CSCI585 Final exam

2017-12-07; duration: 1 hour

Hi everyone. There are 11 questions below (10 plus a bonus), each question starting in a new page. **Please read each question carefully before answering.** There's need to elaborate on anything, so you shouldn't need extra sheets (that said, there are three blank sheets at the end).

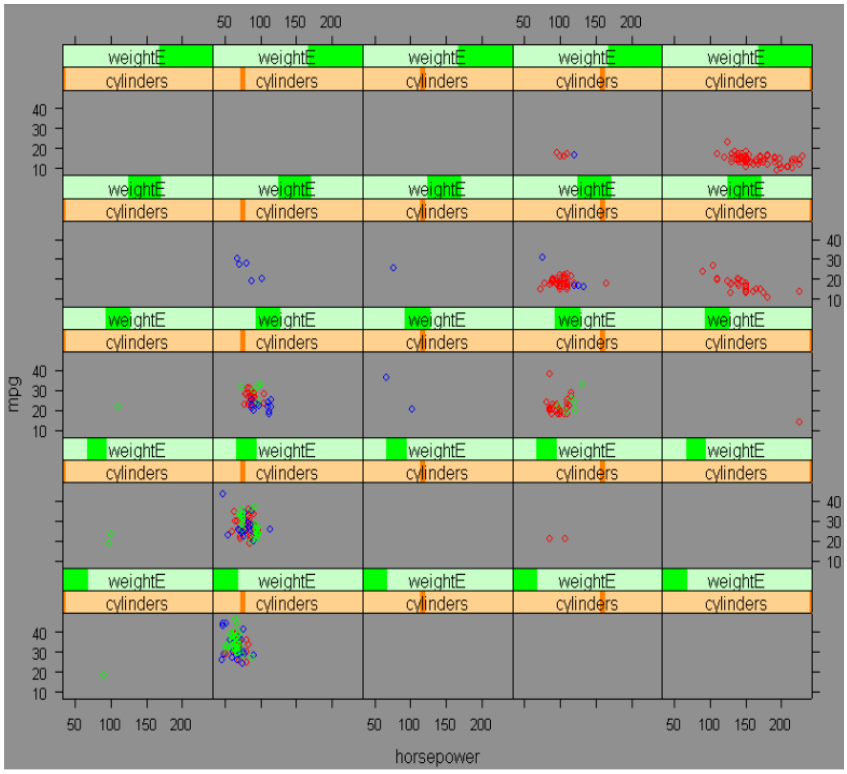
The exam is **CLOSED** book/notes/devices/neighbors(!) but 'open mind' :) If you are caught cheating in any manner, you will get a 0 on the test and also be reported to SJACS - so please don't cheat! **DO YOUR OWN WORK**.

When we announce that the time is up, you NEED to stop writing immediately, and turn in what you have; if you continue working on the exam, we will not grade it (ie. you will get a 0). So **please stick to the limit of one hour, use time wisely**!

Question	Points possible	Your score
Q1	1	
Q2	2	
Q3	4	
Q4	3	
Q5	3	
Q6	4	
Q7	4	
Q8	5	
Q9	2	
Q10	2	
BONUS	**1**	
Total	**31**	

Q1 (1 point).

Shown below is a 'trellis view' (grid view) of cars-related data. **What is a more technical term** to describe such visualization?



Trellis Display of an Auto Dataset

● American ● European ● Japanese

A. Multivariate data visualization [the word 'multivariate' does need to occur in the answer].

Q2 (1+1=2 points).

When we have numerical data (eg. home price-related), we can make predictions on a continuous scale, using linear regression (including multiple linear regression). For example, we can fit a multi linear equation for the following variables (that belong to a historical (and racially biased) dataset of home prices in Boston).

```
:Attribute Information (in order):
    - CRIM     per capita crime rate by town
    - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
    - INDUS    proportion of non-retail business acres per town
    - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
    - NOX      nitric oxides concentration (parts per 10 million)
    - RM       average number of rooms per dwelling
    - AGE      proportion of owner-occupied units built prior to 1940
    - DIS      weighted distances to five Boston employment centres
    - RAD      index of accessibility to radial highways
    - TAX      full-value property-tax rate per $10,000
    - PTRATIO  pupil-teacher ratio by town
    - B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
    - LSTAT    % lower status of the population
    - MEDV     Median value of owner-occupied homes in $1000's
```

What are two other regression-related alternatives for predicting numerical targets? Note - the alternatives can't be nearly identical to each other. Explain each, using a few sentences.

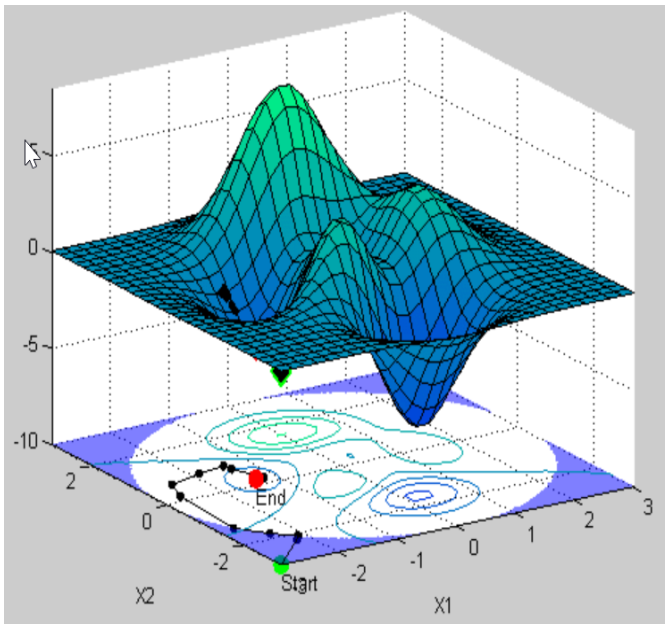
A. Non-linear (eg. polynomial, parameter-free) regression, regression tree.

Q3 (1+1+2=4 points).

a. After an 'AI winter' that lasted nearly 25 years (1985-2010), we are seeing a resurgence/explosion in machine learning, implemented using deep neural networks. **What is the technical reason why** the success rate, and flexibility (in the type of data can be learned) is astonishingly high?

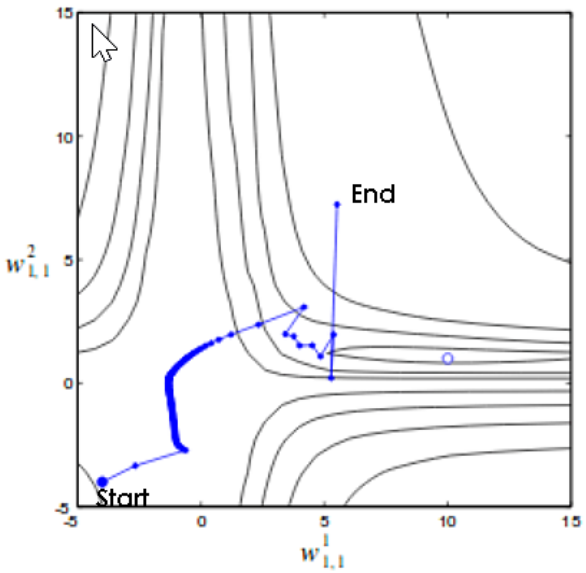
A. Layers of neurons with nonlinear (eg. sigmoid) activation functions makes it possible to approximate any smooth, continuous signal with low-enough error [they are universal function approximators]. FYI, ref: http://www.dartmouth.edu/~gvc/Cybenko_MCSS.pdf

b. For a neuron with two inputs, the error surface is shown below. Also shown below the error surface (in the 'contour plot') are the start and end values of the weights (we begin backpropagation with 'Start' weights, and stop when we get to the 'End' weights). **What do you notice**, about the training?



A. The minimum error that was reached is a local mininum, ie not the best one - there is a better (global) minimum to the right!

c. Shown below for a different neuron is its error surface's contour plot, and a training sequence of weights from 'Start' to 'End'. **What do you notice, and what is the reason you would attribute to it?**



A. The 'End' stopping point has overshot past the ideal end. This typically happens when the learning rate is set to be too high.

Q4 (3 points).

Google's TensorFlow API offers a powerful, dataflow-based approach to implementing DM/ML algorithms that process vast amounts of data (eg. realtime processing of data generated by a self-driving car). A 'tensor' is simply a multi-dimensional array datatype. Most tensorflow functions output tensors (which can be passed to other functions as inputs), some output a scalar (ie. single) value. Consider the following TensorFlow snippet: the tf.constant() calls create 1D arrays X and Y; tf.reduce_mean() finds the average (mean) of its input, and tf.reduce_sum() outputs the sum of elements. **What does the snippet calculate?** Explain, in a few sentences.

```
X = tf.constant(data[:,0], name="X")
Y = tf.constant(data[:,1], name="Y")

Xavg = tf.reduce_mean(X, name="Xavg")
Yavg = tf.reduce_mean(Y, name="Yavg")
num = (X - Xavg) * (Y - Yavg)
denom = (X - Xavg) ** 2
rednum = tf.reduce_sum(num, name="numerator")
reddenom = tf.reduce_sum(denom, name="denominator")
m = rednum / reddenom
b = Yavg - m * Xavg
```

A. The snippet does linear regression between values in arrays X and Y, fitting a line through the data. It solves for m and b, in the line equation y=mx+b, using least-squares estimates [from elementary statistics].

