General Insructions

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
```

===Instructions for midterm===

Instructions:

1: Please acknowledge receipt of exam by sending a quick reply to the instructors 2: Review the submission form first to scope it out (it will take a 5-10 minutes to input your answers and other information into this form) 3: Please keep all your work and responses in ONE (1) notebook only (and submit via the form) 4: Please make sure that the NBViewer link for your Submission notebook works 5: Please do NOT discuss this exam with anyone (including your class mates) until after 8AM (West coast time) Friday, March 4, 2016

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 notebook 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)

=====Exam questions 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 less 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

===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

===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

===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

===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

===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

===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))

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%20div MIDS-Midterm.ipynb (http://nbviewer.ipython.org/urls/dl.dropbox.com/s/9onx4c2dujtkgd7/Kullback%E2%80%93Leibler%20di 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)

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

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

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

===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).

===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)

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 $\mu 1 = \cdots = \mu k$ 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.

===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 less 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

Answers: 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

Answers: c

===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

Answers : d

===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

Answers: b , c

===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

Answers: 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

Answers: b

===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)) 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%20div
MIDS-Midterm.ipynb
(http://nbviewer.ipython.org/urls/dl.dropbox.com/s/9onx4c2dujtkgd7/Kullback%E2%80%93Leibler%20di
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 [4]:
```
%%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.
```

```
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 [5]:
```
import numpy as np
np.log(3)
```

Out[5]:
```
1.0986122886681098
```

In [20]:
```
%%writefile kldivergence.py
from mrjob.job import MRJob
import re
import numpy as np
import pandas as pd
import math

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):
        doc_list ={}
        for doc,prob in values:
            doc_list[doc]=prob
        p = doc_list[1] ## Probability for key in document 1
        q = doc_list[2] ## Probability for key in document 2
        kl = p * math.log(p/q) ## KL Divergence
        yield key, kl


    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 [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, 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.
```

```
---------------------------------------------------------------------
---------
TypeError                                 Traceback (most recent c
all last)
<ipython-input-21-36dcacf1e251> in <module>()
      2 mr_job = kldivergence(args=['kltext.txt'])
      3 with mr_job.make_runner() as runner:
----> 4     runner.run()
      5     # stream_output: get access of the output
      6     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/Lissette/Dropbox/Machine Learning at Scale/Solutions/kldive
rgence.py in reducer1(self, key, values)
     18             yield key, [index, count[key]*1.0/len(letter_l
ist)]
     19
---> 20
     21     def reducer1(self, key, values):
     22         doc_list ={}

TypeError: 'generator' object has no attribute '__getitem__'
```

===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).

Answers: a, 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)

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.

Answers: c , d

In [ ]: