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 sy
stems to extract knowledge or insights from large volumes of data i
n various forms (data in various forms, data in various forms, data
in various forms), either structured or unstructured,[1][2] which i
s a continuation of some of the data analysis fields such as statis
tics, data mining and predictive analytics, as well as Knowledge Di
scovery in Databases.
2.Machine learning is a subfield of computer science[1] that evolve
d from the study of pattern recognition and computational learning
theory in artificial intelligence.[1] Machine learning explores the
study and construction of algorithms that can learn from and make p
redictions on data.[2] Such algorithms operate by building a model
from example inputs in order to make data-driven predictions or dec
isions,[3]:2 rather than following strictly static program instruct
ions.
```

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

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

In [5]:

```
%%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):
        p_prob = 0.0
        q_prob = 0.0
        probabilities = []
        current_key = None
        for value in values:
            doc = value[0]
            cond = float(value[1])
            if current_key:
                if current_key != key:
                    probabilities.append(p_prob*np.log((p_prob+1)/
(q_prob+24)))
                    current_key = key
                    if doc == '1':
                        p_prob = cond
                    if doc == '2':
                        q_prob = cond
                else:
                    if doc == '1':
                        p_prob = cond
                    if doc == '2':
                        q_prob = cond
            else:
                current_key = key
                if doc == '1':
                    p_prob = cond
                if doc == '2':
                    q_prob = cond
        probabilities.append(p_prob*np.log((p_prob+1)/(q_prob+24)))
        yield None, probabilities

    def reducer2(self, key, values):
        kl_sum = 0.0
        for value in values:
            kl_sum = kl_sum + float(value)
        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 [6]:
```
%reload_ext autoreload
%autoreload 2
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, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
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.
```

```
---------------------------------------------------------------------
---------
TypeError                                 Traceback (most recent c
all last)
<ipython-input-6-f8229b095fc9> in <module>()
      4 mr_job = kldivergence(args=['kltext.txt'])
      5 with mr_job.make_runner() as runner:
----> 6     runner.run()
      7     # stream_output: get access of the output
      8     for line in runner.stream_output():

//anaconda/lib/python2.7/site-packages/mrjob/runner.pyc in run(sel
f)
    468             raise AssertionError("Job already ran!")
    469
--> 470         self._run()
    471         self._ran_job = True
    472

//anaconda/lib/python2.7/site-packages/mrjob/sim.pyc in _run(self)
    184
    185                 # run the reducer
--> 186                 self._invoke_step(step_num, 'reducer')
    187
    188         # move final output to output directory

//anaconda/lib/python2.7/site-packages/mrjob/sim.pyc in _invoke_st
ep(self, step_num, step_type)
    258
    259             self._run_step(step_num, step_type, input_pat
h, output_path,
--> 260                            working_dir, env)
    261
    262             self._prev_outfiles.append(output_path)

//anaconda/lib/python2.7/site-packages/mrjob/inline.pyc in _run_st
ep(self, step_num, step_type, input_path, output_path, working_di
r, env, child_stdin)
    158                 child_instance = self._mrjob_cls(args=chil
d_args)
    159                 child_instance.sandbox(stdin=child_stdin,
stdout=child_stdout)
--> 160                 child_instance.execute()
    161
    162         if has_combiner:

//anaconda/lib/python2.7/site-packages/mrjob/job.pyc in execute(se
lf)
    474
    475         elif self.options.run_reducer:
--> 476             self.run_reducer(self.options.step_num)
    477
    478         else:
```

```
//anaconda/lib/python2.7/site-packages/mrjob/job.pyc in run_reduce
r(self, step_num)
    578                                                 key=lambda
(k, v): k):
    579             values = (v for k, v in kv_pairs)
--> 580             for out_key, out_value in reducer(key, values)
or ():
    581                 write_line(out_key, out_value)
    582

/Users/JingXu/Dropbox/DataScience/W261/W261/Midterm/kldivergence.p
yc in reducer2(self, key, values)
     48         kl_sum = 0.0
     49         for value in values:
---> 50             kl_sum = kl_sum + float(value)
     51         yield None, kl_sum
     52

TypeError: float() argument must be a string or a number
```

In [ ]:

In [ ]: