MIDS Machine Learning at Scale MidTerm exam, Week 8, Spring, 2016

Exam location is at: https://www.dropbox.com/s/jdkkttnwd88uxkl/MIDS-MLS-MidTerm-2016-Spring-Live.txt?dl=0 (https://www.dropbox.com/s/jdkkttnwd88uxkl/MIDS-MLS-MidTerm-2016-Spring-Live.txt?dl=0) ===Instructions for midterm===

Please use your live session time from week 8 to complete this mid term (plus an additional 30 minutes if you need it). This is an open book exam meaning you can consult webpages and textbooks (but not each other or other people). Please complete this exam by yourself.

Please submit your solutions and notebooks via the following form:

```
http://goo.gl/forms/ggNYfRXz0t
```

===Exam durations (All times are in California Time)===

Live Session Group #4 4:00 PM - 6:00 PM (Tuesday) Live Session Group #2 4:00 PM - 6:00 PM (Wednesday) Live Session Group #3 6:30 PM - 8:30 PM (Wednesday)

In [11]:
```
# take care of Jupyter and MRJob weirdnesses
%load_ext autoreload
%autoreload 2
```

=====Exam begins here=====

===Map-Reduce===

MT0.

Which of the following statememts about map-reduce are true? Check all that apply

(a) If you only have 1 computer with 1 computing core, then map-reduce is unlikely to help
(b) If we run map-reduce using N computers, then we will always get at least an N-Fold speedup compared to using 1 computer
(c) Because of network latency and other overhead associated with map-reduce, if we run map-reduce using N computers, then we will get leas than N-Fold speedup compared to using 1 computer

(d) When using map-reduce with gradient descent, we usually use a single machine that accumulates the gradients from each of the map-reduce machines, in order to compute the paramter update for the iterion

(a)(c)(d)

===Order inversion===

MT1.

Suppose you wish to write a MapReduce job that creates normalized word co-occurrence data form a large input text. To ensure that all (potentially many) reducers receive appropriate normalization factors (denominators) in the correct order in their input streams (so as to minimize memory overhead), the mapper should emit according to which pattern:

(a) emit (*,word) count*
(b) There is no need to use order inversion here
(c) emit (word,) count
(d) None of the above

(a)

===Apriori principle===

MT2.

When searching for frequent itemsets with the Apriori algorithm (using a threshold, N), the Apriori principle allows us to avoid tracking the occurrences of the itemset {A,B,C} provided

(a) all subsets of {A,B,C} occur less than N times.
(b) any pair of {A,B,C} occurs less than N times.
(c) any subset of {A,B,C} occurs less than N times.
(d) All of the above

(c)

===Bayesian document classification===

MT3.

When building a Bayesian document classifier, Laplace smoothing serves what purpose?

(a) It allows you to use your training data as your validation data.
(b) It prevents zero-products in the posterior distribution.
(c) It accounts for words that were missed by regular expressions.
(d) None of the above

(b)

===Bias-variance tradeoff===

MT4.

By increasing the complexity of a model regressed on some samples of data, it is likely that the ensemble will exhibit which of the following?

(a) Increased variance and bias
(b) Increased variance and decreased bias
(c) Decreased variance and bias
(d) Decreased variance and increased bias

(b)

===Combiners===

MT5.

Combiners can be integral to the successful utilization of the Hadoop shuffle. This utility is as a result of

(a) minimization of reducer workload
(b) both (a) and (c)
(c) minimization of network traffic
(d) none of the above

(c) Reducer workload doesn't really change; it's the amount of data in the shuffle that changes

===Pairwise similarity using K-L divergence===

In probability theory and information theory, the Kullback–Leibler divergence (also information divergence, information gain, relative entropy, KLIC, or KL divergence) is a non-symmetric measure of the difference between two probability distributions P and Q. Specifically, the Kullback–Leibler divergence of Q from P, denoted DKL(P‖Q), is a measure of the information lost when Q is used to approximate P:

For discrete probability distributions P and Q, the Kullback–Leibler divergence of Q from P is defined to be

KLDistance(P, Q) = Sum over i (P(i) log (P(i) / Q(i))

$$\text{KLDistance}(P, Q) = \sum_i P_i log\left(\frac{P_i}{Q_i}\right)$$

In the extreme cases, the KL Divergence is 1 when P and Q are maximally different and is 0 when the two distributions are exactly the same (follow the same distribution).

For more information on K-L Divergence see:

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence (https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence)

For the next three question we will use an MRjob class for calculating pairwise similarity using K-L Divergence as the similarity measure:

Job 1: create inverted index (assume just two objects) Job 2: calculate/accumulate the similarity of each pair of objects using K-L Divergence

Download the following notebook and then fill in the code for the first reducer to calculate the K-L divergence of objects (letter documents) in line1 and line2, i.e., KLD(Line1||line2).

