**Fall**

15

Machine Learning Assignment 2

Sundari Selvarajan

ss2738

08

**Fall**

**Gradient Descent Algorithm**

**Location:**

/afs/cad.njit.edu/courses/ccs/f15/cs/675/001/ss2738/Assignment2

**File Name:**

gradient\_descent.py

**Argument:**

data file , label file and test\_label file

**Program:**

\_\_author\_\_ = 'Sundu'

import sys

import random

import math

################# Reading Data and Label Sets #############################

input\_file = sys.argv[1]

with open(input\_file) as f:

data = f.read()

dataset = []

dataset1 = []

for line in data.split("\n"):

if line:

# print line

dataset = []

a = line.split()

dataset.append('1')

# print "len:",len(a)

for i in range(len(a)):

dataset.append(a[i])

# print dataset

dataset1.append(dataset)

# dataset.append([float(x) for x in line.split()])

# print dataset1

train\_file = sys.argv[2]

with open(train\_file) as f:

data = f.read()

labelset = {}

for line in data.split("\n"):

if line:

temp = int(line.split()[0])

# print temp

if temp != 0 and temp in labelset:

labelset[temp].append(int(line.split()[1]))

elif (temp == 0 and -1 in labelset):

labelset[-1].append(int(line.split()[1]))

else:

labelset[-1 if temp == 0 else int(line.split()[0])] = [int(line.split()[1])]

# print labelset

# print dataset

# print labelset

##################### Step 1: Initialize w , w belongs to [0,1] ##############

l = len(a)

w = []

for i in range(l):

w.append(random.random())

# for i in range(l):

# print w[i]

################ Step 2 : Absorbing w0 into w and append 1 in train\_data ####

w0 = random.random()

w1 = [w0]

for i in w:

w1.append(i)

################ Compute obj = sum(y(i)-transpose(x(i))\*w)^2 ############

y = []

for i in range(0, len(dataset1) - 1, 1):

y.append(0)

for i in range(0, len(dataset1) - 1, 1):

for k, v in labelset.items():

# print k

for j in range(len(v)):

# print v[j]

if i == v[j]:

y[v[j]] = k

z1 = []

for i in range(len(dataset1) - 1):

z1.append(0.0)

t1 = 0

for j in range(1, len(dataset1) - 1, 1):

for i in range(len(w1)):

t1 += w1[i] \* float(dataset1[j - 1][i])

z1[j] = t1

t1 = 0

t1 = 0

for j in range(1, len(dataset1) - 1, 1):

t1 += (y[j] - z1[j]) \*\* 2

####################### Prev obj #############################################

prev\_obj = float("Inf")

################## Update obj ###############################################

f = []

sum2 = []

sum4 = []

eta = 0.001

for i in range(len(dataset1)):

sum2.append(0)

for i in range(len(w1)):

sum4.append(0)

#print dataset1

#print "t1:", t1

while (prev\_obj - t1) > 0.0:

prev\_obj = t1

const = 0

for j in range(1, len(dataset1) - 1, 1):

sum2[j] += y[j] - z1[j]

temp = 0

for i in range(len(w1)):

for j in range(1, len(dataset1) - 1, 1):

temp += sum2[j] \* (-2) \* float(dataset1[j][i])

sum4[i] = temp

temp = 0

# print sum2[i]

for i in range(len(w1)):

#print sum4[i]

w1[i] -= (eta \* sum4[i])

#print w1

z1 = []

for i in range(len(dataset1) - 1):

z1.append(0.0)

t1 = 0

for j in range(1, len(dataset1) - 1, 1):

for i in range(len(w1)):

t1 += w1[i] \* float(dataset1[j][i])

z1[j] = t1

t1 = 0

t1 = 0

for j in range(1, len(dataset1) - 1, 1):

t1 += (y[j] - z1[j]) \*\* 2

sum2 = []

for i in range(len(dataset1) - 1):

sum2.append(0)

sum4 = []

for i in range(len(w1)):

sum4.append(0)

# print sum1

#print t1

# print t1

print ("Weight:", w1)

sum3 = 0.0

for i in range(1, len(w1), 1):

#print w1[i]

sum3 += (w1[i] \*\* 2)

#print abs

sum3 = math.sqrt(sum3)

#print sum3

dist = 0

dist = w1[0] / sum3

print ("absolute distance:",abs(dist))

####################Prediction################################################

name = sys.argv[1] + ".predictions"

op = open(name, "w")

test\_data = sys.argv[3]

with open(test\_data) as f:

data\_file = f.read()

# data\_array=array('i')

data = []

data1 = []

count = 0

for line in data\_file.split("\n"):

count += 1

if line:

# print line

a = line.split()

for i in range(len(a)):

data1.append(float(a[i]))

data1.append(1)

data.append(data1)

data1=[]

#print "count", count

#print "data:", data

rows = len(data)

#print rows, "rows"

