta, w_1 - w_2 + w_3 - w_4 \leq \eta,$

w$_1 - w_2 - w_3 + w_4 \leq \eta, w_1 - w_2 - w_3 - w_4 \leq \eta, -w_1 + w_2 + w_3 + w_4 \leq \eta,$

-w$_1 + w_2 + w_3 - w_4 \leq \eta, -w_1 + w_2 - w_3 + w_4 \leq \eta, -w_1 + w_2 - w_3 - w_4 \leq \eta,$

-w$_1 - w_2 + w_3 + w_4 \leq \eta, -w_1 - w_2 + w_3 - w_4 \leq \eta,$

-w$_1 - w_2 - w_3 + w_4 \leq \eta, -w_1 - w_2 - w_3 - w_4 \leq \eta$

In order to adopt quadratic programming:

$$J(w) = \frac{1}{2} w^T H w + f^T w$$

subject to

$$Aw \leq b$$

where

$$\begin{aligned} H &= X^T X, \\ f &= X^T y \\ A &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \end{aligned}$$

The function implemented in L1Regularization.m.

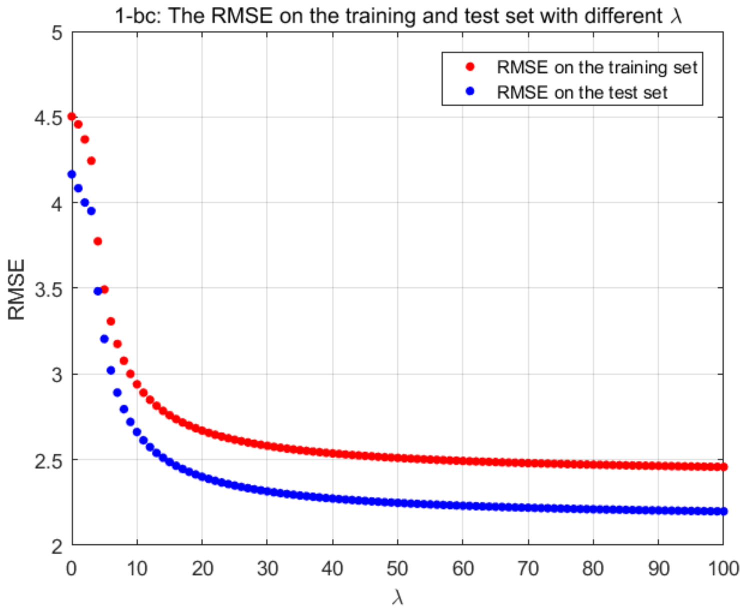


Figure 3: The RMSE on the training and test set (L1)

1.3 c

Figure 3 shows the RMSE (root mean squared error) on the training set and the test set, as a function of the regularization parameter λ.
Figure 4 shows all the weights as a function of λ.

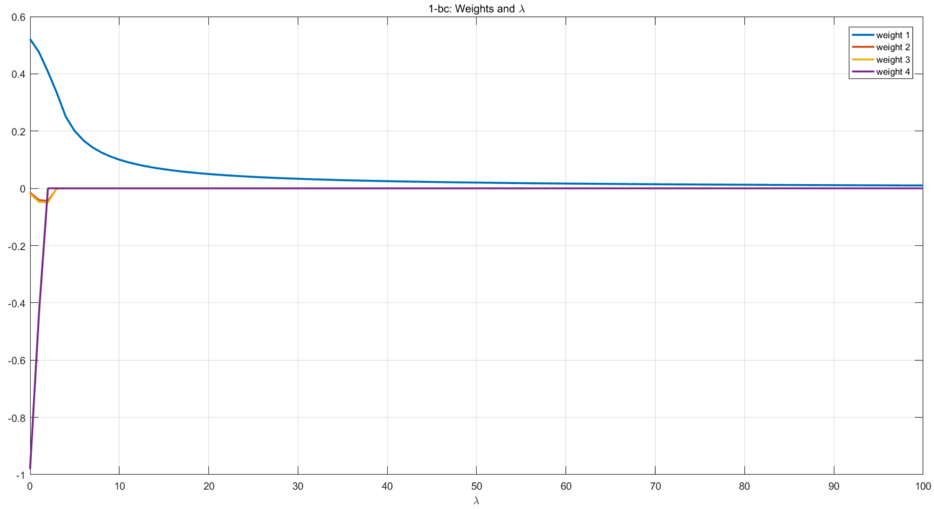


Figure 4: All the weights as a function of $\lambda(L1)$

I observed that as the λ growing, all weights are gradually close to 0. A regression model that uses L1 regularization technique is called Lasso Regression and model which users L2 regression is called Ridge Regression. The key difference between these techniques is that Lasso shrinks the less important feature's weight to 0, thus remove some features. Because in this data set, L1 performs better than L2, since the input data has 3 features (without bias term), I think the generation method of this data set is copying some features(maybe 1 or 2) from one feature, add some noise, so these features may more or less have some relevance.

2 Dealing with missing data

The output Y is modeled as

$$P(y) = \theta^y (1-\theta)^{1-y}$$

using the Baye's rule, we have:

$$P(y=1|X) = \frac{P(X|y=1)P(y=1)}{p(X)}$$

and the two classes are modeled as:

$$\begin{aligned} P(X|y=0) &= \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(X-\mu_0)^T\Sigma^{-1}(X-\mu_0)) \\ P(X|y=1) &= \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(X-\mu_1)^T\Sigma^{-1}(X-\mu_1)) \end{aligned}$$

Since the log-odds:

$$a = \ln \frac{P(X|y=1)P(y=1)}{P(X|y=0)P(y=0)}$$

The function can be written as:

$$\begin{aligned} P(y=1|X) &= \frac{1}{1+\exp(-a)} \\ &= \frac{1}{1+\exp(-(\mu_1-\mu_0)^T\Sigma^{-1}(X-\frac{\mu_0}{\mu_1^2})+\log(\frac{1-\theta}{\theta}))} \end{aligned}$$

Since the class-conditional means:

$$\begin{aligned} E(x_n|y=0) &= \mu_0 \\ E(x_n|y=1) &= \mu_1 \end{aligned}$$

In this situation, the value of x_n is missing and "fill in" the value by its class-conditional means. By the expression of $P(y=1|X)$, as the filled data will not change μ_0, μ_1, Σ and θ, thus the prediction of Y will not change.

3 Naive Bayes assumption

3.1 a

For the initial Naive Bayes classifier, it has 5 parameters, more specifically, these parameters are:

$$\begin{aligned} \theta_{1,0} &= P(x_1=1|y=0); \\ \theta_{2,0} &= P(x_2=1|y=0); \\ \theta_{1,1} &= P(x_1=1|y=1); \\ \theta_{2,1} &= P(x_2=1|y=1); \\ \theta_1 &= P(y=1); \end{aligned}$$

For the classifier with three features, it has 7 parameters, more specifically, these parameters are:

$$\begin{aligned} \theta_{1,0} &= P(x_1=1|y=0); \\ \theta_{2,0} &= P(x_2=1|y=0); \\ \theta_{3,0} &= P(x_3=1|y=0); \\ \theta_{1,1} &= P(x_1=1|y=1); \\ \theta_{2,1} &= P(x_2=1|y=1); \\ \theta_{3,1} &= P(x_3=1|y=1); \\ \theta_1 &= P(y=1); \end{aligned}$$

3.2 b

In this situation, the feature x_3 are added duplicate from x_2, when the parameters are learned perfectly, the parameter $\theta_{2,0}$ should be equal to $\theta_{3,0}$, the parameter $\theta_{2,1}$ should be eqaul to $\theta_{3,1}$. The decision boundary:

$$\frac{P(y=1|x)}{P(y=0|x)} = \frac{P(y=1)\prod_{i=1}^{3}\theta_{i,1}^{x_i}(1-\theta_{i,1})^{1-x_i}}{P(y=0)\prod_{i=1}^{3}\theta_{i,0}^{x_i}(1-\theta_{i,0})^{1-x_i}}$$

Using the log trick:

$$\log\frac{P(y=1|x)}{P(y=0|x)} = \log\frac{P(y=1)}{P(y=0)} + \log(\frac{P(x_1|y=1)}{P(x_1|y=0)}) + \log(\frac{P(x_2|y=1)}{P(x_2|y=0)}) + \log(\frac{P(x_3|y=1)}{P(x_3|y=0)})$$

So the worst scenario is when the distribution of feature is highly unbalanced to let the value of $P(x_3|y=1)$ too large or too small. However, since most of the x are i.i.d, the influence to the decision boundary would be very small, the classifier will works well.

4 Conjugate priors for Gaussian distribution

4.1 a

Since the data are N i.i.d, the likelihood of the entire data set is equal to the product of the likelihoods of each data samples:

$$\begin{aligned}
L(X|\Gamma,\mu) &= \prod_{i=1}^{N} L(x_i|\Gamma,\mu)\\
&= \prod_{i=1}^{N} \Gamma^{\frac{1}{2}}\exp((\frac{-\Gamma}{2}(x_i-\mu)^2))\\
&= \Gamma^{\frac{N}{2}}\exp(\frac{-\Gamma}{2}\sum_{i=1}^{N}(x_i-\mu)^2)\\
&= \Gamma^{\frac{N}{2}}\exp(\frac{-\Gamma}{2}\sum_{i=1}^{N}(x_i-\bar{x}+\bar{x}-\mu)^2)\\
&= \Gamma^{\frac{N}{2}}\exp(\frac{-\Gamma}{2}\sum_{i=1}^{N}((x_i-\bar{x})^2+(\bar{x}-\mu)^2))\\
&= \Gamma^{\frac{N}{2}}\exp(\frac{-\Gamma}{2}(Ns+N(\bar{x}-\mu)^2))
\end{aligned}$$

where $\bar{x}=\frac{1}{N}\sum_{i=1}^{N}x_i$ is the mean value of the data samples, and $s=\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2$ is the variance of the data samples.

