Practice Questions for Midterm - 10-605

Oct 14, 2015 (version 1)

10-605
Fall 2015
Sample Questions
Time Limit: n/a

Name: ____________________

Andrew ID: ____________________

Grade Table (for teacher use only)

Question	Points	Score
1	6	
2	6	
3	15	
4	6	
5	4	
6	5	
7	4	
8	4	
9	8	
Total:	58	

Review questions from previous years

From `http://www.cs.cmu.edu/~wcohen/10-605/practice-questions/s2014-final.pdf` questions 1.1-1.2, 1.5-1.7; 3.1-3.2, 3.4; 4; 5.

From `http://www.cs.cmu.edu/~wcohen/10-605/practice-questions/s2015-final.pdf` questions 1, 4, 5, 10, 14, 15, 16.

Parallel learning methods

1. (6 points) Recall that iterative parameter mixing (IPM) algorithm for perceptrons works as follows: First, divide the data into s shards, and initialize a weight vector $\mathbf{w}^0$ to zero. Then, in each iteration t, run, in parallel, a perceptron for one pass over a single shard, starting with weight vector $\mathbf{w}^{t-1}$; and average the final weight vectors for each shard to create the next weight vector $\mathbf{w}^t$. Assume that the average is unweighted, i.e., uniform mixing.

   Mark the statements as true or false.

   - ◯ The mistake bound for IPM for perceptrons shows that the number of iterations needed to converge does *not* depend on the number of shards.
   - ◯ If the original perceptron algorithm makes at most m mistakes while training on the data, and there are s shards, then IPM for perceptrons will proveably make at most m mistakes during training.
   - ◯ If the original perceptron algorithm makes at most m mistakes while training on the data, and there are s shards, then IPM for perceptrons will proveably make at most $s * m$ mistakes during training.

2. (6 points) Recall that the AllReduce operation combines a reduce operation with a broadcast operation.

   (a) AllReduce is useful in iterative parameter mixing (IPM). In one sentence, what part of IPM would it be useful for?

(b) AllReduce typically communicates information along a k-ary spanning tree of worker nodes. In one or two sentences, what are the advantages of this, rather than communicating from each worker to a single central node?

3. (15 points) In the following scenarios, how will you perform the perceptron updates on given training data?

   (a) Number of training instances = 10,000 and dimension of feature vector = 20.

   (b) Number of training instances = 10,000,000 and dimension of feature vector = 20.

   (c) Number of training instances = 1,000 and dimension of feature vector = 10,000.

Hashing and Stochastic gradient

4. (6 points) Mark the statements as true or false.

   ○ Using stochastic gradient descent on logistic regression with a hashing trick is a way to learn classifers that are not linearly separable, because feature hashing is a type of kernel.

   ○ The hashing trick reduces the memory required to store a classifier.

   ○ The hashing trick makes it faster to apply a classifier to an instance.

5. (4 points) Mark the statements as true or false.

   ○ DSGD for matrix factorization is an approximate version of matrix factorization using SGD.

   ○ We cannot use the DSGD algorithm for matrix factorization if the entries in the matrix are negative (e.g. movie ratings from -5 to 5).

6. (5 points) You joined a company which works on finding similar images. You started out with working on a cosine similarity based approach between the image pixel vectors. During this experiment, you found that it takes a lot of time to do this. Can you optimize on the time taken?

Map-reduce

7. (4 points) In the default setting of Hadoop MapReduce jobs, which of the following are true for the input of a reducer?
   - ◯ The values associated with a key appear in sorted order: i.e. each value is strictly larger than the previous value.
   - ◯ Neither the keys nor values are in any predictable order.
   - ◯ The keys given to a reducer are sorted but the values associated with each key are in no predictable order.

8. (4 points) In a MapReduce job with M mappers and N reducers, which is a better guess as to how many pairs of machines will transfer data? (Pick one)
   - ◯ M+N: data will be copied from the M mappers to the head node, then from the head node to the N reducers.
   - ◯ M*N: data will be transferred directly from each mapper to each reducer.

9. (8 points) Map-reduce implementation.

   Briefly describe how to use MapReduce pattern to compute the left Outer Join of two tables A, B by column c. Recall that the result of a left outer join for tables A and B always contains all records of the "left" table (A), even if the join-condition does not find any matching record in the "right" table (B).

   You can assume that c is a primary key–i.e., for any value of "c", there is either no tuple in A such that *tuple.c* has that value, or only one tuple in A that has that value. You can also assume that the mapper's input includes all tuples in A and B, that in each call to the mapper, the value will hold a tuple, and that the function *fromTableA(tuple)* is true iff *tuple* is from table A.

   Mapper:

Reducer: