CS289–Spring 2017 — Homework 1 Solutions

Shuhui Huang, SID 3032129712

Collaborators: Ruonan Hao

1. Problem 1

Answer: To select the subset of each data set, I choose the index of each data set randomly and get the corresponding validation set.

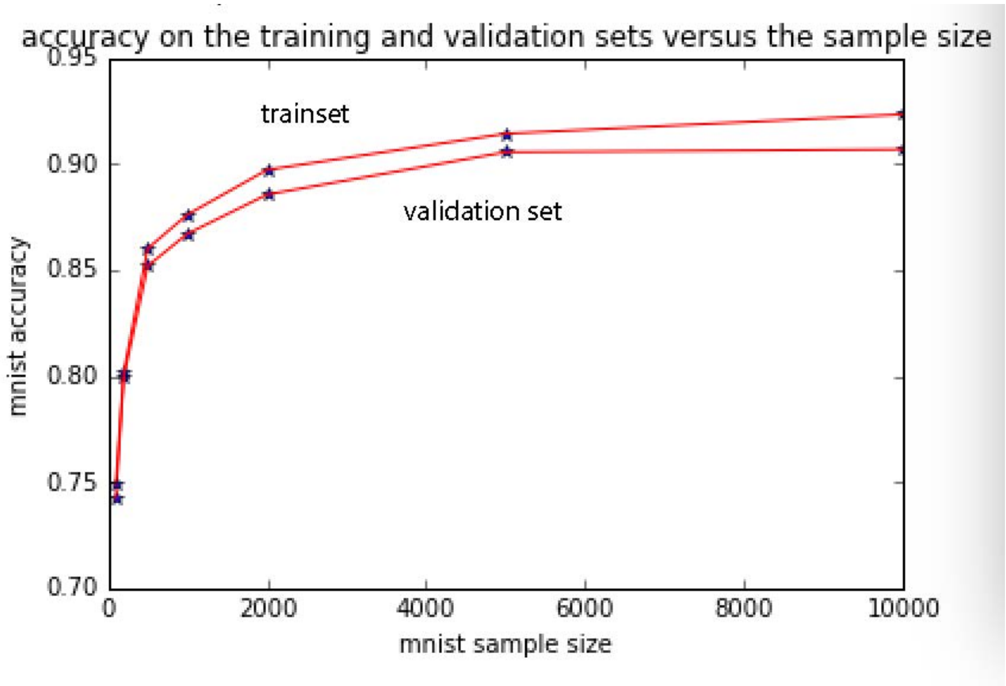
2. Problem 2

Since the method to deal with three data sets are the same, I just write the way and the results for these three data sets.

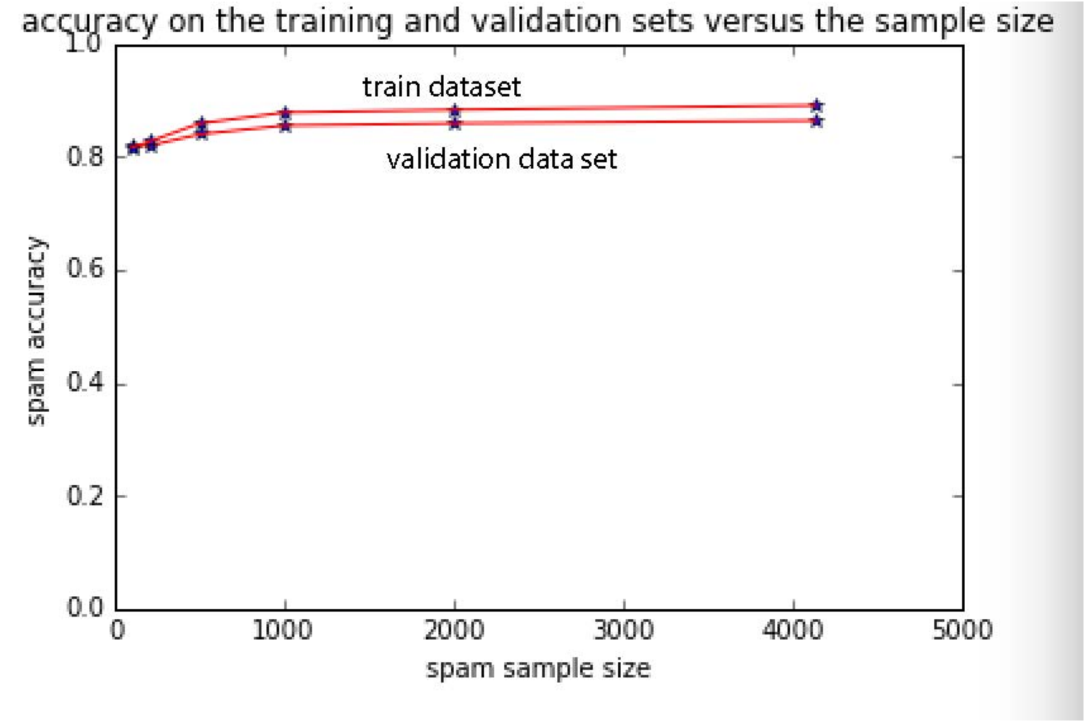
First, I separate the training data sets into two different part: samples and labels. And we do the same thing to the validation data sets.