Here we ignore characters which are not alphabetical. And all alphabetical characters are lower-cased in the first mapper.

http://nbviewer.ipython.org/urls/dl.dropbox.com/s/9onx4c2dujtkgd7/Kullback%E2%80%93Leibler MIDS-Midterm.ipynb (http://nbviewer.ipython.org/urls/dl.dropbox.com/s/9onx4c2dujtkgd7/Kullback%E2%80%93Leibler MIDS-Midterm.ipynb)
https://www.dropbox.com/s/zr9xfhwakrxz9hc/Kullback%E2%80%93Leibler%20divergence-MIDS-Midterm.ipynb?dl=0 (https://www.dropbox.com/s/zr9xfhwakrxz9hc/Kullback%E2%80%93Leibler%20divergence-MIDS-Midterm.ipynb?dl=0)

Using the MRJob Class below calculate the KL divergence of the following two objects.

In [1]:
```
%%writefile kltext.txt
1.Data Science is an interdisciplinary field about processes and systems
2.Machine learning is a subfield of computer science[1] that evolved from
```

```
Writing kltext.txt
```

MRjob class for calculating pairwise similarity using K-L Divergence as the similarity measure

Job 1: create inverted index (assume just two objects)

Job 2: calculate the similarity of each pair of objects

In [2]:
```
import numpy as np
np.log(3)
```

Out[2]:
```
1.0986122886681098
```

In [32]:
```
%%writefile kldivergence.py
from mrjob.job import MRJob, MRStep
import re
import numpy as np
class kldivergence(MRJob):
    def mapper1(self, _, line):
        index = int(line.split('.',1)[0])
        letter_list = re.sub(r"[^A-Za-z]+", '', line).lower()
        count = {}
        for l in letter_list:
            if count.has_key(l):
                count[l] += 1
            else:
                count[l] = 1
        for key in count:
            yield key, [index, (count[key] + 1)*1.0/len(letter_list)+24]
#             yield key, [index, (count[key])*1.0/len(letter_list)]


    def reducer1(self, key, values):
        # assuming 2 objects, '1' and '2' where P => 1 and Q => 2
        # need to handle getting P or Q first
        obj = {}
        for val in values:
            obj[val[0]] = val[1]

        if obj[1]:
            yield None, float(obj[1])*np.log(float(obj[1])/float(obj[2]))

    def reducer2(self, key, values):
        kl_sum = 0
        for value in values:
            kl_sum = kl_sum + value
        yield None, kl_sum

    def steps(self):
        return [MRStep(mapper=self.mapper1,
                       reducer=self.reducer1),
                MRStep(reducer=self.reducer2)]

if __name__ == '__main__':
    kldivergence.run()
```

```
Overwriting kldivergence.py
```

In [33]:
```
from kldivergence import kldivergence
mr_job = kldivergence(args=['kltext.txt'])
with mr_job.make_runner() as runner:
    runner.run()
    # stream_output: get access of the output
    for line in runner.stream_output():
        print mr_job.parse_output_line(line)
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
```

```
(None, 0.008081913888434489)
```

Questions:

MT6.

Which number below is the closest to the result you get for KLD(Line1||line2)?

(a) 0.7
(b) 0.5
(c) 0.2
(d) 0.1

(d) 0.08088278445318145

MT7.

Which of the following letters are missing from these character vectors?

(a) p and t
(b) k and q
(c) j and q
(d) j and f

(c) also z is missing

MT8.

The KL divergence on multinomials is defined only when they have nonzero entries.

For zero entries, we have to smooth distributions. Suppose we smooth in this way:

(ni+1)/(n+24)

where ni is the count for letter i and n is the total count of all letters. After smoothing, which number below is the closest to the result you get for KLD(Line1||line2)??

(a) 0.08 (b) 0.71 (c) 0.02 (d) 0.11

(c) 0.008081913888434489

===Gradient descent===

MT9.

Which of the following are true statements with respect to gradient descent for machine learning, where alpha is the learning rate. Select all that apply

(a) To make gradient descent converge, we must slowly decrease alpha over time and use a combiner in the context of Hadoop.
(b) Gradient descent is guaranteed to find the global minimum for any function J() regardless of using a combiner or not in the context of Hadoop
(c) Gradient descent can converge even if alpha is kept fixed. (But alpha cannot be too large, or else it may fail to converge.) Combiners will help speed up the process.
(d) For the specific choice of cost function J() used in linear regression, there is no local optima (other than the global optimum).

(c)(d)

===Weighted K-means===

Write a MapReduce job in MRJob to do the training at scale of a weighted K-means algorithm.

