CS545 Machine learning Assignment 1

Nupur Kulkarni

September 2016

Contents

1 Introduction

"Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence."[1]Machine learning teaches computer to do stuffs in future,make predictions ,behave intelligently based on some observations,past experiences and behaviours.So in general Machine Learning let the computers learn to do the task by themselves rather than teaching them.

Machine learning algorithms generally recognized by following three types:

- Supervised Learning :
  In supervised learning,algorithm is provided with labeled set of training data.Model is trained on the label data to predict label of the new test data.
- Unsupervised Learning:
  Unsupervised learning algorithms are always provided with large set of unlabeled data.Task is to find useful pattern by analysing the data.
- Reinforcement Learning :
  These algorithms are always applied to maximize the performance of the system.It is based on the feedback from the external environment.It deals with possible examples to reinforce for the betterment.

The majority of the practical problems use supervised learning approach.These problems are mostly grouped into regression and classification.Classification problem is where output is defined by classes.The aim of classification algorithm is to make attempt to learn classifier that can distinguish the two. For example "positive" or "negative".

To classify and train data ,there are many classification algorithms available.Choosing one of them depend on the problem we want to solve,how large the data set is and the accuracy ,error rate we expect to achieve.Moreover,in machine learning,different features are used as the basis to train the model and solve the problem.However,these features are too many to handle.In this case,it is very much necessary that we select only those features that are crucial and relevant to us and then use it on our data. Here,I am dealing with Binary classifier which classifies the data set into the two classes one is positive and other is negative.

2 Part 1:Measuring Classifier Error

In supervised learning we work with labeled dataset.Lets say we have N labeled examples where X is input data and Y is the label associated with X.As we are dealing with binary classification,labels have values $\pm 1 to denote positive/negative examples.$

Estimate of classifier error is given by :

$$E(h) = \frac{1}{N} \sum_{i=1}^{N} I(h(X_i) \neq Y_i) \tag{1}$$

Where h is predictor function and I(.) is the indicator function.

2.1 Estimated Error of Majority Classifier :

Given test data is highly imbalanced.Imbalanced dataset denotes the unequal distribution between its classes and it can be significant or sometimes extreme.As mentioned here we are dealing with the binary classification problem which classify the examples in two classes: one is positive and other is negative.The problem we are dealing with has more number of negative examples than the number of positive examples to classify.

The classifier is trained on the dataset which always classifies an example as belonging to the majority class, i.e. the class to which the largest number of training examples belongs to.In such case,when trained classifier is provided with test data,classification algorithms are always biased towards majority class and try to classify every example as the majority class.In attempt to optimize the error rate ,classifier doesn't consider data distribution and sometimes minority data is treated as outlier even if it is important.
Hence,in our example,there will be two scenarios based on trained classifier:

- If the classifier is trained on the data which has more negative examples than the positive one,the majority classifier will be more trained to classify negative example.As our test data has more negative examples,$h(x_i) \neq y_i$ will not be true for majority of examples.Thus E(h) i.e. estimation error of majority classifier will be less and accuracy will be more.
- If the classifier is trained on the data which has more positive examples than the negative one,the majority classifier will be more biased towards positive examples.But our test data has more negative examples.As classifier will try to classify every example as positive one,there is probability of misclassification. More positive example might get classified as negative.Hence,$h(x_i) \neq y_i$ will qualify for many examples and error rate of majority classifier is more.

2.2 Issues in evaluating classifiers :

Classifiers are evaluated in terms of accuracy or error rate.If the class distribution is balanced,both might be efficient measures of evaluation.But if distribution of the class is imbalanced,both accuracy and rate are biased towards majority classifier.
Consider the breast cancer dataset with "positive" and "negative" labels representing "cancerous" or "healthy" patient respectively.In real life, number of healthy patients is always greater than number of the patients having cancer.Suppose there are 10,000 patients detected as healthy and 300 patient are diagnosed with

cancer. In this case its important to have balance error or accuracy rate on both majority and minority classifier. But in reality,let's say we get imbalanced accuracy on both classifiers,say 100% for majority classifier and 10% for minority classifier.So 10%accuracy on minority classifier suggest that almost 270 samples are misclassified to majority classifier.This in turn means approximately 270 cancerous patients are classified as healthy patients. That is really worst for the medical industry and have severe consequences.
Thus,conventional methods such as accuracy or error rate are not sufficient to evaluate classifiers in case of imbalanced learning.Therefore,we need to have methods like cost matrix,precision recall or ROC Curve.[3]

2.3 Cost Matrix

" Most of the classifiers assume that the misclassification costs(false negative and false positive cost) are the same"[4].In reality,this is not true.The cost of misclassifing cancer patient as healthy could cost patient life.This happens as majority classifier is trained to classify healthy people because majority of population is healthy.So false negative cost is always higher and worst than false positive cost.
we are assuming majority class as "Negative(-1)" and minority class as "Positive(+1)".Given,to solve this issue we need to assign different cost to different types of error.The following table gives confusion matrix for binary classification with labels +1 and -1.

	Positive	Negative
Positive	True Positive $c(+1,+1)$	False Negative $c(+1,-1)$
Negative	False Positive $c(-1,+1)$	True Negative $c(-1,-1)$

Table 1: Confusion Matrix

cost matrix $c(y_i, h(x_i))$ is defined where y_i is the actual class of example i,$h(x_i)$ is the predicted class.

$$c_r = c(+1,-1)(FalseNegative)$$
$$c_a = c(-1,+1)(FalsePositive)$$

In case of imbalanced data, minority class examples gets classified as majority class examples. Thus,False negative cost is larger than false positive cost.

$$c_r > c_a$$

Suppose total examples in training data set is N.If x is the number of negative sample available, then positive samples are N-x.
Here we can say from the given error rate formula that error rate depend on the no of misclassified samples.In case of majority classifier,False Negative cost will be more as positive samples get classified as negative one.But due to less no of positive sample,error rate of majority classifier is low.To penalize for misclassification c_r reject cost is considered.

Similarly,In case of minority class which is positive,False positive rate is low with respect to false negative rate.Again its depend on test data.

$$E(h) = \frac{1}{N}\sum_{i=1}^{N} C(y_i, h(X_i))I(h(X_i) \neq Y_i)$$

With above formula we can say that $c_r = \frac{N-x}{N}$ and $c_a = \frac{x}{}$.So the ration of rejaect and accept cost is ,

$$\frac{c_r}{c_a} = \frac{N-x}{x}$$

So to reduce error rate in case of majority classifier and minority, this ratio should be 1.c_r value can be decrease if false negative samples are less and $C_a$value can be increased if false positive sample are in balnce ration with false negative. Balced data can achieve error rate 0.5 for both majority and minority classifier.

3 Part 2: The Nearest Centroid Classifier

In machine learning,the nearest centroid classifier assign class label to the data point(vector) and classify that point to the appropriate class.While choosing the class it looks for distance of the new point from the centroid of the classes present in the hyperplane.One where the distance is minimum,data point is labeled for that class.
As given we are dealing with binary classification problem, where there are two classes represented by "Positive(+1)" and "Negative(-1)" label.The following figure illustrate the classification of positive and negative labeled data points.
Consider Centroid for both positive and negative classes are represented by μ_2 and μ_1 respectively.If we

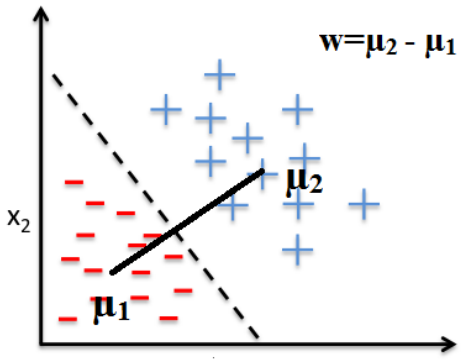


Figure 1: Nearest Centroid Classifier

connect these two centroids,connecting vector is orthogonal to the the plane.Thus,we can represent this linear classification with weight vector "w" as follow.

$$\boldsymbol{w = \mu_2 - \mu_1} \tag{2}$$

Replacing

μ_1 and μ_2 in equation 2 with corresponding equation for centroid ,

$$\boldsymbol{w = \frac{1}{|C_p|} \sum_{i \in C_p} X_i - \frac{1}{|C_n|} \sum_{i \in C_n} X_i} \tag{3}$$

where C_p and C_n are defined as cardinality of positive and negative class respectively.
However,it is only valid if we consider both classes separately. But if we apply this on whole data ,there are chances that in the process we could loose some data.It is difficult to identify data vectors without label associated with them.Hence,it is important to consider respective labels.

$$\boldsymbol{w = \frac{1}{|C_p|} \sum_{i \in C_p} X_i(+1) + \frac{1}{|C_n|} \sum_{i \in C_n} X_i(-1)} \tag{4}$$

Assuming number of positive and negative examples are equal,$C_p = C_n = \frac{N}{2}$.N is the total no of examples in training data set. In general ,labels can be represented with y_i.We could unify the summation in equation 4 to represent weight vector for entire database.