Then, I use "for loop" for different sample size, and each time we randomly choose certain number of samples and its corresponding labels from training data sets as well as validation data sets. And we use svm.SVC to build the model, let the selected samples and labels from the training data sets to fit the model. Finally, we utilize the score function, use selected samples and labels from validation data sets,to get the accuracy of the model.

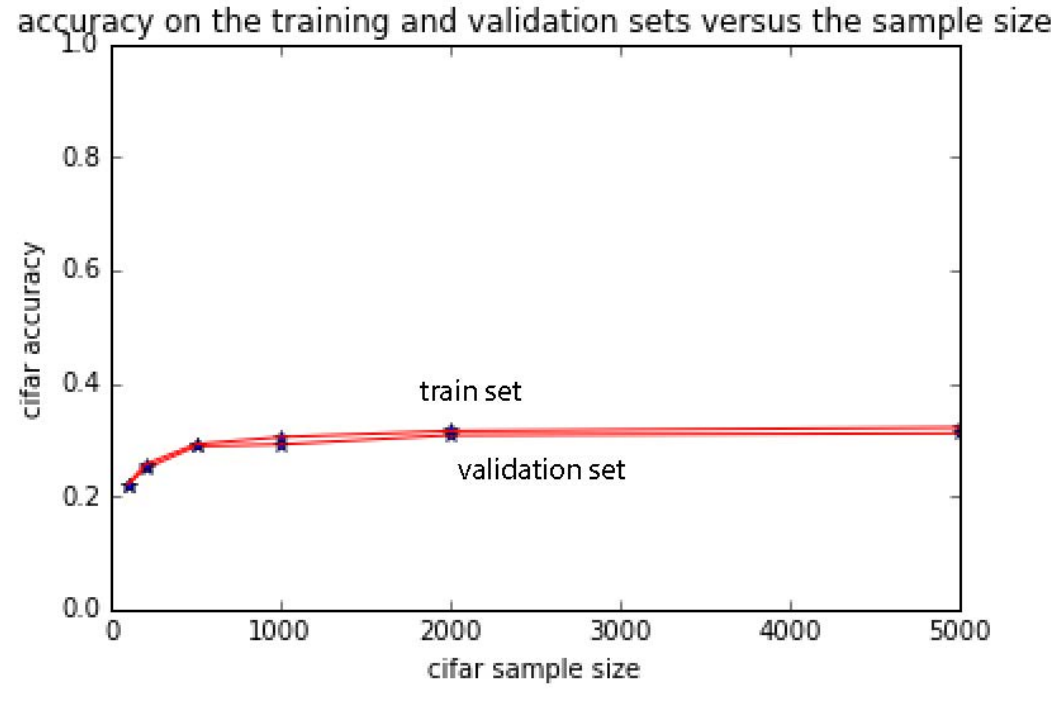
In addition, for the spam, I choose more features and get the new data set. Then I use the new data set to do the same thing.
I plot the error rate on the training and validation sets versus the number of training examples that you used to train your classifier. The pictures are:



picture1: accuracy for mnist data set.