The posterior distribution of the parameters is proportional to the prior times the likelihood.

$$\begin{aligned}
P(\Gamma,\mu|X) &= L(X|\Gamma,\mu)\text{Gam}(\Gamma,\mu)\\
&= \Gamma^{\frac{N}{2}}\exp(\frac{-\Gamma}{2}(Ns+N(\bar{x}-\mu)^2))\Gamma^{a-\frac{1}{2}}\exp(-b\Gamma)\exp(-\frac{\lambda\Gamma(\mu-\mu_0)^2}{2})\\
&= \Gamma^{\frac{N}{2}+a-\frac{1}{2}}\exp(-\Gamma(\frac{1}{2}Ns+b))\exp(-\frac{\Gamma}{2}(\lambda(\mu-\mu_0)^2+N(\bar{x}-\mu)^2))\\
&= \Gamma^{\frac{N}{2}+a-\frac{1}{2}}\exp(-\Gamma(\frac{1}{2}Ns+b))\exp(-\frac{\Gamma}{2}((\lambda+N)(\mu-\frac{\lambda\mu_0+N\bar{x}}{\lambda+N})^2+\frac{\lambda N(\bar{x}-\mu_0)^2}{\lambda+N})\\
&= \Gamma^{\frac{N}{2}+a-\frac{1}{2}}\exp(-\Gamma(\frac{1}{2}Ns+b+\frac{\lambda N(\bar{x}-\mu_0)^2}{2(\lambda+N)}))\exp(-\frac{\Gamma}{2}(\lambda+N)(\mu-\frac{\lambda\mu_0+N\bar{x}}{\lambda+N})^2)
\end{aligned}$$

The final expression is in exactly the same form as a Gaussian-gamma distribution, i.e.,

$$P(\Gamma, \mu|X) = \text{NormalGamma}(\frac{\lambda\mu_0 + N\bar{x}}{\lambda + N}, \lambda + N, a + \frac{N}{2}, b + \frac{1}{2}(Ns + \frac{\lambda N(\bar{x} - \mu_0)^2}{\lambda + N}))$$

4.2 b

Considering the likelihood function of Λ for a given data set $x_1, ..., x_N$:

$$\begin{aligned}\prod_{n=1}^{N} L(x_n|\mu, \Lambda^{-1}) &\propto |\Lambda|^{\frac{N}{2}} \exp(-\frac{1}{2}\sum_{n=1}^{N}(x_n - \mu)^T\Lambda(x_n - \mu)) \\ &= |\Lambda|^{\frac{N}{2}} \exp(-\frac{1}{2}\text{Tr}(\Lambda\sum_{n=1}^{N}(x_n - \mu)(x_n - \mu)^T))\end{aligned}$$

Compare with (2.155), the functional dependence on Λ is indeed the same and thus a product of this likelihood and a Wishart prior will result in a Wishart posterior.

5 Using discriminative vs. generative classifiers

5.1 a

In this problem, I use a leaning-rate version. And the best learning rate that I choose is 1.4.
Figure 5 shows the different learning rates and with its error on training set.

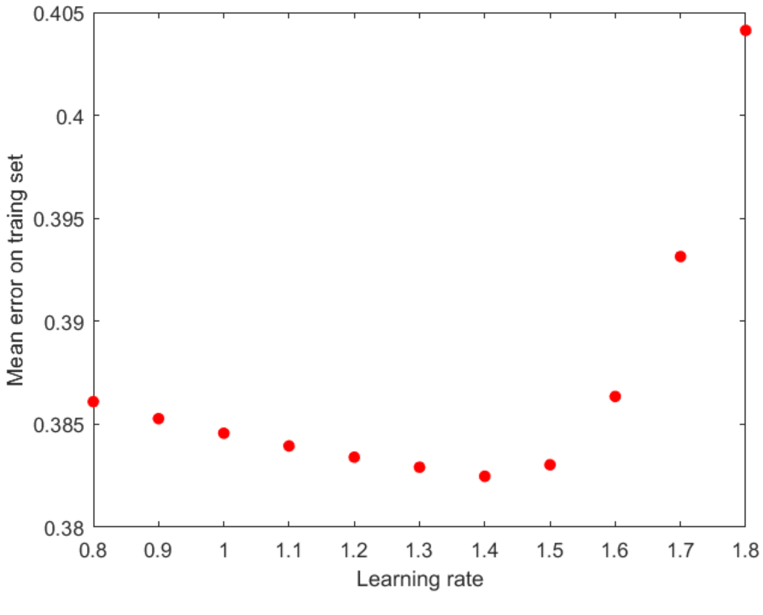


Figure 5: Different learning rate with its error on training set

Figure 6 shows the training error on the full data set over iterations using the best learning rate that I choose.

