DATASCI W261: Machine Learning at Scale

MrJob class for Kmeans

If you want to change the code, please edit Kmeans.py directly

In [10]:

```
%%writefile Kmeans.py
from numpy import argmin, array, random
from mrjob.job import MRJob
from mrjob.step import MRJobStep
from itertools import chain

import math

#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.mappe
r,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(',')) f
or s in open("Centroids.txt").readlines()]
        open('Centroids.txt', 'w').close()
    #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)
        '''
        Let's do normalization
        '''
```

```
        normalization = math.sqrt(D[0]*D[0] + D[1]*D[1])
        norm = 1.0/normalization
        yield int(idx), (D[0]*norm,D[1]*norm,norm)
    #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
            centroids[idx][0] = centroids[idx][0] + x
            centroids[idx][1] = centroids[idx][1] + y
        centroids[idx][0] = centroids[idx][0]/num[idx]
        centroids[idx][1] = centroids[idx][1]/num[idx]
        with open('Centroids.txt', 'a') as f:
            f.writelines(str(centroids[idx][0]) + ',' + str(centroids[id
x][1]) + '\n')
        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 [11]:
```
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [12]:
```
from numpy import random, array
from Kmeans import MRKmeans, stop_criterion
mr_job = MRKmeans(args=['Kmeandata.csv', '--file', 'Centroids.txt', '--n
o-strict-protocol'])

#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 centroi
d_points)

# 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.step:MRJobStep has been renamed to MRStep. The old name w
ill be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name w
ill be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name w
ill be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name w
ill be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name w
ill be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name w
ill be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name w
ill be removed in v0.5.0.

iteration1:
0

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

 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


iteration2:
0
```

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

```
 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


iteration3:
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.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
```

```
 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


iteration4:
0
```

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

 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


iteration5:
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.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


iteration6:
0
```

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

 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


iteration7:
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.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.

 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


iteration8:
0
```

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

```
 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


iteration9:
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.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old name
will be removed in v0.5.0.
```

```
 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


iteration10:
0 [-2.6816121341554244, 0.4387800225117981]
1 [5.203939274722273, 0.18108381085421293]
2 [0.2798236662882328, 5.147133354098043]


Centroids

[[-2.6816121341554244, 0.4387800225117981], [5.203939274722273, 0.18108
381085421293], [0.2798236662882328, 5.147133354098043]]
```

In [ ]:
```
centroids = [[-2.6816121341554244, 0.4387800225117981],
             [5.203939274722273, 0.18108381085421293],
             [0.2798236662882328, 5.147133354098043]]
```

In [22]:
```
from numpy import argmin, array, random
import math
centroids = [[-2.6816121341554244, 0.4387800225117981],
             [5.203939274722273, 0.18108381085421293],
             [0.2798236662882328, 5.147133354098043]]

def MinDist(datapoint, centroid_points):
    datapoint = array(datapoint)
    norm =  math.sqrt(sum(datapoint**2))
    centroid_points = array(centroid_points)
    diff = datapoint - centroid_points
    diffsq = diff**2

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

counts = {}
distances = {}
with open('Kmeandata.csv', 'r') as f:
    for line in f:
        D = (map(float,line.split(',')))
        idx, d =  MinDist(D, centroids)
        counts[idx] = counts.get(idx, 0) + 1
        distances[idx] = distances.get(idx, 0) + d

print counts
print distances

distance = 0.0
for k,v in dist_dict.iteritems():
    print k, v / counts[k]
    distance += v / counts[k]

print ""
print "The distance is: " + str(distance)
```

```
{0: 1024, 1: 998, 2: 978}
{0: 527.02918196006669, 1: 332.92522230882599, 2: 323.52635469657616}
0 0.514676935508
1 0.333592407123
2 0.330804043657

The distance is: 1.17907338629
```

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

In [18]:
```
%%writefile kltext.txt
1.Data Science is an interdisciplinary field about processes and systems to extract knowledge or insights from large volumes of data in various f orms (data in various forms, data in various forms, data in various form s), either structured or unstructured,[1][2] which is a continuation of some of the data analysis fields such as statistics, data mining and pre dictive analytics, as well as Knowledge Discovery in Databases.
2.Machine learning is a subfield of computer science[1] that evolved fro m the study of pattern recognition and computational learning theory in artificial intelligence.[1] Machine learning explores the study and cons truction of algorithms that can learn from and make predictions on data. [2] Such algorithms operate by building a model from example inputs in o rder to make data-driven predictions or decisions,[3]:2 rather than foll owing strictly static program instructions.
```

```
Overwriting 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 [19]:
```
import numpy as np
np.log(3)
```

Out[19]:
```
1.0986122886681098
```

In [20]:
```
%%writefile kldivergence.py
from mrjob.job import MRJob
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]*1.0/len(letter_list))]


    def reducer1(self, key, values):
        #Fill in your code
        indexlist = {}

        kl_values = {}
        for value in values:
            index = value[0]
            frequency = value[1]
            if index in kl_values:
                kl_values[index] += frequency
            else:
                kl_values[index] = frequency


        kl_value = np.where(kl_values[1] != 0, kl_values[1]* 1.0 * np.lo
g(kl_values[1]*1.0/kl_values[2]), 0)
        print key, kl_value
        yield key, kl_value

    def reducer2(self, key, values):
        kl_sum = 0.0
        for value in values:
            kl_sum = kl_sum + value

        print "Done"

        yield None, kl_sum

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

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

```
Overwriting kldivergence.py
```

In [21]:
```
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 w
ill be strict by default. It's recommended you run your job with --stri
ct-protocols or set up mrjob.conf as described at https://pythonhosted.
org/mrjob/whats-new.html#ready-for-strict-protocols
WARNING:mrjob.runner:
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
WARNING:mrjob.job:mr() is deprecated and will be removed in v0.6.0. Use
mrjob.step.MRStep directly instead.
```

```
a 0.0295721422713
b -0.00163041522831
c -0.00732786747342
d 0.0164906236566
e -0.0129926189574
f 0.00674079918689
g -0.00826965428728
h -0.00992358514474
i 0.00373655435066
k 0.000733812807303
l -0.0134916702888
m -0.00829112158145
n -0.021708593752
o -0.00910212088756
p -0.0094296551709
r -0.0071047011805
s 0.0907342592609
t -0.0102420842309
u 0.0147136183439
v 0.0198601378947
w 0.0176343237035
x -0.00165393085746
y 0.00183453201826
```

In [ ]: