CSCI585 Final exam

2017-05-04

Duration: 1 hour

Last Name: ______________________________

First Name: ______________________________

Student ID: ______________________________

Email: ______________________________

Hi there! There are 9 questions below (8 plus a bonus), one question per page. Please read each question carefully before answering. There's need to elaborate on anything, so you shouldn't need extra sheets.

The exam is **CLOSED** book/notes/devices/neighbors(!) but 'open mind' :) If you are observed cheating, or later discovered to have cheated in any manner, you will get a 0 on the test and also be reported to SJACS - so please don't! **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**!

Have fun, and good luck - hope you do well!

Saty

Q1 (1+1=2 points).

a. In what sense is Data Mining, an expanded version of ‘statistics’?

A.
Statistics is about summarization of data: we collect and analyze numerical data in large quantities, for the purpose of inferring proportions in a whole, from those in a smaller representative sample.

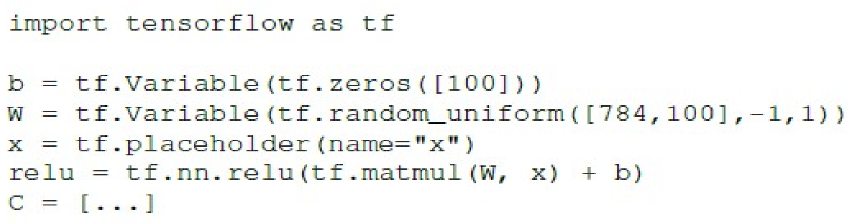
With Data Mining, we don’t summarize or make inferences about a larger population - we analyze all available data, and look for patterns in it.

b. How is Machine Learning related to Data Mining?

A.
Data Mining stops with the discovery of patterns in data. In Machine Learning, we ‘publish’ the model that we mine, and continue processing new incoming data, using our generated model.

Q2 (4 points).

The following shows a small TensorFlow fragment:

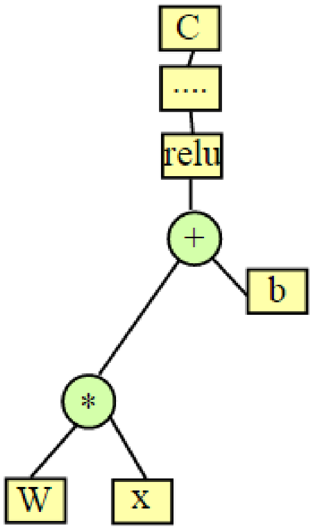


```
import tensorflow as tf

b = tf.Variable(tf.zeros([100]))
W = tf.Variable(tf.random_uniform([784,100],-1,1))
x = tf.placeholder(name="x")
relu = tf.nn.relu(tf.matmul(W, x) + b)
C = [...]
```

Draw a graph (where the output would be C) that shows the computation above.

A.



Q3 (2+2=4 points).

‘R’ is at its core, a statistics programming language, which is why it has enjoyed a recent resurgence, in data mining and machine learning. What are the two most important datatypes in ‘R’ that are specifically meant for data processing?

A.

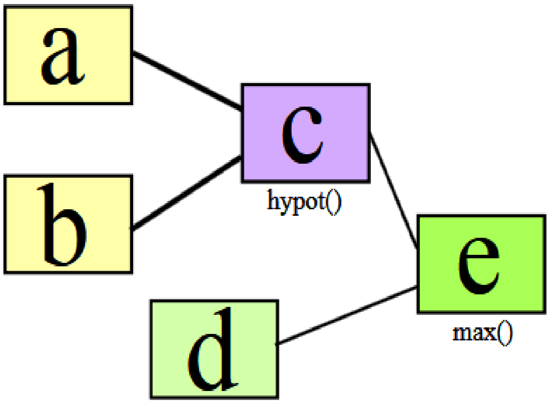
Vector - used to create an array of values that can be processed as a single entity (together, without an explicit loop).

Data Frame - used to create tabular (table-like) objects, with column names that are assigned column data as vectors. In other words, a data frame is composed of named columns, where each column is a vector type.

Q4 (4 points).

As you know, Apache Pig is a framework for expressing MapReduce calculations in a simple (compared to writing mappers and reducers in Java, Python etc) way; TensorFlow is a Python framework for expressing computation (Google uses it for DNNs). What is common to these two systems? In other words, what manner of computation do they help us carry out? Explain in 3 or 4 sentences, using diagrams.

