

MIDS Machine Learning at Scale

End of Term exam

Week 15 Summer, 2016

Student Name: James Gray

Email: jamesgray@ischool.berkeley.edu

This code was run on a Docker image with Apache Spark, Python 2

Exam Schedule (All times are in California Time)

4:00 PM - 6:00 PM

Instructions for exam

1. Please submit your solutions and notebook via the following form:

[Submission Form \(http://goo.gl/forms/ggNYfRXz0t\)](http://goo.gl/forms/ggNYfRXz0t)

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 submission 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 Monday, Midday, of week 16
6. This is an open book exam meaning you can consult webpages and textbooks, class notes, slides etc. but you can not consult each other or any other person/group. Please complete this exam by yourself within the time limit.
7. For markdown help in iPython Notebooks please see:
https://sourceforge.net/p/ipython/discussion/markdown_syntax
https://sourceforge.net/p/ipython/discussion/markdown_syntax

Exam questions begins here

ET:1

Assume your tasked with modeling a REGRESSION problem. How do you determine which variables may be important?

1. If your data has unknown structure, start with: Tree-based methods
2. If statistical measures of importance are needed, start with: linear models (think Generalized linear models (GLMs))
3. If statistical measures of importance are not needed, start with: Regression with shrinkage (e.g., LASSO, Elastic net)
4. If statistical measures of importance are not needed, use : Stepwise regression

Select the single most correct response from the following:

- (a) 1
- (b) 2
- (c) 3
- (d) 1, 2, 3, 4

Answer: B

ET:2

Using one-hot-encoding, a categorical feature with four distinct values would be represented by how many features?

- (a) 1 feature
- (b) 2 features
- (c) 4 features
- (d) none of the above

Answer: C

ET:3

In the following (and also referring to HW12:

<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/1wb2rdqbet54y1h/MIDS-MLS-Project-Criteo-CTR.ipynb> (<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/1wb2rdqbet54y1h/MIDS-MLS-Project-Criteo-CTR.ipynb>)) we have hashed the three sample points using numBuckets=4 and numBuckets=100. Complete the three statements below about these hashed features summarized in the following table using each answer once.

Name	Raw Features	4 Buckets	100 Buckets
sampleOne	[(0, 'mouse'), (1, 'black')]	{2: 1.0, 3: 1.0}	{14: 1.0, 31: 1.0}
sampleTwo	[(0, 'cat'), (1, 'tabby'), (2, 'mouse')]	{0: 2.0, 2: 1.0}	{40: 1.0, 16: 1.0, 62: 1.0}
sampleThree	[(0, 'bear'), (1, 'black'), (2, 'salmon')]	{0: 1.0, 1: 1.0, 2: 1.0}	{72: 1.0, 5: 1.0, 14: 1.0}

With 100 buckets, sampleOne and sampleThree both contain index 14 due to ____.

- (a) A hash collision
- (b) Underlying properties of the data
- (c) The fact that used 100 buckets
- (d) none of the above

Answer: B

ET:4

In the following (and also referring to HW12:

<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/1wb2rdqbet54y1h/MIDS-MLS-Project-Criteo-CTR.ipynb> (<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/1wb2rdqbet54y1h/MIDS-MLS-Project-Criteo-CTR.ipynb>)) we have hashed the three sample points using numBuckets=4 and numBuckets=100. Complete the three statements below about these hashed features summarized in the following table using each answer once.

Name	Raw Features	4 Buckets	100 Buckets
sampleOne	[(0, 'mouse'), (1, 'black')]	{2: 1.0, 3: 1.0}	{14: 1.0, 31: 1.0}
sampleTwo	[(0, 'cat'), (1, 'tabby'), (2, 'mouse')]	{0: 2.0, 2: 1.0}	{40: 1.0, 16: 1.0, 62: 1.0}
sampleThree	[(0, 'bear'), (1, 'black'), (2, 'salmon')]	{0: 1.0, 1: 1.0, 2: 1.0}	{72: 1.0, 5: 1.0, 14: 1.0}

It is likely that sampleTwo has two indices with 4 buckets, but three indices with 100 buckets due to ____.

- (a) A hash collision
- (b) Underlying properties of the data
- (c) The fact that we go from 4 to 100 buckets
- (d) none of the above

Answer: C

ET:5

In the following (and also referring to HW12:

<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/1wb2rdqbet54y1h/MIDS-MLS-Project-Criteo-CTR.ipynb> (<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/1wb2rdqbet54y1h/MIDS-MLS-Project-Criteo-CTR.ipynb>)) we have hashed the three sample points using numBuckets=4 and numBuckets=100. Complete the three statements below about these hashed features summarized in the following table using each answer once.

Name	Raw Features	4 Buckets	100 Buckets
sampleOne	[(0, 'mouse'), (1, 'black')]	{2: 1.0, 3: 1.0}	{14: 1.0, 31: 1.0}
sampleTwo	[(0, 'cat'), (1, 'tabby'), (2, 'mouse')]	{0: 2.0, 2: 1.0}	{40: 1.0, 16: 1.0, 62: 1.0}

sampleThree	[(0, 'bear'), (1, 'black'), (2, 'salmon')]	{0: 1.0, 1: 1.0, 2: 1.0}	{72: 1.0, 5: 1.0, 14: 1.0}
-------------	--	--------------------------	----------------------------

With 4 buckets, sampleTwo and sampleThree both contain index 0 due to ____.

- (a) A hash collision
- (b) Underlying properties of the data
- (c) The fact that we use 4 buckets

Answer: A (I suspect this is due to a hash collision given that some of features map to the same index)

ET:6 When applying numerical machine learning approaches (and for non-numerical approaches if required) to big data problems which of the following steps are could be used during modeling and are recommended:

- (a) Convert categorical features to numerical features via one-hot-encoding and store in a dense representation
- (b) Transform categorical features using hashing regardless of how many unique categorical values exist in training and test data
- (c) Use matrix factorization to remap your input vectors to latent concepts
- (d) none of the above

Answer: A, B, C

ET:7

When dealing with numerical data which of the following are ways to deal with missing data:

- (a) Delete records that have missing input values
- (b) Standardize the data and set all missing values to 1 (one)
- (c) Use K-nearest neighbours based on the test set to fill in missing values in the training set
- (d) none of the above

Answer: A (with a large data the impact should be minor)

ET:8

In the Criteo project, we're trying to predict what:

- (a) Revenue from click events
- (b) Click-through vs not click event
- (c) Probability of a purchase
- (d) none of the above

Answer: B

ET:9

Which of the following are true about the purpose of a loss function?

- (a) It's a way to penalize a model for incorrect predictions
- (b) It precisely defines the optimization problem to be solved for a particular learning model
- (c) Loss functions can be used for modeling both classification and regression problems
- (d) none of the above

Answer: A, B, C

ET:10

When implementing Logistic Regression with Regularization in Spark which of the following apply?

- (a) When lambda equals one, it provides the same result as standard logistic regression
- (b) One only needs to modify the standard logistic regression by modifying the Mapper
- (c) Can be framed as minimizing a convex function
- (d) none of the above

Answer: B, C

ET:11

In the context of ecommerce you have just deployed a new conversion rate prediction model to production. This model (aka treatment model) will challenge the control model (i.e., the current model) in AB Test manner to see if it can produce better revenue. Here is the data that was taken from this live AB Test.

CONTROL MODEL (our new CTR model)

Impression ID	Revenue
1	\$0.50
2	\$0.50
3	\$3.00
.....	
20000	\$3.00
20001	\$3.00
20002	\$3.00
20003	\$3.00
.....	
50,001	\$3.00
.....	
100,000	\$4.00

All other impressions in this 100,000 sample resulted in zero transactions and therefore zero revenue.

TREATMENT MODEL (our new CTR model)

Impression ID	Revenue
1	\$1.50
2	\$0.50
3	\$0.00
.....	
50,001	\$3.00
.....	
100,000	\$4.00

All other impressions in this 100,000 sample resulted in zero transactions and therefore zero revenue.

P-values are a common way to determine the statistical significance of a test. The smaller it is, the more confident you can be that the test results are due to something other than random chance. A common p-value of .05 is a 5% significance level. Similarly, a p-value of .01 is a 1% significance level. A p-value of .20 is a 20% significance level. For this problem set the p-value to 0.01

Which of the following are true:

- (a) Based on revenue there is no statistical significant difference between the Control and the Treatment at p-value of 0.05 for a one-sided t-test
- (b) Based on transaction rates (transactions that generated revenue versus not) there is no statistical significant difference between the Control and the Treatment at p-value of 0.05 for a one-sided t-test
- (c) AB testing using differences in revenue for this problem is a useful means of determining if the Treatment conversion rate prediction model is better than the control model.
- (d) none of the above

```
In [1]: import numpy as np
        from scipy.sparse import coo_matrix
        import scipy.stats

        control=[.5, .5, 3, 3, 3, 3, 3, 3, 4]
        control.extend([0]*(100000-len(control)))
        control=np.asarray(control)

        treatment=[ 0, 0, 0, 0, 0, 0, 0, 0, 0]
        treatment.extend([0]*(100000-len(treatment)))
        treatment=np.asarray(treatment)

        #nb 2-sided test
        scipy.stats.ttest_ind(control, treatment), scipy.stats.ttest_ind(control,ast
```

```
Out[1]: (Ttest_indResult(statistic=2.7393492169656311, pvalue=0.00615663988141617
29),
        Ttest_indResult(statistic=3.000120008400637, pvalue=0.002699064857051299
4))
```

Answer: C

ET:12

Given this graph expressed in the form of an adjacency list,

```

Node    adjacentNode:weightAssociatedWithEdge
N1      N6:10,   N2:2
N2      N3:1
N3      N4:1
N4      N5:1
N5      N6:1
N6      N7:1
N7      N8:1
N8      N9:1

```

Using the parallel breadth-first search algorithm for determining the shortest path from a single source, how many iterations are required to discover the shortest distances to all nodes from Node 1

- A 7
- B 8
- C 13
- D None of the above

Answer: B

ET:13 Assume the Lagrangian for SOFT SVMs (unconstrained optimization) is as follows:

minimize $[\lambda/2 * w'w + C \sum_{i=1:n} \xi_i + 1/n \sum_{i=1:n} \max(0, 1 - \xi_i - y_i(w'x_i - b))]$

- (a) When λ is super small (e.g., 0.000001), then the above Lagrangian will yield a Hard SVM
- (b) In the context of support vector machines, linear kernels can be readily parallelized in map reduce frameworks such as Spark
- (c) Sequential learning via algorithms such as perceptron can take advantage of map-reduce frameworks and yield exactly the same results as a single core implementation with significant reductions in training time
- (d) When λ is 1.0, then the above Lagrangian will yield a Soft SVM

Answer: A, D

ET:14 Given the following paired RDDs

RDD1 = {(1, 2), (3, 4), (3, 6)} RDD2 = {(3, 9), (3, 6)}

Using PySpark, write code to perform an inner join of these paired RDDs. What is the resulting RDD? Make your Spark available in your notebook:

- A: [(3, (4, 9)), (3, (6, 9))]
- B: [(3, (4, 9)), (3, (4, 6)), (3, (6, 9)), (3, (6, 6))]
- C: [(3, (4, 9)), (3, (4, 6)), (3, (6, 9)), (3, (6, 9))]
- D: None of the above

```
In [2]: import os
import sys

import pyspark
app_name = "finalexam"
master = "local[*]"
conf = pyspark.SparkConf().setAppName(app_name).setMaster(master)
sc = pyspark.SparkContext(conf=conf)
```

```
In [3]: # perform inner join of paired RDD's
rdd1 = sc.parallelize([(1, 2), (3, 4), (3, 6)])
rdd2 = sc.parallelize([(3, 9), (3, 6)])
# join and return results
rdd1.join(rdd2).collect()
```

```
Out[3]: [(3, (4, 9)), (3, (4, 6)), (3, (6, 9)), (3, (6, 6))]
```

Answer: B

ET:15 You have been tasked to build a predictive model to forecast beer sales for a chain of stores.

After doing basic exploratory analysis on the data, what is the first thing you do regarding modeling?

- (a) Construct a baseline model
- (b) Determine a metric to evaluate your machine learnt models
- (c) Split your data into training, validation and test subsets (or split using cross fold validation)
- (d) All of the of the above

Answer: D (I believe there are all reasonable steps to perform. I would also probably profile the data to evaluate data quality and determine if missing or inconsistent data should be removed).

ET:16

Use Spark and the following notebook to answer this question:

- <http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/6s5ph41h74bqqwi/Linear-Regression-on-Beer-Data.ipynb>
(<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/6s5ph41h74bqqwi/Linear-Regression-on-Beer-Data.ipynb>)
- <https://www.dropbox.com/s/6s5ph41h74bqqwi/Linear-Regression-on-Beer-Data.ipynb?dl=0> (<https://www.dropbox.com/s/6s5ph41h74bqqwi/Linear-Regression-on-Beer-Data.ipynb?dl=0>)

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a model for say a forecasting method in statistics, for example in trend estimation. It usually expresses accuracy as a percentage, and is defined by the formula:

MAPE = average over all examples $(100 * \text{Abs}(\text{Actual} - \text{Predicted}) / \text{Actual})$

Note when Actual is zero that test row is dropped from the evaluation.

Construct a mean model for target variable CASES18PK. Calculate the MAPE for the mean model over the training set. Select the closest answer.