# cols = len(data[0])

# print cols

#prod = []

y = []

for i in range(0, len(data) - 1, 1):

y.append(0)

#print "Y:", y

z1 = []

t2 = 0

# t3=[]

#print "dataset:", data

#print "w", w1

q = 0

#print "len od data",len(data)

for j in range(0, len(data), 1):

for i in range(len(w1)):

t2 += w1[i] \* float(data[j][i])

#print "t2",t2

#print q

z1.append(t2)

t2 = 0

#print "t2:", t2

#print "t3:", z1

for i in z1:

if i > 0:

pred = 1

#print "1"

else:

pred = -1

#print "-1"

op.write(str(pred)+"\t"+str(j)+"\n")

j+=1

# print (data,cls)

**Coordinate Descent Algorithm**

**Location:**

/afs/cad.njit.edu/courses/ccs/f15/cs/675/001/ss2738/Assignment2

**File Name:**

coordinate\_descent.py

**Argument:**

data file , label file and test\_label file

**Program:**

\_\_author\_\_ = 'Sundu'

from sys import argv

from array import array

import random

from math import copysign as sign

import pdb

import math

import copy

def dot(a, b):

res = float(0)

for i in range(0, len(a)):

res += (a[i] \* b[i])

#print res

return res

def find\_alpha(data, label):

x = sorted([a \* b for a, b in zip(data, label)])

######### FIND WHERE THE SIGN CHANGES IN x###############################

for i in range(len(x)):

if x[i + 1] - x[i] >= 0.0:

alpha = (x[i] + x[i + 1]) / 2

return alpha

return 0

data\_path = argv[1]

label\_path = argv[2]

############## READ THE DATA IN AN ARRAY##################################

# data\_file = open(data\_path)

with open(data\_path) as f:

data\_file = f.read()

data = []

for line in data\_file.split("\n"):

if line:

dataset = array('f')

a = line.split()

for i in range(len(a)):

dataset.append(float(a[i]))

dataset.append(1)

data.append(dataset)

#print data

### READ DATA AND ADD 1 TO THE END OF EACH ROW.#############################

rows = len(data) - 1

cols = len(data[0])

##### READ THE LABELS IN AN ARRAY

with open(label\_path) as f:

label\_file = f.read()

labelset = array('i')

for line in label\_file.split("\n"):

if line:

temp = int(line.split()[0])

if temp == 0:

labelset.append(-1)

else:

labelset.append(temp)

#print labelset

##### Starting The Coordinate Descent######################################

W = array('f')

w\_new = array('f')

d = array('f')

############### INITIALIZE W TO A RANDOM PLANE.############################

for i in range(cols):

W.append(random.random())

w\_new.append(0.0)

for j in range(cols):

d.append(0.0)

delta = float(0)

data\_prim = array('f')

label\_prim = array('i')

alpha = 0

prev\_error = 0

error = 10000000

stop\_condition = 100

while (stop\_condition != 0 or err != 0):

prev\_error = error

for j in range(0, cols):

d[j] = 1

for i in range(0, rows):

#print i

delta = dot(d, data[i])

if (delta != 0):

####CREATE THE NEW DATA AND APPEND IT TO DATA\_PRIME AND LABEL\_PRIME#######

data\_prim.append(dot(W, data[i]))

label\_prim.append(int(sign(float(labelset[i]), delta)))

#print data\_prim

#print label\_prim

alpha = find\_alpha(data\_prim, label\_prim)

error = 0

### CHANGE W AND COMPUTE ERROR

for k in range(cols):

w\_new[k] = W[k] + alpha \* d[k]

wtx = 0.0

sign\_array = array('i')

for i in range(rows+1):

wtx = dot(w\_new, data[i])

if (wtx > 0.0):

sign\_array.append(1)

else:

sign\_array.append(-1)

err=dot(labelset,sign\_array)

print("Error:",err)

if err < prev\_error:

W = array('f', w\_new)

d[j] = 0

del data\_prim[:]

del label\_prim[:]

stop\_condition -= 1

###### ASSIGN THE CORRECT CLASS TO THE TEST DATA BASEDD ON THE SIGNS#######

name = argv[1] + ".predictions"

op = open(name, "w")

test\_data = argv[3]

with open(test\_data) as f:

data\_file = f.read()

data = []

for line in data\_file.split("\n"):

if line:

dataset = array('f')

a = line.split()

for i in range(len(a)):

dataset.append(float(a[i]))

dataset.append(1)

data.append(dataset)

#print "data:",data

rows = len(data)

#print rows,"rows"

cols = len(data[0])

#print cols

wxi = []

for i in range(rows):

#print W

#print data[i]

wxi.append(dot(W, data[i]))

j=0

for i in wxi:

if i > 0:

pred = 1

#print "1"

else:

pred=-1

#print "-1"

op.write(str(pred)+"\t"+str(j)+"\n")

j+=1