More FYI:

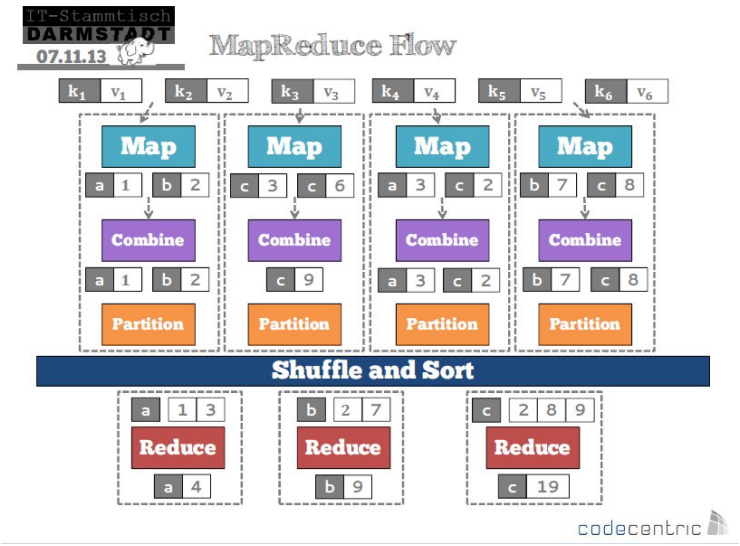
$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$$

Q5 (2+1=3 points).

In Map(Shuffle)Reduce, there is sometimes an optional 'Combine' step. **Illustrate and explain** this with a small example (diagram). If included, **what is its purpose** (what does it achieve)?

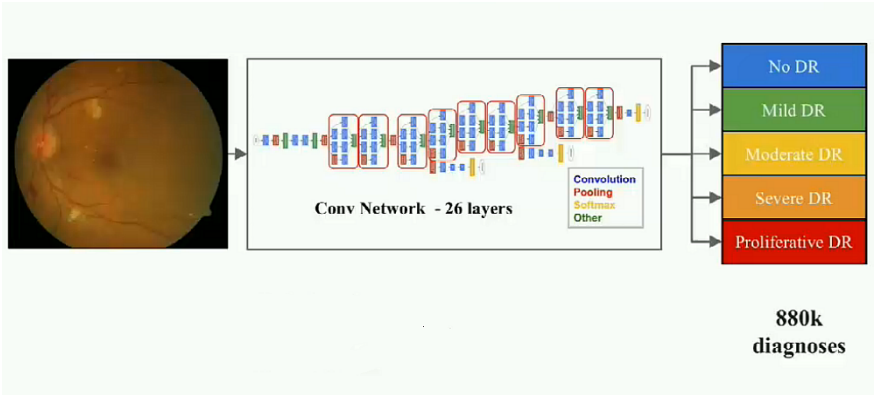
A. A combiner performs local reduction, where it collates/aggregates values of multiple identical keys output by a mapper. From the class notes:



We would do this in order to minimize the work the shuffler has to do (ie to minimize data transfer volume), and to decrease the load on the reducers.

Q6 (3+1=4 points).

Diabetic retinopathy is an eye disease (that eventually leads to blindness, if left untreated!), caused by high blood sugar levels damaging the retina's blood vessels. This can be detected via a retinal scan (which looks like the image in the left of the figure below) - an ophthalmologist makes the diagnosis from the scan. Machine learning, using a large body of existing patient data, can be used to automate such diagnosis, as summarized in the picture below (where 880,000 such diagnoses were used!). In the diagram, 'DR' stands for 'diabetic retinopathy'. **Explain, precisely, how this (automated diagnosis) would be achieved** (what is being summarized in the diagram). Also, it is likely that at first, the error (misdiagnosis) rates might be higher than those of human doctors. **Why isn't this a problem, in such cases?**



A. A deep CNN (with 26 layers, pictured above) is used to train the network, using existing patient scans and hand-labeled (derived from doctors' diagnoses) results. Of the 880K diagnoses available as 'past data', a large fraction would be used for training, after which the network is ready to classify unseen data; the rest of the prior data is treated as 'test data', to evaluate the accuracy of prediction. The network would need to be tuned (number of layers, neurons in each layer, learning rate, momentum) in order to end up with a small-enough error rate. After this, the network can be deployed, and diagnose new patients' scans into one of 5 different classes (No DR through Proliferative DR).

We aren't concerned with initial error levels because each new misclassified input (as verified by a doctor) would be used as training data (along with the prior 880K inputs) to improve classification accuracy - over time, the network will only get asymptotically better at this, never worse.

