COMS 4772 Fall 2015: Homework #1

Daniel M Sheehan - dms2203@columbia.edu

Discussants: tudor

February 10, 2016

Problem 1

```
from scipy.io import loadmat
ocr = loadmat('ocr.mat')

import matplotlib.pyplot as plt
from matplotlib import cm

import random
import numpy as np

import datetime
import csv

def dist(x, y):
    diff = x.astype(np.float32) - y.astype(np.float32)
    return diff.dot(diff)

def nnClassifier(trainingData, trainingLabels, testData, testLabels):
    total_count = len(testLabels)
    wrong_count = 0
    for testVector, testLabel in zip(testData, testLabels):
        # find the closest training vector
        minDist = float('inf')
        minLabel = None
        for trainingVector, trainingLabel in zip(trainingData, trainingLabels):
            d = dist(testVector, trainingVector)
            #print 'd:', d, 'minDist:', minDist
            if d < minDist:
                minDist = d
                minLabel = trainingLabel

        if minLabel[0] != testLabel[0]:
            wrong_count += 1

    print 'wrongcount:', wrong_count
    errorRate = float(wrong_count) / total_count
    print errorRate
    return errorRate

testData = ocr['testdata']
testLabels = ocr['testlabels']

sample_sizes = [1000,2000,4000,8000]

for i in range(10):
    dt = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    dt = dt.replace(' ','-').replace(':','-')
    print dt

    error_rates = []
    for n in sample_sizes:
        print n
        sel = random.sample(xrange(60000),n)
        trainingData = ocr['data'][sel]
        trainingLabels = ocr['labels'][sel]
        e = nnClassifier(trainingData, trainingLabels, testData, testLabels)
        error_rates.append(e)

    print error_rates

    with open("data/output"+dt+".csv", "wb") as f:
        writer = csv.writer(f)
        row = sample_sizes, error_rates
        writer.writerows(row)


import glob, os
import pandas as pd

theFiles = glob.glob('data/*.csv')
df_list = []

for i in theFiles:
    df = pd.read_csv(i)
    df = df.T
    df.columns = ['error_rate']
    df['date'] = i.replace('data/output','').replace('.csv','')
    df['samp'] = df.index
    df_list.append(df)

df = pd.concat(df_list)
df.head(5)
```

	error_rate	date	samp
1000	0.1144	2016-02-05-11-00-00	1000
2000	0.0894	2016-02-05-11-00-00	2000
4000	0.0702	2016-02-05-11-00-00	4000
8000	0.0534	2016-02-05-11-00-00	8000
1000	0.1119	2016-02-05-11-41-39	1000

```
df['count'] = 1
dfg = df.groupby(['samp'],as_index=False).sum()
dfg['error_rate'] = dfg['error_rate'] / dfg['count']
dfg.head(10)
```

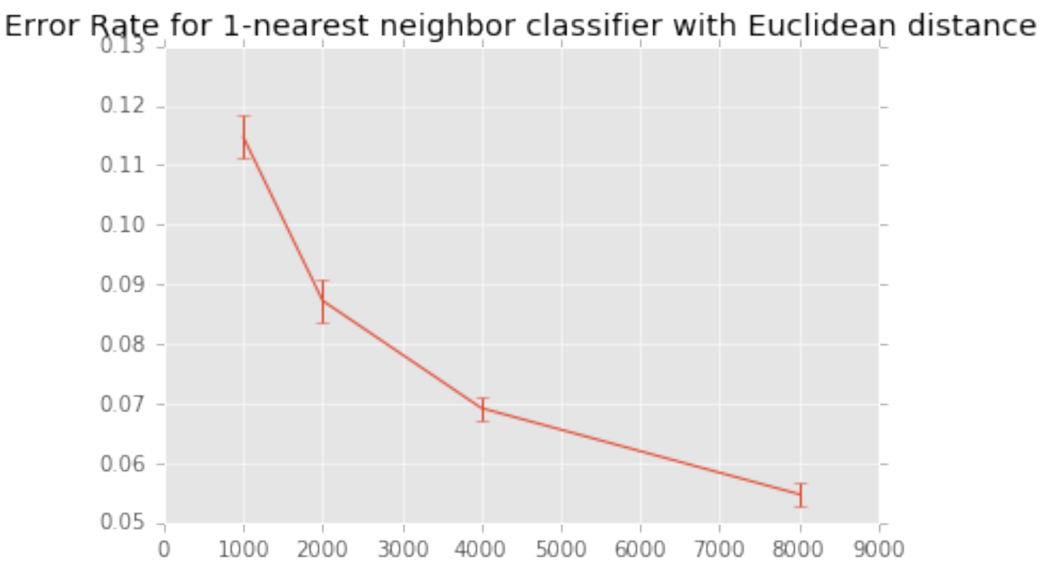
	samp	error_rate	count
0	1000	0.11471	10
1	2000	0.08724	10
2	4000	0.06922	10
3	8000	0.05479	10

```
dfgby = df.groupby(['samp']).std()
dfgby.head(10)
```

