Big-data Analytics: Challenges and Opportunities

Chih-Jen Lin
Department of Computer Science
National Taiwan University

Talk at 台灣資料科學愛好者年會, August 30, 2014

Everybody talks about big data now, but it's not easy to have an overall picture of this subject

In this talk, I will give some personal thoughts on technical developments of big-data analytics. Some are very pre-mature, so your comments are very welcome

Outline

1. From data mining to big data
2. Challenges
3. Opportunities
4. Discussion and conclusions

Outline

1. From data mining to big data
2. Challenges
3. Opportunities
4. Discussion and conclusions

From Data Mining to Big Data

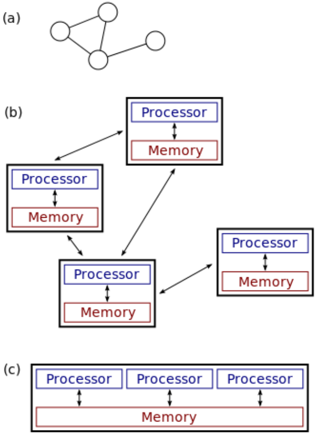
- In early 90's, a buzzword called data mining appeared
- Many years after, we have another one called big data
- Well, what's the difference?

Status of Data Mining and Machine Learning

- Over the years, we have all kinds of effective methods for classification, clustering, and regression
- We also have good integrated tools for data mining (e.g., Weka, R, Scikit-learn)
- However, mining useful information remains difficult for some real-world applications

What's Big Data?

- Though many definitions are available, I am considering the situation that data are larger than the capacity of a computer
- I think this is a main difference between data mining and big data
- So in a sense we are talking about distributed data mining or machine learning



(a), (b): distributed systems

Image from Wikimedia

From Small to Big Data

Two important differences:

Negative side:

- Methods for big data analytics are not quite ready, not even mentioned to integrated tools

Positive side:

- Some (Halevy et al., 2009) argue that the almost unlimited data make us easier to mine information

I will discuss the first difference

Outline

Possible Advantages of Distributed Data Analytics

Parallel data loading

- Reading several TB data from disk is slow
- Using 100 machines, each has $1/100$ data in its local disk $\Rightarrow 1/100$ loading time
- But having data ready in these 100 machines is another issue

Fault tolerance

- Some data replicated across machines: if one fails, others are still available

Possible Advantages of Distributed Data Analytics (Cont'd)

Workflow not interrupted

- If data are already distributedly stored, it's not convenient to reduce some to one machine for analysis

Possible Disadvantages of Distributed Data Analytics

- More complicated (of course)
- Communication and synchronization

  Everybody says moving computation to data, but this isn't that easy

Going Distributed or Not Isn't Easy to Decide

- Quote from Yann LeCun (KDnuggets News 14:n05)
  "I have seen people insisting on using Hadoop for datasets that could easily fit on a flash drive and could easily be processed on a laptop."
- Now disk and RAM are large. You may load several TB of data once and conveniently conduct all analysis
- The decision is application dependent
- We will discuss this issue again later

Distributed Environments

- Many easy tasks on one computer become difficult in a distributed environment
- For example, subsampling is easy on one machine, but may not be in a distributed system
- Usually we attribute the problem to slow communication between machines

Challenges

- Big data, small analysis

  versus

  Big data, big analysis
- If you need a single record from a huge set, it's reasonably easy
- For example, accessing your high-speed rail reservation is fast
- However, if you want to analyze the whole set by accessing data several time, it can be much harder

Challenges (Cont'd)

- Most existing data mining/machine learning methods were designed without considering data access and communication of intermediate results
- They iteratively use data by assuming they are readily available
- Example: doing least-square regression isn't easy in a distributed environment

Challenges (Cont'd)

So we are facing many challenges

- methods not ready
- no convenient tools
- rapid change on the system side
- and many others

What should we do?

Outline

Opportunities

- Looks like we are in the early stage of a research topic
- But what is our chance?

Outline

3. Opportunities
   - Lessons from past developments in one machine
   - Successful examples?
   - Design of big-data algorithms

Algorithms for Distributed Data Analytics

This is an on-going research topic.

Roughly there are two types of approaches

1. Parallelize existing (single-machine) algorithms
2. Design new algorithms particularly for distributed settings

Of course there are things in between

Algorithms for Distributed Data Analytics (Cont'd)

- Given the complicated distributed setting, we wonder if easy-to-use big-data analytics tools can ever be available?
- I don't know either. Let's try to think about the situation on one computer first
- Indeed those easy-to-use analytics tools on one computer were not there at the first day

Past Development on One Computer

- The problem now is we take many things for granted on one computer
- On one computer, have you ever worried about calculating the average of some numbers?
- Probably not. You can use Excel, statistical software (e.g., R and SAS), and many things else
- We seldom care internally how these tools work
- Can we go back to see the early development on one computer and learn some lessons/experiences?