Substituting equation $C_p + C_n = N$ and label as Y_i,

$$\boldsymbol{w = \frac{1}{N} \sum_{i \in N} X_i y_i} \tag{5}$$

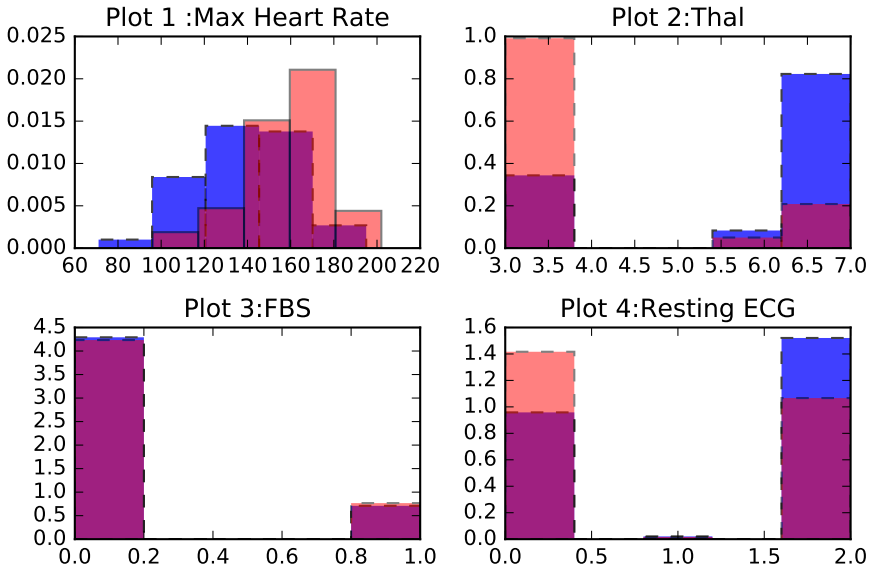
from equation 2 and 5,weight vector for linear classification can be represented as,

$$\boldsymbol{\mu_2 - \mu_1 = \frac{1}{N} \sum_{i \in N} X_i y_i} \tag{6}$$

where number of positive and negative samples are equal.

4 Part 3: Histograms depicting Usefulness of Features

I have chosen Heart dataset which has 14 attributes.These attributes are age,sex,chest pain,resting blood pressure,cholesterol,fasting blood sugar,resting ECG,maximum heart rate,exercise induced angina, ST depression induced by exercise relative to rest,the slope of the peak exercise ST segment ,number of major vessels and thal.Attributes have labels associated with them classifying dataset in to two classes "presence of heart disease(+1)" and "Healthy(-1)".The following figure represent two features that are useful for classifying dataset efficiently and two which has limited use in my judgement.



I have selected Four features.From the above graph "Plot 1:Max heart rate(thalach)" and Plot 2:Thal i think are the important one for classification.While "Plot 3 : FBS" and Plot 4:Resting Blood Sugar" are of limited use. Blue color shows "Positive" Samples and red shows "Negative" one.

4.1 Reasoning of behind Each Plot

- **Plot1:Max Heart Rate (Thalach) :**
  Thalach is a maximum heart rate achieve.It can be interpreted from the "Plot 1" that in certain range "140 to 170" there is maximum overlapping of both negative and positive data and uniform ditribution of both classes.But we can say with some certainty that below 140 ,people have heart disease(blue:positive) and above 170 they do not have heart disease (red :negative)..Hence we can say that in this feature we can linearly seperate Positive and negative class.

- **Plot2:Thal :**
  Thal feature is defined with 3 classe "3 = normal; 6 = fixed defect; 7 = reversable

defect".It can be interpreted from the graph that if thal value is in between "3 to 4" ,"negative(No heart disease)" examples are more.If thal is in range "5.5 to 7" ,there is maximum positive examples.Hence we can predict that if thal value is in range 3 to 3.5 there is maximum possibility that people will not have heart disease and if its between "5.5 to 7",there are maximum chance of presence of heart disease.Again this feature enables linear classification of data.

- **Plot3:FBS:**
  FBS is fasting blood sugar.Graph shows maximum overlapping of both positive and negative data.It is very difficult to classify data into classes just with observing graph.So in my opinion it will be least useful feature.

- **Plot 4:Resting ECG:**
  Plot 4 show overlapping between positive and negative classes.As data can be linearly classified,Overlapping is significant.So i think it can counted as a limited use feature.