	error_rate	count
samp		
1000	0.003527	0
2000	0.003654	0
4000	0.001981	0
8000	0.002001	0

```
import matplotlib.pyplot as plt
plt.style.use('ggplot')
%matplotlib inline

#!/usr/bin/env python
import numpy as np
import matplotlib.pyplot as plt

x = dfg['samp']
y = dfg['error_rate']

theError = dfgby['error_rate']

# First illustrate basic pyplot interface, using defaults where possible.
plt.figure()
plt.errorbar(x, y, yerr=theError)#, xerr=0.2)
plt.title("Error Rate for 1-nearest neighbor classifier with Euclidean distance")
plt.ylim(ymax=0.13)
plt.xlim(xmin=0, xmax=9000)
```

```
(0, 9000)
```



Problem 2

$P(X \mid class) = \prod_{j=1}^{d} \mu_j^{x_j} (1 - \mu_j)^{x_j}$

$L = log(.....) \rightarrow \left(\frac{\delta L}{\mu_j}\right) = \delta$

$P(motorcycle)\ P(gas)....P(mouse) \leftarrow O$

$\delta/N \rightarrow \left(\frac{1+O}{2+N}\right)$

$= (MLE \mid Laplace)$

$P(Y \mid X)$

$prediction = argmax_{forum}\ P(Y = forum|X)$

$\propto P(X|forum)P(forum)$

$\Rightarrow \prod_{j=1}^{d} \mu_j^{x_j} (1 - \mu_j)^{1-x_j}$ # forum/total

$P(data|\mu) \rightarrow P(data|\mu forum_1)\ , P(data|\mu forum_2)$

$Likelihood = \prod P(word\ appears) \quad P(!word\ appears)$

$argmax_y(\prod_{j=1}^{d} \mu_{y,j}^{x_j}(1 - \mu_{y,j})^{1-x_j})\pi_y$

$Linear\ classification : y = w_o + w_1 x_1 + w_2 x_2 + ...$

$y = mx + b = w_o + \sum_{j=1}^{d} w_j x_j$

$log[(\prod_{j=1}^{d} \mu_{y,j}^{x_j}(1 - \mu_{y,j})^{1-x_j})\pi_y]$

$= log\pi_y + \sum_{j=1}^{d} x_j log\mu_{y,j} + (1 - x_j)log(1 - \mu_{i,j})$

$= log\pi_y + \sum_j log(1 - \mu_{i,j}) + \sum_j [log\mu_{j,y} * log(1 - \mu_{i,j})]x_j$

$= b_y + w_y x_j \rightarrow w_{y(forum\ or\ class),j(dimension\ or\ word\ index)}$

$\prod_{j=1} \mu.....(1 - \mu) \prod_y$

$b + wx(dotproduct)$

```
from scipy.io import loadmat
import numpy as np
import pandas as pd

news = loadmat('news.mat')

def get_params(Xtr, Ytr):
    # finding w and b for each class
    Ytr = Ytr.flatten()
    theClasses = set(Ytr)#list of classes
    totCnt = len(Ytr)
    paramDict = {}
    for theClass in theClasses:
        idx = np.where(Ytr==theClass)[0]
        cnt =  len(idx)
        xForThisClass = Xtr[idx]
        mu = (1 + xForThisClass.sum(axis=0)) / ( 2 + cnt)
        pi = cnt/float(totCnt)
        bias = np.log(pi) + np.log(1 - mu).sum() #lgo pi + sum of all log minus meu
        weights = np.log(mu) - np.log(1 - mu) #log meu, log 1 - meu
        paramDict[theClass] = {'bias':bias, 'weights':weights}
    return paramDict

def get_error_rate(params, Xtest, Ytest):
    total_count = len(Ytest)
    wrong_count = 0
    for testVector, testLabel in zip(Xtest, Ytest):
        #print testVector
        x = testVector
        maxLL = float('-inf')
        minLabel = None
        for key, value in params.iteritems():
            # make a prediction for input testVector
            w = value['weights']
            b = value['bias']
            # print "weights shape:", w.shape, "bias shape:", b.shape, "x shape:", x.shape
            # # np.dot(w, x) #### no! x is not a numpy array
            # # w.dot(x) ###### x is not a numpy array
            # # np.dot(w, x.to_array()) #### slow
            # # w.dot(x.to_array()) #### same
            # # x.dot(w)
            pred = x.dot(w.T) + b #should pred label 1-20?
            # if this prediction != testLabel, increment wrong_count
            if pred > maxLL:
                maxLL = pred
                minLabel = key
        if int(minLabel) != int(testLabel[0]):
            wrong_count += 1

    #print 'wrongcount:', wrong_count
    errorRate = float(wrong_count) / total_count
    #print errorRate
    return errorRate
    #which class has the highest w*x + b

Xtrain = news['data']
Ytrain = news['labels']
params = get_params(Xtrain, Ytrain)
print "Training error:", get_error_rate(params, Xtrain, Ytrain)

Xtest = news['testdata']
Ytest = news['testlabels']
print "Test error:", get_error_rate(params, Xtest, Ytest)

#-----

dfv = pd.read_csv("news.vocab",header=None)
dfv.columns = ['word']
dfv['vuid'] = dfv.index + 1

dfg = pd.read_csv("news.groups",header=None)
dfg.columns = ['topic'] #just delcare col names here
dfg['topic'] = dfg.topic.str.split(' ',1).str[0]
dfg['guid'] = dfg.index + 1
```

```
df = pd.DataFrame(params)
df = df.T
df['guid'] = df.index

#print df.head(20) #print dfv.head(20) #print dfg.head(20)

df = df.merge(dfg, on='guid', how='left')
#print df.head(20)

dfw = df[['weights']]
dfw = dfw['weights'].tolist()

df_list = []

for i, k in enumerate(dfw):
    j = k.tolist()
    df = pd.DataFrame(j)
    df = df.T
    df.columns = ['weights']
    df['guid'] = i + 1
    df['vuid'] = df.index + 1
    df = df.sort('weights', ascending=False).head(20)
    #print df.head(25)
    df_list.append(df)

df = pd.concat(df_list)
df = df.merge(dfv, on='vuid', how='left').merge(dfg, on='guid', how='left')
print df.head(400)

df.to_csv('hw1-2.csv',index=False)

Training error: 0.216256988198
Test error: 0.37601598934
      weights  guid  vuid     word              topic
0    3.854394     1    29      the        alt.atheism
1    2.516800     1    12       of        alt.atheism
2    2.487156     1    30       in        alt.atheism
3    2.375473     1    33       to        alt.atheism
4    2.070139     1    23      and        alt.atheism
5    1.989290     1   233     that        alt.atheism
6    1.755913     1    60       is        alt.atheism
7    1.457848     1   778   writes        alt.atheism
8    1.404548     1    42       it        alt.atheism
9    1.266150     1   474      you        alt.atheism
10   0.903257     1   722      not        alt.atheism
```

```
11   0.883123     1   144       be         alt.atheism
12   0.775212     1   775      edu         alt.atheism
13   0.756061     1    81      for         alt.atheism
14   0.746535     1   251     this         alt.atheism
15   0.671441     1    27      are         alt.atheism
16   0.652956     1   922     have         alt.atheism
17   0.508613     1    51      but         alt.atheism
18   0.490957     1    48       on         alt.atheism
19   0.464613     1   473       if         alt.atheism
20   1.897778     2    29      the       comp.graphics
21   1.539057     2    33       to       comp.graphics
22   1.161351     2    23      and       comp.graphics
23   1.123930     2    30       in       comp.graphics
24   1.105485     2    12       of       comp.graphics
25   0.929536     2    81      for       comp.graphics
26   0.774026     2    60       is       comp.graphics
27   0.596870     2    42       it       comp.graphics
28   0.182322     2   233     that       comp.graphics
29   0.106447     2    48       on       comp.graphics
..        ...   ...   ...      ...                 ...
370  1.070201    19   474      you  talk.politics.misc
371  0.960271    19   251     this  talk.politics.misc
372  0.938933    19   722      not  talk.politics.misc
373  0.917793    19    27      are  talk.politics.misc
374  0.896844    19   770  article  talk.politics.misc
375  0.696368    19   922     have  talk.politics.misc
376  0.657944    19   144       be  talk.politics.misc
377  0.629431    19    48       on  talk.politics.misc
378  0.426971    19    52     with  talk.politics.misc
379  0.400103    19   388       as  talk.politics.misc
380  3.186353    20    29      the  talk.religion.misc
381  2.251292    20    12       of  talk.religion.misc
382  2.191359    20    30       in  talk.religion.misc
383  2.162438    20    33       to  talk.religion.misc
384  1.976494    20    60       is  talk.religion.misc
385  1.904237    20    23      and  talk.religion.misc
386  1.517065    20   233     that  talk.religion.misc
387  1.008897    20    42       it  talk.religion.misc
388  0.942363    20   722      not  talk.religion.misc
389  0.765468    20   474      you  talk.religion.misc
390  0.657640    20   778   writes  talk.religion.misc
391  0.622530    20   922     have  talk.religion.misc
392  0.610909    20    81      for  talk.religion.misc
393  0.599328    20    27      are  talk.religion.misc
394  0.530628    20   251     this  talk.religion.misc
```

```
395  0.519300    20   144        be  talk.religion.misc
396  0.463131    20    48        on  talk.religion.misc
397  0.363747    20    52      with  talk.religion.misc
398  0.352821    20    51       but  talk.religion.misc
399  0.331033    20   770   article  talk.religion.misc

[400 rows x 5 columns]


/usr/local/lib/python2.7/site-packages/IPython/kernel/__main__.py:98: FutureWarning: sort(colu
```

