CS595 Assignment 1

Jinyang Li

3. (5 points) Submit a document with very brief answers (or bullet points) to the following questions:

   - **Describe any prior experience you might with, data mining, machine learning, statistics, data science and big data**

     I'm not sure wether it is, I used a web crawler to pull movie information in various websites and store them in my own database. Plus, I think google advertise is kind of big data, it collect your behavior and likelihood of advertisement, finally get you the right ADs that you have more intend to click.

   - **Share any big data interests and personal learning goals for the course**

     As I mentioned in previous question, I'd like to learn a more sufficient way to write a web crawler and I'd like to learn how to precisely deliver user the content that he/she might like.

   - **Indicate if there are additional topics in the scope of the course of special interest to you**

     User behavior analyze.

   - **Indicate if you have access to big data technology and data sets, of what nature, and in what industry.**

     I'm the owner of a WeChat Official Account, which is developed by Tencent. It can deliver content to users everyday and receive messages from users.

   - Do you have any anticipated personal issues such expected absences or other necessary accommodations with course impact? (Of course, these will be held in strictest confidence.) **No**

- **The Parable of Google Flu (just 3 pages!)**

4. **(5 points) Summarize the main points of the above article and your thoughts (questions you might want to ask the authors, areas where you disagree, other comments)**

The Google Flu Trends has disadvantages. Author pointed out those disadvantages and then stated out the problems that needs to be considered during any big data technology predication applications. Also, author recommend combine big data and traditional statistics together during predication process.

AS author said, "*Big data hubris" is the often implicit assumption that big data are a substitute for, rather than a supplement to, traditional data collection and analysis. Elsewhere, we have asserted that there are enormous scientific possibilities in big data. However, quantity of data does not mean that one can ignore foundational issues of measurement and construct validity and reliability and dependencies among data.* Google collect data from search engines. When flu becomes big during 2009 - 2010, it obviously users would use Internet search regarding the flu topic. It is true the relativity that between flu and this 'user behavior' is high, but Google Flu Trends made mistake that it ignore the causality when the output of instruments designed to produce valid and reliable data amenable for scientific analysis. Also, the google's search algorithm is another reason it falls. Google's predation is based on relative magnitude of certain search, plus, the algorithm changes quickly, search behavior is not just exogenously determined, it is also endogenously cultivated by the service provider.

In my opinion, I agree with the point that big data should combine with traditional statistics prediction. Traditional way is slow, cannot collect samples in a sufficient way, however it important when questions become seriously. Big data is the future, but It needs to be more "strict".

- Byzantine Fault Tolerant MapReduce

The fault tolerance mechanisms of current MapReduce implementations, namely Hadoop, cannot deal with such accidental arbitrary faults or accidental Byzantine faults. They cannot be detected using checksums and often do not crash the task they affect, so they can silently corrupt the result of a task. They have to be detected and their effects masked by executing each task more than once. This basic idea was proposed in the context of volunteer computing to tolerate malicious volunteers, that return false results of the tasks they were supposed to execute [29]. That work, however, considered bag-of-tasks applications, which are simpler than MapReduce jobs. A similar but more generic solution is Byzantine fault-tolerant state machine approach, in which a set of programs are executed in parallel by different servers that execute commands in the same order. This approach, however, is not directly applicable to the replication of MapReduce tasks, only of a service that follows the client-server model (e.g., a file server). A naive solution for MapReduce would be to execute each job twice and re-execute it if the results do not match, but its cost is excessive in case there is a fault.

In my opinion, every new algorithm has space to improve. We need face the disadvantages and improve it over time.