4.2 Code Snippet

```
import numpy as np

import pylab as pl


header_row = ['label','Max Heart rate','Thal','Cholestoral','FBS']
data = np.genfromtxt('heart.data', delimiter=',',usecols=(1,9,14,7,8),names=header_row,com


from matplotlib import pyplot as plt


#dictionary for positive data
dp=dict()
#dictionary for negative data
dn=dict()
for m in range(1,5):
    positive =[]
    negative=[]
    for i in range(len(data)):
        if data[i][0]==1.0:
            positive.append(data[i][m])
        else:
            negative.append(data[i][m])

    dp[m]=positive
    dn[m]=negative


plt.figure(figsize=(100,60))
f, axarr = plt.subplots(2,2)
plt.subplots_adjust(hspace=0.75)
axarr[0, 0].hist(dp[1],5, normed=True, facecolor='b', ls='dashed',alpha=0.75,label='Positiv
axarr[0, 0].hist(dn[1],5, normed=True, facecolor='r',alpha=0.5,label='Negative', cumulative
axarr[0, 0].set_title('Plot 1 :Max Heart Rate')
```

```
axarr[0, 1].hist(dp[2],5, normed=True, facecolor='b', ls='dashed',alpha=0.75,label='Positiv
axarr[0, 1].hist(dn[2],5, normed=True, facecolor='r', ls='dashed',alpha=0.5,label='Negative
axarr[0, 1].set_title('Plot 2:Thal')

axarr[1, 0].hist(dp[3],5, normed=True, facecolor='b', ls='dashed',alpha=0.75,label='Positiv
axarr[1, 0].hist(dn[3],5, normed=True, facecolor='r', ls='dashed',alpha=0.5,label='Negative
axarr[1, 0].set_title('Plot 3:FBS')

axarr[1, 1].hist(dp[4],5, normed=True, facecolor='b', ls='dashed',alpha=0.75,label='Positiv
axarr[1, 1].hist(dn[4],5, normed=True, facecolor='r', ls='dashed',alpha=0.5,label='Negative
axarr[1, 1].set_title('Plot 4:Resting ECG')

#plt.figlegend(,("Positive","Negative"), loc = 'upper right', ncol=5, labelspacing=0. )
plt.tight_layout()
plt.savefig('out.pdf',dpi=200)
plt.show()
```

References

[1] Machine Learning,https://en.wikipedia.org/wiki/Machine_learning.

[2] Learning From Imbalance Classes,http:// www.svds.com/learning-imbalanced classes.

[3] He, Haibo, and Edwardo A. Garcia. "Learning from imbalanced data." IEEE Transactions on knowledge and data engineering 21.9 (2009): 1263-1284.

5 Appendix

5.1 Feature selection with Histogram

```
# -*- coding: utf-8 -*-
"""
Name :Nupur Kulkarni
Machine learning Assignment 1, part 3
Data Set : heart.data


Useful Features :
Thal(13)
Thalach(8)-maximum heart rate

Limited Use features :
Fasting blood sugur(FBS)(6)
RestEcg(7)


"""

import numpy as np

import pylab as pl


header_row = ['label','Max_Heart_rate','Thal','Cholestoral','FBS']
data = np.genfromtxt('heart.data', delimiter=',',usecols=(1,9,14,7,8),names=header_row,com


from matplotlib import pyplot as plt


#dictionary for positive data
dp=dict()
#dictionary for negative data
dn=dict()
for m in range(1,5):
    positive =[]
    negative=[]
    for i in range(len(data)):
        if data[i][0]==1.0:
            positive.append(data[i][m])
```

```
        else:
            negative.append(data[i][m])

    dp[m]=positive
    dn[m]=negative


plt.figure(figsize=(100,60))
f, axarr = plt.subplots(2,2)
plt.subplots_adjust(hspace=0.75)
axarr[0, 0].hist(dp[1],5, normed=True, facecolor='b', ls='dashed',alpha=0.75,label='Positiv
axarr[0, 0].hist(dn[1],5, normed=True, facecolor='r',alpha=0.5,label='Negative', cumulative
axarr[0, 0].set_title('Plot 1 :Max Heart Rate')

axarr[0, 1].hist(dp[2],5, normed=True, facecolor='b', ls='dashed',alpha=0.75,label='Positiv
axarr[0, 1].hist(dn[2],5, normed=True, facecolor='r', ls='dashed',alpha=0.5,label='Negative
axarr[0, 1].set_title('Plot 2:Thal')

axarr[1, 0].hist(dp[3],5, normed=True, facecolor='b', ls='dashed',alpha=0.75,label='Positiv
axarr[1, 0].hist(dn[3],5, normed=True, facecolor='r', ls='dashed',alpha=0.5,label='Negative
axarr[1, 0].set_title('Plot 3:FBS')

axarr[1, 1].hist(dp[4],5, normed=True, facecolor='b', ls='dashed',alpha=0.75,label='Positiv
axarr[1, 1].hist(dn[4],5, normed=True, facecolor='r', ls='dashed',alpha=0.5,label='Negative
axarr[1, 1].set_title('Plot 4:Resting ECG')

#plt.figlegend(,("Positive","Negative"), loc = 'upper right', ncol=5, labelspacing=0. )
plt.tight_layout()
plt.savefig('out.pdf',dpi=200)
plt.show()
```