You can write your own code or you can use most of the code from the following notebook:

http://nbviewer.ipython.org/urls/dl.dropbox.com/s/kjtdyi10nwmk4ko/MrJobKmeans-MIDS-Midterm.ipynb (http://nbviewer.ipython.org/urls/dl.dropbox.com/s/kjtdyi10nwmk4ko/MrJobKmeans-MIDS-Midterm.ipynb) https://www.dropbox.com/s/kjtdyi10nwmk4ko/MrJobKmeans-MIDS-Midterm.ipynb?dl=0 (https://www.dropbox.com/s/kjtdyi10nwmk4ko/MrJobKmeans-MIDS-Midterm.ipynb?dl=0)

Weight each example as follows using the inverse vector length (Euclidean norm):

weight(X)= 1/||X||,

where ||X|| = SQRT(X.X)= SQRT(X1^2 + X2^2)

Here X is vector made up of X1 and X2.

Using the following data answer the following questions:

https://www.dropbox.com/s/ai1uc3q2ucverly/Kmeandata.csv?dl=0 (https://www.dropbox.com/s/ai1uc3q2ucverly/Kmeandata.csv?dl=0)

In [ ]:

In [61]:
```
%%writefile Kmeans.py
from numpy import argmin, array, random, sqrt
from mrjob.job import MRJob
from mrjob.step import MRJobStep
from itertools import chain

#Calculate find the nearest centroid for data point
def MinDist(datapoint, centroid_points):
    datapoint = array(datapoint)
    centroid_points = array(centroid_points)
    diff = datapoint - centroid_points
    diffsq = diff**2

    distances = (diffsq.sum(axis = 1))**0.5
    # Get the nearest centroid for each instance
    min_idx = argmin(distances)
    return min_idx

#Check whether centroids converge
def stop_criterion(centroid_points_old, centroid_points_new,T):
    oldvalue = list(chain(*centroid_points_old))
    newvalue = list(chain(*centroid_points_new))
    Diff = [abs(x-y) for x, y in zip(oldvalue, newvalue)]
    Flag = True
    for i in Diff:
        if(i>T):
            Flag = False
            break
    return Flag


class MRKmeans(MRJob):
    centroid_points=[]
    k=3
    def steps(self):
        return [
```

```
            MRJobStep(mapper_init = self.mapper_init,
                      mapper=self.mapper,
                      combiner = self.combiner,
                      reducer=self.reducer)
                ]
    #load centroids info from file
    def mapper_init(self):
        self.centroid_points = \
        [map(float,s.split('\n')[0].split(',')) for s in open("/Users/rco
        open('/Users/rcordell/Documents/MIDS/W261/centroids.txt', 'w').cl
    #load data and output the nearest centroid index and data point
    def mapper(self, _, line):
        D = (map(float,line.split(',')))
        idx = MinDist(D,self.centroid_points)
        yield int(idx), (D[0],D[1],1)
    #Combine sum of data points locally
    def combiner(self, idx, inputdata):
        sumx = sumy = num = 0
        for x,y,n in inputdata:
            num = num + n
            sumx = sumx + x
            sumy = sumy + y
        yield int(idx),(sumx,sumy,num)
    #Aggregate sum for each cluster and then calculate the new centroids
    def reducer(self, idx, inputdata):
        centroids = []
        num = [0]*self.k
        distances = 0
        for i in range(self.k):
            centroids.append([0,0])
        for x, y, n in inputdata:
            num[idx] = num[idx] + n
            # compute the weights
            w = 1.0/(sqrt(x**2+y**2))
            # compute the center of the centroid with the weighting
            centroids[idx][0] = centroids[idx][0] + x*w
            centroids[idx][1] = centroids[idx][1] + y*w
        centroids[idx][0] = centroids[idx][0]/num[idx]
        centroids[idx][1] = centroids[idx][1]/num[idx]
        with open('/Users/rcordell/Documents/MIDS/W261/centroids.txt', 'a
            f.writelines(str(centroids[idx][0]) + ',' + str(centroids[idx
        yield idx,(centroids[idx][0],centroids[idx][1])

if __name__ == '__main__':
    MRKmeans.run()
```

```
Overwriting Kmeans.py
```

Driver:

Generate random initial centroids

New Centroids = initial centroids

```
While(1):
  Cacluate new centroids
  stop if new centroids close to old centroids
  Updates centroids
```