picture2： accuracy of spam data set



picture3： accuracy of cifar10 data set

3. Problem 3

best C value	tried C value	accuracy
0.000001	0.00000001	0.68260000000000005
0.000001	0.0000001	0.92429999999999997
0.000001	0.000001	0.92949999999999995
0.000001	0.00001	0.92570000000000003
0.000001	0.0001	0.92570000000000003
0.000001	0.1	0.92570000000000003
0.000001	1	0.92570000000000003
0.000001	10	0.92570000000000003
0.000001	100	0.92570000000000003
0.000001	1000	0.92570000000000003

so we can see the best C is around 10^{-6}

4. Problem 4

By utilizing K-Folds cross-validator, I try different C to get corresponding accuracies. (notice that I have use the new dataset whose features are decided by myself) So the best value will achieve when C is around 1.

best C value	tried C value	accuracy
1	0.00001	0.70309477756286265
1	0.0001	0.72630560928433274
1	0.001	0.77369439071566726
1	0.01	0.82398452611218564
1	0.1	0.8646034816247582
1	1	0.87427466150870403
1	10	0.86943907156673117
1	50	0.87227466156510403

5. Problem 5

I use predict function on test data set, and get the result.
My kaggle name is: Shuhui Huang.
And my kaggele score is:

Mnist : 0.93120

Spam : 0.84392

In [ ]:

```
# -*- coding: utf-8 -*-
#CS289 homework1

## import the functions will be used later
import scipy.io as sio
import numpy as np
import random
from sklearn import svm
import matplotlib.pyplot as plt
## import the data set

mnist_contents=sio.loadmat('/Users/huangshuhui/Desktop/hw01_data/mnist/train.mat
')
mnist_testdata=sio.loadmat('/Users/huangshuhui/Desktop/hw01_data/mnist/test.mat'
)['testX']
mnist_train=mnist_contents['trainX']

spam_contents=sio.loadmat('/Users/huangshuhui/Desktop/hw01_data/spam/spam_data.m
at')
spam_test=spam_contents['test_data']
spam_train=spam_contents['training_data']
spam_trainlabel=spam_contents['training_labels']

cifar_contents=sio.loadmat('/Users/huangshuhui/Desktop/hw01_data/cifar/train.mat
')
cifar_train=cifar_contents['trainX']
cifar_testdata=sio.loadmat('/Users/huangshuhui/Desktop/hw01_data/cifar/test.mat'
)['testX']
## problem1: Data Partitioning: 10,000 training images for mnist, 20% training s
amples for spam, 5,000 training images for cifar

mnist_range=range(0,len(mnist_train))
mnist_sample=random.sample(mnist_range,10000)
mnist_validation=np.array(mnist_train[mnist_sample])
mnist_rtrain=np.delete(mnist_train,mnist_sample,0)
mnist_label=mnist_rtrain[...,784]
mnist_labelval=mnist_validation[...,784]

spam_length=len(spam_train)
spam_range=range(0,spam_length)
spam_sample=random.sample(spam_range,int(spam_length/5))
spam_validation=np.array(spam_train[spam_sample])
spam_rtrain=np.delete(spam_train,spam_sample,0)
spam_labelval=np.array(spam_trainlabel[0][spam_sample])
spam_rlabel=np.delete(spam_trainlabel,spam_sample,1)

cifar_range=range(0,len(cifar_train))
cifar_sample=random.sample(cifar_range,5000)
cifar_validation=np.array(cifar_train[cifar_sample])
cifar_rtrain=np.delete(cifar_train,cifar_sample,0)
```

```
cifar_label=cifar_rtrain[...,3072]
cifar_labelval=cifar_validation[...,3072]

## problem2: Train a linear SVM on all three datasets

## for mnist data set
mnist_traindata=mnist_rtrain[...,0:784]
