10605 Machine Learning for Large Datasets
Final Exam Practice Questions

William Cohen, William Wang, Siddarth Varia, Chun Chen

April 18, 2014

1 True or False. Please briefly explain your reasoning. (16 points)

1. Naive Bayes is a discriminative classifier, which directly models the conditional probability of the class variable given the observation.

2. When using the hashing trick, it is impossible to have two distinct features hashed into the same bucket.

3. In the approximate PageRank algorithm that we show in the class, increasing value of the approximation error hyperparameter ϵ will result in longer runtime of the algorithm.

4. Parallelizing the inference for LDA is difficult, because the choice of one latent topic often depends on other latent topics.

5. The phrase finding method we covered in class is based on semi-supervised learning.

6. Hadoop does not exploit data locality.

7. In Hadoop, the sorting between map and reduce phase happens completely in memory.

8. When bloom filter answers "Yes" for an item, the item may not actually exist in the set.

2 Multiple Choice. There maybe one or more correct answers to each question. (8 points)

1. Semi-supervised learning (SSL) on graphs. If we demote $\hat{Y}_{v,l}$ as the score of estimated label l on node v, $Y_{v,l}$ as the score of seed label l on node v, S as the seed node indicator (diagonal matrix), and W_{uv} as the weight of edge (u, v) in the graph, then the SSL optimization problem can be formulated as:

$$\arg\min_{\hat{Y}} \sum_{l=1}^{m} W_{uv}(\hat{Y}_{ul} - \hat{Y}_{vl})^2 \tag{1}$$

$$= \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l \tag{2}$$

$$s.t.\, Y_{ul} = \hat{Y}_{ul}, \forall S_{uu} = 1 \tag{3}$$

where the graph Laplacian $L = D - W$, where D is the degree matrix.

- A. The Laplacian matrix L must be positive semidefinite.
- B. The main idea of the objective function is that two nodes connected by an edge with high weight should be assigned similar labels.
- C. The idea of using graph Laplacian here is similar to Laplace smoothing.
- D. The solution of this objective function is harmonic.

2. You recently graduated from Cranberry-Lemon University, and were hired by a company to recognize cat faces on MeTube, an online video streaming website. You decide to take the pixels as features, sampling every 10 ms from the videos. To account the changes of these pixels, you decide to take the delta between each adjacent frame, and the delta of the detla. By the end of the day, you have a total of 100 million videos, 500 thousand labels, and 100 billion features. You define this as a multiclass classification problem, and decide to try training many binary classifiers using the standard logistic regression algorithm, which only maximize the log conditional likelihood function. Unfortunately, you only observe 3×10^{-6} classification accuracy. To save your job, which of the following options sound correct to you, and maybe reasonable to try?

   - A. Try increasing L_2 regularization coefficients to generate sparser estimations of the parameters.
   - B. Try using only features that occur more than three times in the data.
   - C. Try using multitask learning by looking at the correlations among different tasks (parameter vectors).
   - D. Use locality sensitive hashing to reduce the dimensionality of input data.

3 Short answers. Use a few sentences to answer the question. (32 points)

1. Recall the Rocchio's algorithm that we discussed in the class. One major difference between Rocchio's algorithm and the naive Bayes algorithm is that the former uses the TF-IDF representation. What is

TF-IDF? Explain the idea using one or two sentences, and write the formula for $TFIDF(w, d)$ where w is the word that you observe in the document d. You may define additional symbols if it is needed.

2. How would you parallelize the training process for perceptrons?

3. Assume $W[i, j]$ in the matrix W indicates the probability of walk from node i to j. Let $0 < \alpha < 1$. Let:

$$Y = I + \alpha W + (\alpha W)^2 + (\alpha W)^3 + ... + (\alpha W)^n$$

prove that Y can be approximated by $Y \approx (I - \alpha W)^{-1}$.

4. A problem of parallelizing SGD is that it often requires memory locking or synchronization techniques to maintain the global parameter vector, which significantly slow down the speedup. To solve this problem, your friend designed an asynchronous algorithm to improve the performance, which allows processors access to shared memory with

the possibility of over-writing each other's work. Now, you are given two datasets to test his algorithm: one is a sparse text dataset, the other one is a dense dataset of daily stock prices. Which one will work better? Why?

5. Explain why Count-Min Sketch only overestimates, but never underestimate.

6. You'd like to do a similarity join on a large set of documents (say, the news stories from RCV1). Of the randomized algorithms discussed in class, would any be helpful? If so, give a short explanation of how one of them might be used for this task.

7. Explain in a line, why Spark discussed in class is faster that Hadoop ?

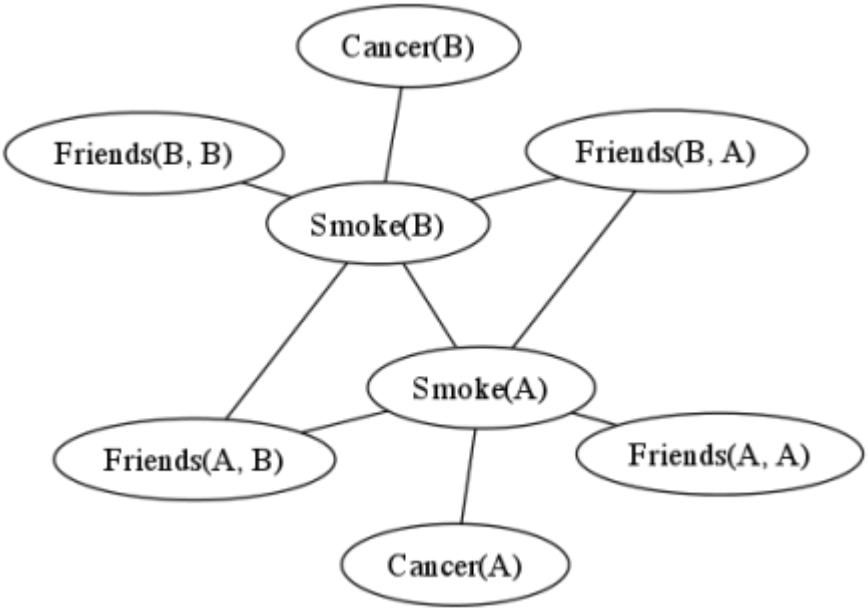

Figure 1: A example of grounded Markov Logic Network.

8. Figure 1 shows an example of grounded Markov Logic Network where the instances in the database have already been mapped to the predefined clauses. Now, circle the Markov blanket of the node **Smoke(B)**.

4 Wall Street (10 points)

Assume you are working on Wall street as a financial software developer. you have a file containing company's NYSE symbol and its stock price. The file contains multiple entries per company. your job is to find the average stock price for each company. The file is 10Gb and so you decide to use AWS EMR. The mapper is identity mapper & the reducer aggregates the

prices. In order to optimize your code, you decide to use the reducer as a combiner as well.

1. Do you think your code will produce the correct output? Please explain your answer. (4 points)

2. Describe a scheme for correctly using a combiner. (Hint: weighted averages.) (6 points)

5 Phase Finding (20 points)

Recall the pipeline of streaming phrase finding:

```
cat bigram.txt | sort -k1 | java -Xmx128m Aggregate 1 > bigram_processed.txt
cat unigram.txt | sort -k1 | java -Xmx128m Aggregate 0 > unigram_processed.txt
cat bigram_processed.txt | java -Xmx128m MessageGenerator > message.txt
cat message.txt unigram_processed.txt | sort -k1,1 | \
java -Xmx128m MessageUnigramCombiner > message_unigram.txt
cat message_unigram.txt bigram_processed.txt | sort -k1,2 | \
java -Xmx128m PhraseGenerator
```

Also consider the corpus as below:

```
Ipod classic from Apple is amazing. 2001
Apple will release Ipod shuffle in April. 2003
Steve Jobs declares himself interim CEO of Apple. 1997
```

We consider corpus before 2000 as background corpus and corpus after 2000 as corpus of interest.

1. We omit some pre-processing steps for generating bigram.txt/unigram.txt from the raw corpus. Assume the corpus is very large; can you design a pipeline to preprocess the raw corpus to bigram.txt/unigram.txt. Note the format of bigram.txt/unigram.txt is

   ```
   <text>\t<decade>\t<count>
   ```

   please transform the specific year to decade. Please write down the output format of each step. (4 points)

2. What are the entries in unigram.txt associated with "Apple", "Ipod", "release", and "Steve"? (4 points)

3. What are the entries in unigram_processed.txt associated with "Apple", "Ipod", "release" and "Steve"? (4 points)

4) What are the outputs of message.txt associated with "Ipod Classic", "Ipod Shuffle" and "interim CEO". (4 points)

5) What are the outputs of message_unigram.txt associated with "Ipod Classic", and "from Apple"? (4 points)

6 PageRank with Hadoop (14 points)

Let's say, you have to implement Pagerank algorithm using MapReduce. To start with, you have the edge file(src_id dest_id) and the initial pagerank values file(id value). Using these two files, propose the 2 map reduce jobs to iteratively compute pagerank values. Don't worry about convergence. Briefly describe the input and output for Map1, Reduce1, Map2 & Reduce2 in terms of key and value. Assume you know N, where N is the total number of pages. Also assume the pagerank value has some special characters in the beginning to help you differentiate between node id and pagerank value. Hint: $pr_i = 0.15/N + 0.85 \times \sum_{j \in INEDGES_i} pr_j/d_j$