A. Both Pig and TF help carry out dataflow computation, where data processing nodes are connected in the form of an acyclic graph, with data flowing through the nodes. The systems (Pig, TF) track the dependencies between the nodes, and schedule parallel node execution whereever possible. Here is a sample dataflow graph:



Q5 (2+1+1=4 points).

a. What is Apache Spark?
A. Spark makes Big Data real-time and interactive - it is an in-memory data processing engine (so it is FAST), specifically meant for iterative processing of data.

b. Name two Spark addons (libraries), and mention what they are used for.
A.
Spark Streaming [for handling streaming data]
Spark SQL [for executing SQL queries]
Spark MLib [for machine learning applications]
Spark GraphX [for creating graph DBs]

Q6 (1+3=4 points).

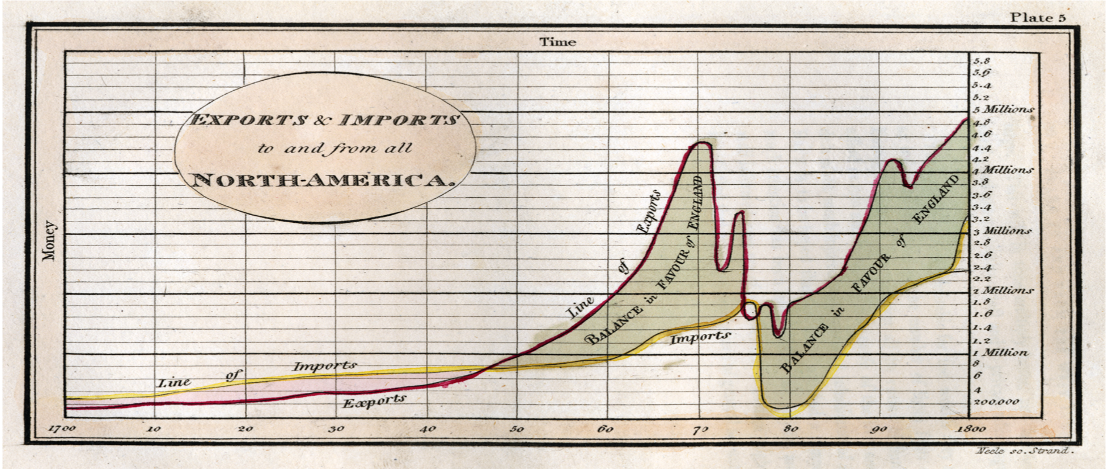
Usually in Big Data processing, we would employ horizontal fragmentation (split up a relation into groups of rows) to speed up processing. But we also use a strategy where we do vertical fragmentation (where we split columns).

a. What is this type of NoSQL database called?
A. Column family database.

b. How does this strategy help process queries faster (what is the advantage of doing this)?
A. Column families (groups of columns) that are accessed more frequently can be kept in a separate file, and non-essential columns in another file. The essential columns file can then be stored in high speed memory (eg. SSD) and accessed faster.

Q7 (2+2=4 points).

Following is an old chart that shows, from England's point of view, exports and imports to/from the US, between 1700 and 1800:



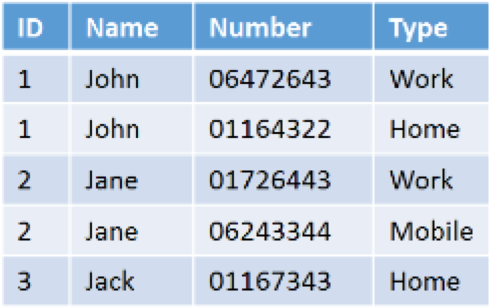
If the UK (England, Scotland, Ireland, Wales) wants to create a similar plot of export/import with the US, for 2020 to 2030, it would presumably want to collect more fine-grained data (categorized) for those 11 years.

a. What data categories (dimensions) can you think of?
A. Countries (England, Scotland..), cities within countries, months between 2020 and 2030, product categories (eg. food, clothing, toys..)..

b. Once all the data is collected, plotting all the categories in a single graph like the above would make it cluttered. So what would you do, to help viewers understand the data best?
A. Create an interactive plot, where the user can turn on/off the various categories such as countries and product categories; have a time slider that helps display data at a chosen time (eg at a specific month in a given year).

Q8 (4 points).

Here is a small table of phone numbers:



ID	Name	Number	Type
1	John	06472643	Work
1	John	01164322	Home
2	Jane	01726443	Work
2	Jane	06243344	Mobile
3	Jack	01167343	Home

Represent the above data as (valid!) JSON, using ‘id’, ‘name’ and ‘phoneNumber’ as keys. You can use ‘data’ as the key for the entire data above.

A.

```
{
  data: [
    {
      ‘id’: 1,
      ‘name’: ‘John’,
      ‘phoneNumber’:[
          {‘Work’: 06472643},
          {‘Home’: 01164322}
       ]
    },
    ....objects for Jane and Jack
  ]
}
```

Bonus question (1 point).

How would you draw the following figure using a single, unbroken line: you can't lift the pen while drawing, and you can't draw over even a part of an existing line.