# CS6140 Assignment3 Guanglei “Garrett” Wu

# 1 Logistic Regression

1.3

1,

## Spambase:

Evaluation for training data:

Mean Accuracy: 0.918594542381

Std Accuracy: 0.00210648592461

Mean Precision: 0.924624342506

Std Precision: 0.00280342992158

Mean Recall: 0.863828200311

Std Recall: 0.00375707763759

Evaluation for testing data:

Mean Accuracy: 0.915

Std Accuracy: 0.0126180624878

Mean Precision: 0.917958759762

Std Precision: 0.0186741495177

Mean Recall: 0.861744993047

Std Recall: 0.0223033673753

## Breast Cancer:

Evaluation for training data:

Mean Accuracy: 0.984015594542

Std Accuracy: 0.0014587358233

Mean Precision: 0.990371802316

Std Precision: 0.0021168735218

Mean Recall: 0.966575994938

Std Recall: 0.00331070567625

Evaluation for testing data:

Mean Accuracy: 0.975

Std Accuracy: 0.0163663417677

Mean Precision: 0.985929951691

Std Precision: 0.021755981794

Mean Recall: 0.941119688854

Std Recall: 0.0461416824071

## Diabets:

Evaluation for training data:

Mean Accuracy: 0.775289017341

Std Accuracy: 0.0060365891833

Mean Precision: 0.725090478463

Std Precision: 0.00984529312748

Mean Recall: 0.573956946479

Std Recall: 0.015405730368

Evaluation for testing data:

Mean Accuracy: 0.775

Std Accuracy: 0.0284952734575

Mean Precision: 0.726433125698

Std Precision: 0.0742447575789

Mean Recall: 0.566873209073

Std Recall: 0.114533225728

2,

## Breast Cancer:

Learning rate: 0.0001

Tolerance: 0.03



3,

I choose to select max iterations first, make it large enough, to let the iteration stop before reaching the max value, but not too large so if it does not converge or converges very slow, it can stop after all. I use 1000 for all the cases and it runs well. Then, after choosing appropriate learning rate, try different values of tolerance, observe the testing outcome. If under fitting, should try smaller tolerance. If over fitting, try larger tolerance. If the outcomes are not much difference, I tend to use larger tolerance, to make the program run faster. Another issue for the logistic function is because it has a 1+e^-a field, it causes Float Precision issue on my machine, I need to choose carefully the tolerance to make sure that field does not approximate to 1.0. Or we can fix this issue using some big number type. However, it compromises efficiency.

## Breast Cancer:

Learning rate: 0.0001

Tolerances = [0.3e-1, 0.1, 0.3, 1, 3, 10]

We can see the training cost increase as tolerance increases. If tolerance < 0.3e-1, will cause Float Precision problem.



1.4

1,

2,

For P(y = 1), = P(y = 1)

We need to show: P(y = -1) = 1 – P(y = 1):

So proved.

3,

Likelihood function is

Log-likelihood is

Logistic loss function is -LL

When y and are the same sign (correct prediction), is negative, is small, and contribute to smaller value of loss function.

When y and are the opposite sign (incorrect prediction), is position, is large, and contribute to larger value of loss function.

# 2 Naïve Bayes

2.5

1

 

It can be seen, generally Multinomial model outperforms Bernoulli model, especially when vocabulary size is large. It is intuitive since multinomial contains more information. However, when vocabulary size is large, continue increase vocabulary size does not help to the outcome. It may because the words added are less characteristic and less relevant.

2

Choose size 25000, seems the best performance.

Precisions:





For Precisions, generally, Multinomial outperforms Bernoulli, which is accordance with Accuracy. It seems Bernoulli produce bad precision for class 7, means it classify a large number of not 7 as 7. May need additional analysis for that class.

Recalls:





For recalls, generally, Multinomial outperforms Bernoulli, which is also accordance with Accuracy. Both models perform bad on class 20, especially Bernoulli model, means they can hardly distinguish class 20 from other classes. May need additional analysis for that class 20, to see if any more remarkable character for the class.

2.6

## 1 Bernoulli with Beta:

Likelihood:

Log Likelihood:

is the same as the version without conjugate prior,

(This formula cannot be maximized by simply increase , so Lagrange Multiplier is not needed here. We can also get Lambda = 0 if included)

## 2 Multinomial with Dirichlet:

Likelihood:

Log likelihood:

Add Lagrange Multiplier:

is the same as the version without conjugate prior,

## 3

MLE estimates the likelihood and parameters based on the data, while MAP gives a weight onto a predetermined distribution, i.e. provided some information for the parameters.

For Bernoulli case, MAP assumes before seeing the data, while MLE has 0 information before seeing the data.

For Multinomial case, MAP assumes before seeing the data, while MLE has 0 information before seeing the data.

2.7

## 1 & 2

Given for Bernoulli and for Multinomial, we have exactly the same estimates for MAP and MLE with Laplace smoothing. The plots are the same as problem 2.5.

## 3

MAP in somehow, prevents overfitting, which is a problem of MLE. The differences of accuracy may depends on the data and the parameter chosen for MAP.

MLE estimates the likelihood and parameters based on the data, while MAP gives a weight onto a predetermined distribution, i.e. provided some information for the parameters. For MLE, because the parameters are also estimated from the data, it has overfitting issues. It can be relieved by introducing smoothing. While MAP already introduced predetermined estimated distribution for the model, the performance may relate to the choices of the parameters of the conjugate prior distribution. If the prior distribution plays a large weight, may cause under fitting, if it plays a light weight, may cause over fitting. It can be seen as another way of regularization or smoothing. In our case, choosing those specific parameters makes our MAP estimate the same as MLE with smoothing. Showing that MAP is a generalized method of MLE with smoothing.