Guidelines for the report in the seminar of big data management 2017

Jiaheng Lu

University of Helsinki

1. Difference between presentations and reports

First, please note that the goals of presentations and reports in this seminar are different:

- Presentations: Present to the audiences and let them understand your topic.
- Reports: Show your own critical thinking and new ideas on this topic based on the papers you read.

Therefore, a presentation is like "*an introduction to one topic*", but a report is like "*a paper review and an opportunity for critical thinking*". In presentations, students are expected to introduce the research problems and challenges, and show intuitive examples and applications. But **in reports, students are expected to give novel ideas and to show critical thinking based on the papers students read**. So the purposes are different for presentations and reports.

2. Goal of report writing

A general goal of report writing is to show that students have good understanding on one specific topic of big data management and students can produce independently a well-formed and finished written report on it. Therefore, the following questions should be answered in the report.

1. What are the research problems?

2. What are the strengths of the previous paper(s)?

3. What are the main weaknesses of the previous paper(s)?

4. If you were to solve this problem, what would you do?

3. Structures of a report

The main structure of a report includes the following sections.

- **Abstract**

Use one or two paragraphs to summarize the main contributions of your report. Some example sentences are like: "I read two papers [1,2] on big graph data in this seminar. In this report, I review their main contributions on the problem of big graph processing. "

- **Keywords**

Give three to five keywords in the report.

- **Section 1. Introduction**

Give the motivation of the research problem and discuss the main results of previous papers. The length of Introduction is around 1.5 - 2 pages.

At the end of the Introduction, give the organization of the rest of the report. An example is like: "The rest of the report will be divided as follows. In Section 2, I will discuss how big data applies to health by providing explanations. Then, in Section 3, I will give an overview of the potential areas… Finally, I will give a brief conclusion to summarize this report."

- **Section 2. Preliminaries**

Describe some basic definitions and theorems for readers to understand this report. The length is around 0.5 - 2 pages.

- **Section 3. Main results of previous works**

This part can be extended to several sections which depend on the number of papers you read and the contents you want to present. Please add more examples and figures to illustrate the results. **Do not directly copy the sentences from previous papers**. Rephrase the ideas according to your own understanding.

- **Section 4. Strong points of papers**

Several strong points or interesting parts of papers should be mentioned and explained in this section. The length is around 0.5 – 1 pages.

Some examples are like "*This paper is well-written and in general it gives me a very good impression.", "I am especially curious to understand the setting of their big data experiments*.", "*The main architecture of big data management system is described very well in this paper*".

- **Section 5. Weak points of papers**

Several weak points of papers are found and explained in this section. The length is around 0.5 – 1 pages. Some examples are like " *The main weak point I figured out is that there is no enough explanation on…*" , " *The concept is not well defined and it is confused for readers*".

- **Section 6. My ideas (or algorithms or experiments)**

Give new ideas/algorithms/experiments of the author on the research problem. This part is very important, because it shows the potential of the author to be an independent innovative researcher. This section can be as long as possible.

- **Section 7. Conclusion**

Summarize the research problem and the main contributions of previous papers. The main weakness of previous works should be also mentioned here. Some future works can be described as well.

References

Add the reference list including 5-10 papers. Each paper should be cited in the report.

Note that:

1. The deadline of the first version of your report is 13th March, 2017.
2. The total length of a report is 6-10 pages. You may choose the single column or double columns. Generally speaking, a single column report may take more pages than a double column one. Please submit the PDF version of your report to Moodle page.
3. Please add more examples and figures in your report to illustrate the ideas and algorithms. There are at least one example and one figure for each report.
4. Please make a careful proofreading before the submission of your report. Read your report at least three times before the submission.

Example of abstract

The amount of data is growing with ever larger velocity into larger volumes with larger variety calling for better, faster and more accurate analyzing methods. Aggregation queries have been at the heart of business intelligence and data analysis for a long time. These aggregation queries however are expensive and time consuming because the query must consume the entire data set. To reduce costs of these queries different statistical analysis method called data sampling has been adopted. The basic idea of data sampling is that instead of performing queries over the entire data set, a sample will first be taken from the whole database and the query is done over that sample. This sample would be taken with an error-bound so that the sample would actually be statistically valid. This method would thus reduce the time taken by the query and the error-bound would maintain the accuracy and validity of the query.

In this report I will explore two papers related to this issue. First, Error-bounded Sampling for Analytics on Big Sparse Data, will present the error-bounded stratified sampling and compare it to uniform random sampling and the second one, Congressional Samples for Approximate Answering of Group-by Queries, will compare Congressional sampling to uniform random sampling. I will find that while dealing with many different cases, there are problems with using uniform random sampling that these two methods of sampling will try to address. Results of these show major performance gains compared to uniform sampling.

Example of Introduction

Increasingly, a massive amount of SBD (spatial big data) is being collected because of the rising amount of mobile technologies (such as mobile phones and wearable technologies) and location-aware Internet browsers [1]. Spatial data means all types of data objects or elements that have geographical information present in it. SBD is so massive spatial data that it exceeds the capacity of commonly used spatial computing systems due to its volume, variety and velocity [4]. For example, traffic speeds for each road segment in America (that are used in economic routing services) is spatial big data. Applications that utilize SBD have different workloads depending on the time of the day and the location where they are being used [1].

SBD is getting easier to collect as sensors are becoming more and more common. The problem is not collecting the data but retaining computational efficiency within systems. Growing diversity of SBD increases computational cost compared to traditional routing services (such as travel-distance and travel-time measurement services) because SBD uses richer information, larger sets of choices and more preference functions (like fuel efficiency, greenhouse gas emissions, safety) [4]. Secondly, when using cloud computing

services one must pay attention to storing SBD into the cloud. Partitioning of SBD in the cloud is crucial because if the data partitions are not being accessed, the servers storing them remain idle. As an example, Amazon and Microsoft users are charged based on the amount of time they reserve in each server without considering utilization. Therefore, the increase in the server utilization leads into less number of servers to support the same workload and having to pay for less servers saves cost [1]. Thirdly, most of the SBD changes and new data should be applied to the system in hourly or daily basis so there is a need to rewrite or apply new data. Therefore, repartitioning of the data is needed. However, repartitioning from the beginning might even takes several hours and it halts down the system. A more practical solution is needed in order to keep the system alive when doing data repartitioning.

This seminar report is based on several papers that discuss managing SBD in the cloud. Firstly, a cost-efficient way of partitioning SBD in the cloud is presented. That means having partitions of diverse access patterns in the same server to maximize server utilization. After that, AQWA is presented which is a query-workload aware partitioning system that repartitions data on the fly as query workloads change and more SBD becomes available. AQWA tries to partition SBD so that query workload for each partition is as equal as possible. If query workloads are equal, servers are less likely to become overloaded. In addition, one primary object of AQWA is to have datasets partitioned so that when answering to a query, server has to look from as minimal spatial datasets as possible. Approaches of both of those papers can be combined to build the best way to partition and divide SBD in the cloud system. The last paper describes a complete solution called PAIRS to manage SBD in a cloud based system. Starting from choosing spatial data formats to finally having SBD stored in the cloud.

The rest of this seminar report will be divided as follows. Section 2 gives an overview of SBD with some example use cases. Then, Section 3 discusses a cost-efficient way of partitioning SBD in the cloud and Section 4 presents a system called AQWA for a detailed way of partitioning SBD in the cloud. Section 5 introduces a complete platform called PAIRS for handling SBD efficiently. Finally, the last chapter gives a summary of this seminar report.

Example of the strengths of the paper

The paper is well-written and in general gives a very good impression. The authors have a strong practical focus: their task is to build a useful resource for the real-life applications, not to invent a beautiful algorithm. Thus, they tell openly than they use simple heuristics or a manual work. I suppose that some people may consider that as a disadvantage of the paper; at least, at the NLP conferences it is very hard to publish a paper that involves a manual work instead of pure machine learning. However, in many practical applications it may be much easier and faster to write a set of rules manually instead of trying to learn them from the data. Even when machine learning is used it may be possible to do simple heuristic fixes that would improve performance of the system comparing with the fully automatic analysis. Rule-based approaches are wildly used in practical applications and publication of the best practices and development methodologies is highly appreciated. The authors claim that this is the first paper that describes an end-to-end KB development project. Indeed, it was very interested to read and the paper may be useful for many teams.

I was especially curious to read about the team size and structure: in total, four persons were working on the KB: a developer, a data analyst, a system person (half time) and a UI developer (half time). I had never seen

such details in any other paper. I would expect to see there more human experts and was surprised how relatively small manual work was required to maintain the KB. The reasons for that, I guess, is a carefully developed procedure and tools, including in-house editing language and a graphical interface, as well as automatic preprocessing whenever it was possible.

Example of the weakness of the paper

Despite I like the paper in general I think it can gain from some formal evaluations. Obviously it is hard to evaluate a KB since there is no gold standard and the KB is too huge for the direct annotation. However, the authors mention that human experts weakly evaluate a) all KB node that have more than 200 children, b) a random set of paths going from root to a leaf. It would be interesting to know how many correction are done during these evaluations: this would give some impression about automatic KB construction quality. It would also be interesting to see how the number of corrections changes over time: this might be used to estimate an effectiveness of the previous manual work. Another way to evaluate the KB may be to compare it to some freely available analogues, such as Freebase or DBpedia (both mentioned in the related works section); but these type of evaluating is obviously time consuming and may require too much e orts from the team.

Moreover, the numerical properties of KB provided by the authors look more like curious facts (concepts with maximum children) than a useful information. It would be more useful to see a distribution of in- and out- degree of the nodes and a distribution of lengths of paths from the root to the leaves.

Example of suggestions

To be more concrete I think the paper may benefit by adopting some techniques from the relation extraction papers, such as [5,4, 14, 10, 1, 7]. Another example of the highly relevant paper that should have been mentioned in the related works section is [8]: this is about building a huge database by combining WordNet and Wikipedia, as well as some other sources. The paper not only describes the algorithm for that but also give some hints on how such KB may be evaluated.