5.2 b

The training process of Gaussian Naive Bayes classifier is implemented in GaussianNaiveBayesTrain.m and the prediction process of Gaussian Naive Bayes classifier is implemented in GaussianNaiveBayesPredict.m. I use 10-fold cross-validation, obtain the prediction accuracy on training set and test sets. Figure 7 shows the result. I observed that in this situation, the Logistic regression and Gaussian Naive Bayes achieve nearly the same accuracy and performance.

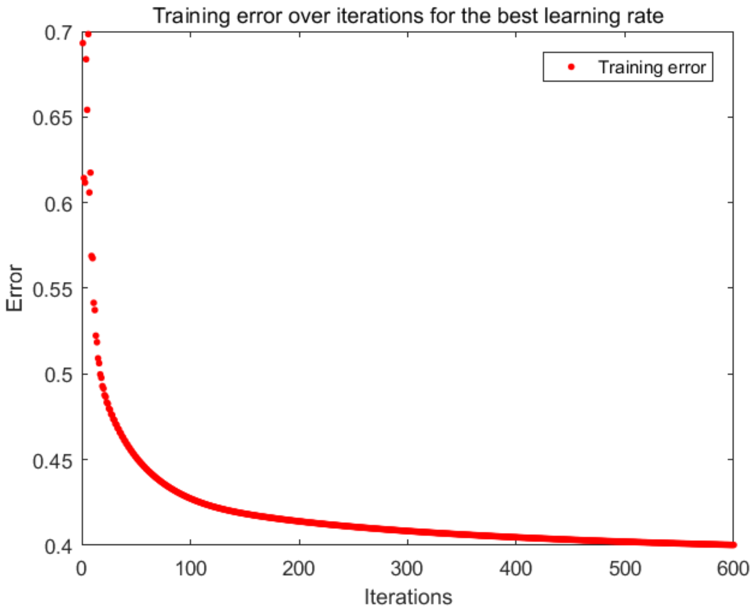
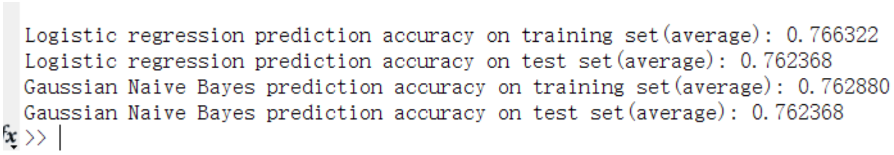


Figure 6: Training error on the full data set using the best learning rate



```
Logistic regression prediction accuracy on training set(average): 0.766322
Logistic regression prediction accuracy on test set(average): 0.762368
Gaussian Naive Bayes prediction accuracy on training set(average): 0.762880
Gaussian Naive Bayes prediction accuracy on test set(average): 0.762368
>> |
```

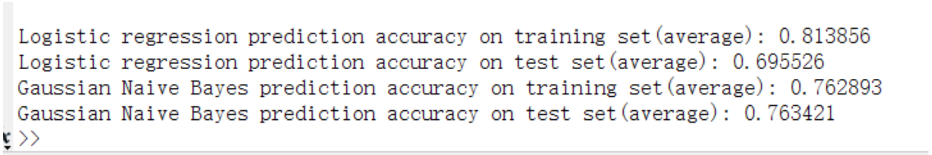
Figure 7: prediction accuracy of logistic regression and Gaussian Naive Bayes

5.3 c

I implement Gaussian naive Bayes which using all the inputs (features) in Solution_5cd.m.

5.4 d

Figure 8 shows the prediction accuracy results of logistic regression and Gaussian Naive Bayes. Logistic regression is a type of discriminative classifier, it directly estimates the parameters of $P(Y|X)$, and Naive Bayes is a type of generative classifier which directly estimates parameters of $P(Y)$ and $P(X|Y)$. Actually, if the assumptions of Gaussian Naive Bayes hold, with more and more training examples, the performance of the GNB and Logistic Regression are become more and more identical.



```
Logistic regression prediction accuracy on training set(average): 0.813856
Logistic regression prediction accuracy on test set(average): 0.695526
Gaussian Naive Bayes prediction accuracy on training set(average): 0.762893
Gaussian Naive Bayes prediction accuracy on test set(average): 0.763421
>>
```

Figure 8: prediction accuracy of logistic regression and Gaussian Naive Bayes

6 Bayesian learning example for Bishop