Example: Matrix-matrix Product

- Consider the example of matrix-matrix products

$$C = A \times B, \quad A \in R^{n \times d}, B \in R^{d \times m}$$

  where

$$C_{ij} = \sum_{k=1}^{d} A_{ik} B_{kj}$$

- This is a simple operation. You can easily write your own code

Example: Matrix-matrix Product (Cont'd)

- A segment of C code (assume $n = m$ here)

```
for (i=0;i<n;i++)
    for (j=0;j<n;j++)
    {
        c[i][j]=0;
        for (k=0;k<n;k++)
            c[i][j] += a[i][k]*b[k][j];
    }
```

- For $3,000 \times 3,000$ matrices

```
$ gcc -O3 mat.c
$ time ./a.out
3m24.843s
```

Example: Matrix-matrix Product (Cont'd)

- But on Matlab (single-thread mode)

```
$ matlab -singleCompThread
>> tic; c = a*b; toc
Elapsed time is 4.095059 seconds.
```

Example: Matrix-matrix Product (Cont'd)

- How can Matlab be much faster than ours?
- The fast implementation comes from some deep research and development
- Matlab calls optimized BLAS (Basic Linear Algebra Subroutines) that was developed in 80's-90's
- Our implementation is slow because data are not available for computation

Example: Matrix-matrix Product (Cont'd)

CPU

$\downarrow$

Registers

$\downarrow$

Cache

$\downarrow$

Main Memory

$\downarrow$

Secondary storage (Disk)

- $\uparrow$: increasing in speed
- $\downarrow$: increasing in capacity
- Optimized BLAS: try to make data available in a higher level of memory
- You don't waste time to frequently move data

Example: Matrix-matrix Product (Cont'd)

- Optimized BLAS uses block algorithms

$$A \times B = \begin{bmatrix} A_{11} & \cdots & A_{14} \\ & \vdots & \\ A_{41} & \cdots & A_{44} \end{bmatrix} \begin{bmatrix} B_{11} & \cdots & B_{14} \\ & \vdots & \\ B_{41} & \cdots & B_{44} \end{bmatrix}$$
$$= \begin{bmatrix} A_{11}B_{11} + \cdots + A_{14}B_{41} & \cdots \\ \vdots & \ddots \end{bmatrix}$$

- If we compare the number of page faults (cache misses)

  Ours: much larger

  Block: much smaller

Example: Matrix-matrix Product (Cont'd)

- I like this example because it involves both
  - mathematical operations (matrix products), and
  - computer architecture (memory hierarchy)
- Only if knowing both, you can make breakthroughs

Example: Matrix-matrix Product (Cont'd)

- For big-data analytics, we are in a similar situation
- We want to run mathematical algorithms (classification and clustering) in a complicated architecture (distributed system)
- But we are like at the time point before optimized BLAS was developed

Algorithms and Systems

- To have technical breakthroughs for big-data analytics, we should know both algorithms and systems well, and consider them together
- Indeed, if you are an expert on both topics, everybody wants you now
- Many machine learning Ph.D. students don't know much about systems. But this isn't the case in the early days of computer science

Algorithms and Systems (Cont'd)

- At that time, every numerical analyst knows computer architecture well.
- That's how they successfully developed floating-point systems and IEEE 754/854 standard

Example: Machine Learning Using Spark

- Recently we developed a classifier on Spark
- Spark is an in-memory cluster-computing platform
- Beyond algorithms we must take details of
  - Spark
  - Scala

  into account
- For example, you want to know
  - the difference between **mapPartitions** and **map** in Spark, and
  - the slower **for** loop than **while** loop in Scala

Example: Machine Learning Using Spark (Cont'd)

- During our development, Spark was significantly upgraded from version 0.9 to 1.0. We must learn their changes
- It's like when you write a code on a computer, but the compiler or OS is actively changed. We are in a stage just like that.

Outline

3 Opportunities
- Lessons from past developments in one machine
- Successful examples?
- Design of big-data algorithms

Example of Distributed Machine Learning

- I don't think we have many successful examples yet
- Here I will show one: CTR (Click Through Rate) prediction for computational advertising
- Many companies now run distributed classification for CTR problems

Example: CTR Prediction

- Definition of CTR:

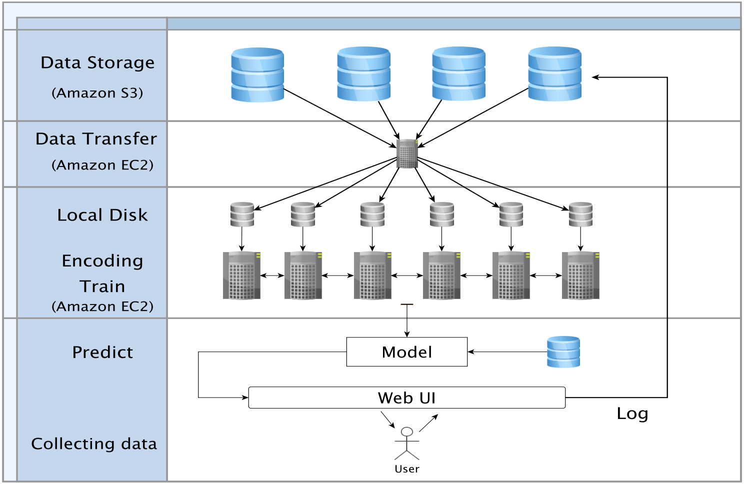
$$\text{CTR} = \frac{\#\ \text{clicks}}{\#\ \text{impressions}}.$$

- A sequence of events

Not clicked	Features of user
Clicked	Features of user
Not clicked	Features of user
...	...

- A binary classification problem.

Example: CTR Prediction (Cont'd)



Outline

3. Opportunities
   - Lessons from past developments in one machine
   - Successful examples?
   - Design of big-data algorithms

Design Considerations

- Generally you want to minimize the data access and communication in a distributed environment
- It's possible that

  method A better than B on one computer

  but

  method A worse than B in distributed environments

Design Considerations (Cont'd)

- Example: on one computer, often we do batch rather than online learning

  Online and streaming learning may be more useful for big-data applications

- Example: very often we design synchronous parallel algorithms

  Maybe asynchronous ones are better for big data?

Workflow Issues

- Data analytics is often only part of the workflow of a big-data application
- By workflow, I mean things from raw data to final use of the results
- Other steps may be more complicated than the analytics step
- In one-computer situation, the focus is often on the analytics step

How to Get Started?

- In my opinion, we should start from applications
- Applications $\rightarrow$ programming frameworks and algorithms $\rightarrow$ general tools
- Now almost every big-data application requires special settings of algorithms, but I believe general tools will be possible

Outline

Risk of This Topic

- It's unclear how successful we can be
- Two problems:
    - Technology limits
    - Applicability limits

Risk: Technology limits

- It's possible that we cannot get satisfactory results because of the distributed configuration
- Recall that parallel programming or HPC (high performance computing) wasn't very successful in early 90's. But there are two differences this time
  1. We are using commodity machines
  2. Data become the focus
- Well, every area has its limitation. The degree of success varies

Risk: Technology Limits (Cont'd)

- Let's compare two matrix products:

  Dense matrix products: very successful as the final outcome (optimized BLAS) is much better than what ordinary users wrote

  Sparse matrix products: not as successful. My code is about as good as those provided by Matlab

- For big data analytics, it's too early to tell
- We never know until we try

Risk: Applicability Limits

- What's the percentage of applications that need big-data analytics?
- Not clear. Indeed some think the percentage is small (so they think big-data analytics is a hype)
- One main reason is that you can always analyze a random subest on one machine
- But you may say this is a chicken and egg problem – because of no available tools, so no applications??

Risk: Applicability Limits (Cont'd)

- Another problem is the mis-understanding
- Until recently, few universities or companies can access data center environments. They therefore think those big ones (e.g., Google) are doing big-data analytics for everything
- In fact, the situation isn't like that

Risk: Applicability Limits (Cont'd)

- A quote from Dan Ariely, "Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it ..."
- In my recent visit to a large company, their people did say that most analytics works are still done on one machine

Risk: Applicability Limits (Cont'd)

- A quote from Dan Ariely, "Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it ..."
- In my recent visit to a large company, their people did say that most analytics works are still done on one machine

Risk: Applicability Limits (Cont'd)

- A quote from Dan Ariely, "Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it ..."
- In my recent visit to a large company, their people did say that most analytics works are still done on one machine

Risk: Applicability Limits (Cont'd)

- A quote from Dan Ariely, "Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it ..."
- In my recent visit to a large company, their people did say that most analytics works are still done on one machine

Open-source Developments

- Open-source developments are very important for big data analytics
- How it works:

  The company must do an application X. They consider an open-source tool Y. But Y is not enough for X. Then their engineers improve Y and submit pull requests
- Through this process, core developers of a project are formed. They are from various companies

Open-source Developments (Cont'd)

- For Taiwanese data-science companies, I think we should actively participate in such developments
- Indeed industry rather than schools are in a better position to do this

Conclusions

- Big-data analytics is in its infancy
- It's challenging to development algorithms and tools in a distributed environment
- To start, we should take both algorithms and systems into consideration
- Hopefully we will get some breakthroughs in the near future