Q7 (4 points).

Many real-world data processing tasks that get parallel-processed on Hadoop/YARN require more than a single map() and reduce() step. Such 'cascades' (dataflow) of map-shuffle-reduce (M-S-R) chains can be managed efficiently in YARN, using Oozie or Mahout.

James Joyce's massive masterpiece book,'Ulysses', has 265,222 words (!) **How would you devise a cascaded M-S-R approach**, to output the distinct occurrences of words in Ulysses, sorted by their decreasing frequencies? Illustrate, using diagrams. Eg. the top 5 words might be output as:
31354 a
20045 of
18342 the
12038 and
9432 in

A.
Stage 1: mapper: output (<word>,1) k-v pairs; reducer: output (<word>,count) pairs.
Stage 2: mapper: output (count,<word>) pairs [swap key and value]; reducer: do nothing, just output (count,<word>) pairs
FYI: the shuffler (always) sorts keys, which is why the second-stage reducers above are sent sorted keys, which in our case is wordcount - so the overall output is a list of most-occurring to least-occurring words.

Q8 (10*0.5=5 points).

By definition, geospatial DBs help visualize data that have spatial extent. Given a map of the US (such as the one below), **give an example** of the type of data for each category listed below (the first one is filled out for you, as a sample answer).

a. Everyday human activity: automobile traffic during rush hour
b. Agricultural: **bushels of corn grown last year**
c. Climate/weather-related: **amount of annual rainfall for last year**
d. Consumer-related: **number of Walmart stores**
e. Environmental: **locations of 'superfund' (nuclear waste processors)**
f. Education-related: **number of universities**
g. Energy-related: **locations of coal-fired generators**
h: Financial: **median home price**
i. Health/medical: **% of obese people**
j. Public safety-related: **number of homicides last year**
k: Science/engineering/tech-related: **locations of national laboratories**

For a,b,c,d,f,h,i and j, the examples are for per-state values.

Q9 (0.5*4=2 points).

The world today is awash in massive quantities of 'Big Data', generated these days from Internet content (web pages, tweets, blogs, videos, pics...), user behavior online, sensors/instruments, etc. Prior to all this, "data" resulted from relatively limited sources and processes/collection practices. **Name 4 distinct sources/practices** that resulted in such "old school" data.

A.
Census collection
Weather data
Data warehousing
Banks: checking/savings, loans
Stock market
...

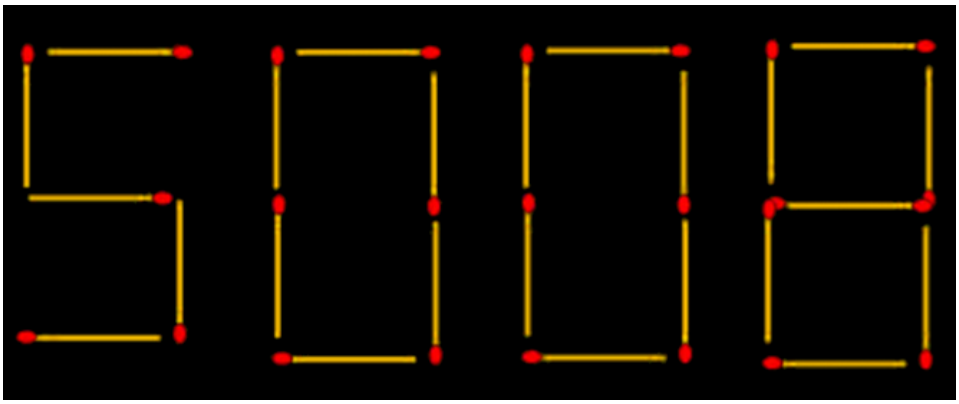
Q10 (1+1=2 points).

Recall 'polyglot persistence', when it comes to NoSQL DBs. **What two [somewhat inter-related, but distinct] reasons can you think of**, for this to be a 'bad' thing (eg. why a company wouldn't choose to encourage this in their data infrastructure)?

A. By possibly needing to 'port' code from one language to another (eg. when a decision is made to switch a part of the application from Python to R), bugs and inefficiencies could creep in; also, the lack of a standard modeling+query language such as SQL, creates lowered productivity overall (solutions can't be reused across other in-house applications, algorithms need to be coded up from scratch...).

Bonus (1 point).

Shown below is 5008, "written" using matches. By moving **exactly two matches** (no more, no fewer), what is the **largest** number you can express? Write/sketch your answer below the puzzle. **There is only one correct answer** - no point (fractional or full) for any other answer :) Read the question carefully... Be creative!



A. 11^81105 (!!!).