mnist_validationdata=mnist_validation[...,0:784]
mnist_number=[100,200,500,1000,2000,5000,10000]

#sample and label from traing dataset
msample=[0,1,2,3,4,5,6]
mlabel=[0,1,2,3,4,5,6]

#fitted model
mmodel=[0,1,2,3,4,5,6]

#the accuarcy the mnist
mnist_validscore=[0,1,2,3,4,5,6]
mnist_trainscore=[0,1,2,3,4,5,6]

# using svm to get the accuracy of mnist
for i in range(0,7):
       sample=random.sample(range(0,50000),mnist_number[i])
       msample[i]=np.array(mnist_traindata[sample])
       mlabel[i]=np.array(mnist_label[sample])
       mmodel[i]=svm.SVC(kernel="linear")
       mmodel[i].fit(msample[i],mlabel[i])
       mnist_validscore[i]=mmodel[i].score(mnist_validationdata,mnist_labelval)
       mnist_trainscore[i]=mmodel[i].score(mnist_traindata,mnist_label)

# plot the errorrate versus the number of training example
plt.plot(mnist_number,mnist_validscore,'b*')
plt.plot(mnist_number,mnist_validscore,'r')
plt.plot(mnist_number,mnist_trainscore,'b*')
plt.plot(mnist_number,mnist_trainscore,'r')
plt.ylabel('mnist accuracy')
plt.xlabel('mnist sample size')
plt.title('accuracy on the training and validation sets versus the sample size')


## for the spam data set
spam_number=[100,200,500,1000,2000,4138]

#sample and label from traing dataset
ssample=[0,1,2,3,4,5]
slabel=[0,1,2,3,4,5]

#fitted model
smodel=[0,1,2,3,4,5]

#the accuracy of spam
spam_validscore=[0,1,2,3,4,5]
```

```
spam_trainscore=[0,1,2,3,4,5]


# using svm to get the accuracy of spam
for i in range(0,6):
       sample=random.sample(range(0,len(spam_rtrain)),spam_number[i])
       ssample[i]=np.array(spam_rtrain[sample])
       slabel[i]=np.array(spam_rlabel[0][sample])
       smodel[i]=svm.SVC(kernel="linear")
       smodel[i].fit(ssample[i],slabel[i])
       spam_validscore[i]=smodel[i].score(spam_validation,spam_labelval)
       spam_trainscore[i]=smodel[i].score(spam_rtrain,spam_rlabel[0])

# plot the errorrate versus the number of training example
plt.plot(spam_number,spam_validscore,'b*')
plt.plot(spam_number,spam_validscore, 'r')
plt.plot(spam_number,spam_trainscore,'b*')
plt.plot(spam_number,spam_trainscore, 'r')
plt.ylabel('spam accuracy')
plt.xlabel('spam sample size')
plt.title('accuracy on the training and validation sets versus the sample size')
plt.ylim(0,1)
plt.xlim(0,5000)

## for Cifar10 data set
cifar_traindata=cifar_rtrain[...,0:3072]
cifar_validationdata=cifar_validation[...,0:3072]
cifar_number=[100,200,500,1000,2000,5000]

#sample and label from traing dataset
csample=[0,1,2,3,4,5]
clabel=[0,1,2,3,4,5]

#fitted model
cmodel=[0,1,2,3,4,5]

#the accuracy of cifar
cifar_validscore=[0,1,2,3,4,5]
cifar_trainscore=[0,1,2,3,4,5]

# using svm to get the accuracy of cifar
for i in range(0,6):
       sample=random.sample(range(0,45000),cifar_number[i])
       csample[i]=np.array(cifar_traindata[sample])
       clabel[i]=np.array(cifar_label[sample])
       cmodel[i]=svm.SVC(kernel="linear")
       cmodel[i].fit(csample[i],clabel[i])
       cifar_validscore[i]=cmodel[i].score(cifar_validationdata,cifar_labelval)
       cifar_trainscore[i]=cmodel[i].score(cifar_traindata,cifar_label)