In [62]:
```
from numpy import random, array
from Kmeans import MRKmeans, stop_criterion
mr_job = MRKmeans(args=['Kmeandata.csv'])

#Geneate initial centroids
centroid_points = [[0,0],[6,3],[3,6]]
k = 3
with open('centroids.txt', 'w+') as f:
        f.writelines(','.join(str(j) for j in i) + '\n' for i in centroid

# Update centroids iteratively
for i in range(10):
    # save previous centoids to check convergency
    centroid_points_old = centroid_points[:]
    print "iteration"+str(i+1)+":"
    with mr_job.make_runner() as runner:
        runner.run()
        # stream_output: get access of the output
        for line in runner.stream_output():
            key,value =  mr_job.parse_output_line(line)
            print key, value
            centroid_points[key] = value
    print "\n"
    i = i + 1
print "Centroids\n"
print centroid_points
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
```

```
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
```

```
iteration1:
0
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
```

```
 [-0.0007331943253071409, 7.400466102054529e-05]
1 [0.0012130921552015385, 3.483577198768393e-05]
2 [5.536940612443592e-05, 0.0012197451416389817]


iteration2:
0
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
```

```
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-0.0010111032671444866, -6.211389024194747e-06]
1 [0.001001990373166972, -5.227251754464414e-06]
2 [2.6800371010696545e-06, 0.000987163193201444]


iteration3:
0

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-0.0009990009813361815, 1.8786788043542625e-07]
1 [0.0010009890561472592, -4.890140941246349e-06]
2 [1.0308682169001185e-05, 0.0009999468641242585]
```

```
iteration4:
0

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-0.0009990009813361815, 1.8786788043542625e-07]
1 [0.001001990373166972, -5.227251754464414e-06]
2 [1.0627828159704498e-05, 0.0009989444655603243]


iteration5:
0

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
```

```
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-0.0009990009813361815, 1.8786788043542625e-07]
1 [0.001001990373166972, -5.227251754464414e-06]
2 [1.0627828159704498e-05, 0.0009989444655603243]


iteration6:
0

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-0.0009990009813361815, 1.8786788043542625e-07]
1 [0.001001990373166972, -5.227251754464414e-06]
2 [1.0627828159704498e-05, 0.0009989444655603243]


iteration7:
0

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
```

```
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-0.0009990009813361815, 1.8786788043542625e-07]
1 [0.001001990373166972, -5.227251754464414e-06]
2 [1.0627828159704498e-05, 0.0009989444655603243]


iteration8:
0

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-0.0009990009813361815, 1.8786788043542625e-07]
```

```
1 [0.001001990373166972, -5.227251754464414e-06]
2 [1.0627828159704498e-05, 0.0009989444655603243]


iteration9:
0

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols
will be strict by default. It's recommended you run your job with --st
rict-protocols or set up mrjob.conf as described at https://pythonhost
ed.org/mrjob/whats-new.html#ready-for-strict-protocols
(https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoc
ols)
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-0.0009990009813361815, 1.8786788043542625e-07]
1 [0.001001990373166972, -5.227251754464414e-06]
2 [1.0627828159704498e-05, 0.0009989444655603243]


iteration10:
0 [-0.0009990009813361815, 1.8786788043542625e-07]
1 [0.001001990373166972, -5.227251754464414e-06]
2 [1.0627828159704498e-05, 0.0009989444655603243]


Centroids

[[-0.0009990009813361815, 1.8786788043542625e-07], [0.0010019903731669
72, -5.227251754464414e-06], [1.0627828159704498e-05, 0.00099894446556
03243]]
```

I am unable to complete these questions because I was unable to get the python notebook working on my local machine and ran out of time.

Questions:

MT10.

Which result below is the closest to the centroids you got after running your weighted K-means code for 10 iterations?

(a) (-4.0,0.0), (4.0,0.0), (6.0,6.0)
(b) (-4.5,0.0), (4.5,0.0), (0.0,4.5)
(c) (-5.5,0.0), (0.0,0.0), (3.0,3.0)
(d) (-4.5,0.0), (-4.0,0.0), (0.0,4.5)

MT11.

Using the result of the previous question, which number below is the closest to the average weighted distance between each example and its assigned (closest) centroid? The average weighted distance is defined as sum over i (weighted_distance_i) / sum over i (weight_i)

(a) 2.5
(b) 1.5
(c) 0.5
(d) 4.0

MT12.

Which of the following statements are true? Select all that apply. a) Since K-Means is an unsupervised learning algorithm, it cannot overfit the data, and thus it is always better to have as large a number of clusters as is computationally feasible.
b) The standard way of initializing K-means is setting μ1=⋯=μk to be equal to a vector of zeros.
c) For some datasets, the "right" or "correct" value of K (the number of clusters) can be ambiguous, and hard even for a human expert looking carefully at the data to decide.
d) A good way to initialize K-means is to select K (distinct) examples from the training set and set the cluster centroids equal to these selected examples.

(c)(d)

Type *Markdown* and LaTeX: α^2