Problem 3

$P(Yes_{Y=1}) = 0.001$

$\pi_1 ='' $

$P(Yes_{Y=0}) = 0.999$

$\pi_0 ='' $

$\pi_1 = 200/300 = 2/3 \ \pi_2 = 100/300 = 1/3$

$not \rightarrow P(X|C_1)\ vs.\ P(X|C_2)$

$V \rightarrow P(C_1|X)\ vs.\ P(C_2|X)$

$P(X|C_1)\pi_1 = P(X|C_2)\pi_2 \ C$

$\qquad * 2/3 \qquad\qquad * 1/3$

$P(X|C_1)\pi_1 = P(X|C_2)\pi_2 C$

$find\ x$

$\pi_0 = 2/3 \ \pi_1 = 1/3$

$$P(X|Y = 0) = N(0, 1) = \frac{1}{\sqrt{2\pi}} e^{-1/2*x^2}$$

$$P(X|Y = 1) = N(1, 1/4) = \frac{1}{\sqrt{2\pi * 1/4}} e^{-1/2*(x-1)^2/(v4)}$$

$P(X|Y=0)\pi_0 = P(X|Y=1)\pi_1 C$

$$f*(x) = \begin{cases} 0 & x < b_1 \\ 1 & b_1 < x < b_2 \\ 0 & x > b_2 \end{cases}$$

Problem 4

a

$C = \{red, orange, yellow, green, blue\}$

$n = 100\ balls$

$n_{red}, n_{orange}, n_{yellow}, n_{green}, n_{blue}$

$p(red) = n_{red}/n$

$p(red, red) = n_{red}/n * n_{red}/n$

$p(differentcolor) = 1 - p(samecolor)$

$= 1 - [p(red, red) + p(orange, orange) + p(yellow, yellow) + p(green, green) + p(blue, blue)]$

$= 1 - \sum_{c \in C} (n_c/n)^2$

$= 1 - \sum_c (p_c)^2$

b

$entropyH = -\{p\ log\ p + (1-p) log(1-p)\}$

$max\ H\ wrt\ p \rightarrow p = 0.5$

$H = -\sum_k P_k logpk$

Paint each color 20 times.

$\sigma(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots) \rightarrow [0, 1]$

$P(\ Y_{class} = 1 \mid X_{data})$

$posterior$

$$P(Y=1|X)=\frac{P(X,Y=1)}{P(X)}$$

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

$$\propto P(X|Y=1)P(Y)$$

$likelihood-prior$

$P(Yes_{Y=1})=0.001$

$\pi_1 ='' $

$P(Yes_{Y=0})=0.999$

$\pi_0 ='' $

$\pi_1 = 200/300 = 2/3 \ \pi_2 = 100/300 = 1/3$

$not \rightarrow P(X|C_1) \ vs. P(X|C_2)$

$V \rightarrow P(C_1|X) \ vs. P(C_2|X)$

$P(X|C_1)\pi_1 = P(X|C_2)\pi_2 \ C$

$* 2/3 \qquad * 1/3$

$P(X|C_1)\pi_1 = P(X|C_2)\pi_2 C$

$find \ x$

$\pi_0 = 2/3 \ \pi_1 = 1/3$

$$P(X|Y=0)=N(0,1)=\frac{1}{\sqrt{2\pi}}e^{-1/2*x^2}$$

$$P(X|Y=1)=N(1,1/4)=\frac{1}{\sqrt{2\pi*1/4}}e^{-1/2*(x-1)^2/(v4)}$$

$P(X|Y=0)\pi_0 = P(X|Y=1)\pi_1 C$

$$f*(x)=\begin{cases} 0 & x<b_1 \\ 1 & b_1<x<b_2 \\ 0 & x>b_2 \end{cases}$$