# plot the errorrate versus the number of training example
plt.plot(cifar_number,cifar_validscore,'b*')
plt.plot(cifar_number,cifar_validscore, 'r')
plt.plot(cifar_number,cifar_trainscore,'b*')
plt.plot(cifar_number,cifar_trainscore, 'r')
plt.ylabel('cifar accuracy')

plt.xlabel('cifar sample size')
plt.title('accuracy on the training and validation sets versus the sample size')
plt.ylim(0,1)
plt.xlim(0,5000)

## problem3: THyperparameter Tuning, find the best C value for mnist data set.
def accuracy(cvalue):
    mmodel[6]=svm.SVC(C=cvalue,kernel="linear")
    mmodel[6].fit(msample[6],mlabel[6])
    mnist_validscore[6]=mmodel[6].score(mnist_validationdata,mnist_labelval)
    return mnist_validscore[6]

accuracy(0.00000001)
accuracy(0.0000001)
accuracy(0.000001)
accuracy(0.00001)
accuracy(0.0001)
accuracy(0.1)
accuracy(1)
accuracy(10)
accuracy(100)

## problem 4: K-Fold Cross-Validation
from sklearn.model_selection import KFold,GridSearchCV

# make sample be able to split
kfsample=random.sample(range(0,5172),5170)
spam_kftrainraw=spam_train[kfsample]
spam_kflabelraw=spam_trainlabel[0][kfsample]

#use k-fold to get the train and test set
kf = KFold(n_splits=5,shuffle=True)
svr = svm.SVC(kernel="linear")

#get the accuracy of the model
def kfspam_accuracy(cvalue):
    p_grid = {"C": [cvalue]}
    clf = GridSearchCV(estimator=svr,param_grid=p_grid, cv=kf)
    clf.fit(spam_kftrainraw, spam_kflabelraw)
    spam_kfscore=clf.score(spam_validation,spam_labelval)
    return spam_kfscore


kfspam_accuracy(0.00001)
kfspam_accuracy(0.0001)
kfspam_accuracy(0.001)
kfspam_accuracy(0.01)
kfspam_accuracy(0.1)
kfspam_accuracy(1)
kfspam_accuracy(10)
kfspam_accuracy(50)
```

```
## problem 5:

#the best model for mnist
mbestnumber=random.sample(range(0,50000),10000)
mbestsample=np.array(mnist_traindata[mbestnumber])
mbestlabel=np.array(mnist_label[mbestnumber])

mnist_bestmodel=svm.SVC(C=0.000001,kernel="linear")
mnist_bestmodel.fit(mbestsample,mbestlabel)

mnist_predict=mnist_bestmodel.predict(mnist_testdata)
np.savetxt('mnist.csv', mnist_predict, delimiter = ',')

#the best model for spam
spam_contents=sio.loadmat('/Users/huangshuhui/Desktop/study/CS289/hw2017/hw1/hw0
1_data/spam/spam_data1.mat')
spam_test=spam_contents['test_data']
spam_train=spam_contents['training_data']
spam_trainlabel=spam_contents['training_labels']

sbestnumber=random.sample(range(0,len(spam_train)),5170)
sbestsample=np.array(spam_train[sbestnumber])
sbestlabel=np.array(spam_trainlabel[0][sbestnumber])

kf = KFold(n_splits=5,shuffle=True)
svr = svm.SVC(kernel="linear")

p_grid = {"C": [1]}
spam_bestmodel = GridSearchCV(estimator=svr,param_grid=p_grid, cv=kf)
spam_bestmodel.fit(sbestsample, sbestlabel)

spam_predict=spam_bestmodel.predict(spam_test)
np.savetxt('spam.csv', spam_predict, delimiter = ',')

#the best model for cifar
cbestnumber=random.sample(range(0,45000),5000)
cbestsample=np.array(cifar_traindata[cbestnumber])
cbestlabel=np.array(cifar_label[cbestnumber])

cifar_bestmodel=svm.SVC(kernel="linear")
cifar_bestmodel.fit(cbestsample,cbestlabel)

cifar_predict=cifar_bestmodel.predict(cifar_testdata)
np.savetxt('cifar.csv', cifar_predict, delimiter = ',')
```