- (a) 200%
- (b) 250%
- (c) 20%
- (d) 180%

```
In [5]: from pyspark.mllib.regression import LabeledPoint, LinearRegressionWithSGD
import os
# beerSales is produced from the notebook provided in the link above
df=sc.textFile("beerSales.txt")
print df.take(1)[0].split('\t')
# make iterator
df=df.zipWithIndex().filter(
    lambda x: x[1]>0
).map(
    lambda x: [float(y) for y in x[0].split('\t')]
)

[u'Week', u'PRICE12PK', u'PRICE18PK', u'PRICE30PK', u'CASES12PK', u'CASES
18PK', u'CASES30PK']
```

```
In [15]: def runRegression(data):
# train model
regression_model=LinearRegressionWithSGD.train(data, iterations=1000, st
# predict value
prediction=data.map(lambda pt: (pt.label, regression_model.predict(pt.fe
MAPE = prediction.map(lambda (v, p): 100*abs(v - p)/v).reduce(lambda x,
print("Mean Absolute Prediction Error(MAPE) = " + str(MAPE))
```

```
In [10]: # run the regression model for target variable CASES18PK
```

```
In [11]: runRegression(df.map(lambda x: LabeledPoint(x[5],x[1:4])))
```

Mean Absolute Prediction Error(MAPE) = 171.737094196

Answer: D (is the closest answer)

ET:17

Use Spark and the following notebook to answer this question:

- <http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/6s5ph41h74bqgwi/Linear-Regression-on-Beer-Data.ipynb>
(<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/6s5ph41h74bqgwi/Linear-Regression-on-Beer-Data.ipynb>)

- <https://www.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb?dl=0> (<https://www.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb?dl=0>)

The target variable CASES18PK is skewed, so take the log of it (and make it more normally distributed) and compute the MAPE of the mean model for CASES18PK. Select the closest answer to your calculated MAPE.

- (a) 200%
- (b) 30%
- (c) 20%
- (d) 10%

```
In [12]: # take the log of the CASES18K data set given that its skewed
import math
# run regression using log data
runRegression(df.map(lambda x: LabeledPoint(math.log(x[5]),x[1:4])))

Mean Absolute Prediction Error(MAPE) = 37.4346369254
```

Answer: B (is the closest answer)

ET:18

Use Spark and the following notebook to answer this question:

- <http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb> (<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb>)
- <https://www.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb?dl=0> (<https://www.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb?dl=0>)

Build a linear regression model using the following variables:

$\text{Log}(\text{CASES18PK}) \sim \text{log}(\text{PRICE12PK}), \text{log}(\text{PRICE18PK}), \text{log}(\text{PRICE30PK})$

Calculate MAPE over the test set and select the closest answer.

- (a) 4.3%
- (b) 4.6%
- (c) 3.5%
- (d) 3.9%

Answer: A (note that this was a guess)

ET:19

Recall that Spark automatically sends all variables referenced in your closures to the worker nodes. While this is convenient, it can also be inefficient because (1) the default task launching mechanism is optimized for small task sizes, and (2) you might, in fact, use the same variable in multiple parallel operations, but Spark will send it separately for each operation. As an example, say that we wanted to write a Spark program that looks up countries by their call signs (e.g., the call sign for Ireland is EJZ) by prefix matching in a table. In the following the "signPrefixes" variable is essentially a table with two columns "Sign" and "Country Name". The goal is to join the following tables:

signPrefixes table with columns "Sign" and "Country Name"
contactCounts table with columns "Sign" and "count"

to yield a new table:

countryContactCounts with the following columns "Country Name" and "count"

Use Spark and the following notebook to answer this question:

- <http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb>
(<http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb>)
- <https://www.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb?dl=0> (<https://www.dropbox.com/s/6s5ph41h74bggwi/Linear-Regression-on-Beer-Data.ipynb?dl=0>)

```

In [ ]: # this is the code from the notebook referenced above

#..... Other code...
#Country lookup code

# Helper functions for looking up the call signs

def lookupCountry(sign, prefixes):
    pos = bisect.bisect_left(prefixes, sign)
    return prefixes[pos].split(",")[1]

def loadCallSignTable():
    f = open("callsign_tbl_sorted.txt", "r")
    return f.readlines()

# Lookup the locations of the call signs on the
# RDD contactCounts. We load a list of call sign
# prefixes to country code to support this lookup.
signPrefixes = loadCallSignTable()

def processSignCount(sign_count, signPrefixes):
    country = lookupCountry(sign_count[0], signPrefixes)
    count = sign_count[1]
    return (country, count)

contactCounts = sc.parallelize([["ZMZ", 1], ["ZMZ", 3]])

countryContactCounts = (contactCounts
                        .map(lambda signCount: processSignCount(signCount, signPrefixes))
                        .reduceByKey(lambda x, y: x + y))

#countryContactCounts.saveAsTextFile("tmp/countries.txt")

```

How can we modify this code to make it more efficient? Choose one response only

- (a) modify line 18 with `sc.broadcast(loadCallSignTable())`
- (b) Use accumulators to store the counts for each country
- (c) The code is already optimal
- (d) none of the above

Answer: A (broadcasting will enable variables to be distributed across the cluster)

END OF EXAM