Big data computing

Instructors:
Luca Becchetti, Ioannis Chatzigiannakis

The plan for today

- A checkist/guided tour of things you may want/need to review
- A very simple programming starter
- A few puzzles
  - Toy examples where you have to be a bit flexible and apply the knowledge you already possess

Prerequisites

- Linear algebra
- Calculus and basic knowledge of probability theory and statistics
- Programming, fundamental algorithms and data structures
- Attendance at the data mining class is a plus

Goal of these slides

- Your self-assessment
  - Should I (possibly with a group of colleagues) review the material suggested at the end of these slides?
  - Which topics should I review?
- My tailoring of the course
  - What previous knowledge should I assume from students?
  - Which foundational topics should I revisit?
- *In any case: we are here to learn*
  - *If you have questions/doubts: **ask**!*
  - *I cannot teach a basic course in probability or linear algebra, but I can advise on what to review*

That boring probability stuff that seemed so disconnected from the real world ...

Checklist/1

- Assume you have a probability P(·) over some sample space Ω
  - E.g.: Ω = the set of all possible prices of VW's stocks next monday at 8am
  - What is an *event* defined over Ω?
    - E.g.: E = (price > 115,00 €)

Topic: Axioms of probability and sample spaces

Checklist/2

- Assume A and B are events, with $A \cap B = \emptyset$.
  - What can you say about $P(A \cup B)$?
  - What can you say in general, even if the above assumption does not hold?
- Consider rolling a die and observing the outcome
  - What is the sample space?
  - What is P(outcome is odd), assuming a fair die?
- Assume we roll a die *twice* and we record both outcomes
  - What is the sample space in this case?
  - Let x = <outcome of first toss> and y = <outcome of second toss>. What is P(x + y < 6) ? Assume a fair die and independence of tosses

Topics: axioms of probability, events, random variables

Checklist/3

- Assume Toshiro in Tokyo and Luca in Rome roll two dice at the same time
  - Let x = <outcome of Luca's toss> and y = <outcome of Toshiro's toss>
  - What is P(x = 6 ∩ y = 2) ?
- What is the probability that in rolling two dice the sum is 8 given that the sum was even?

Topics: independence, conditional probability

Checklist/4

- Assume $E_1....E_n$ are events, such that i) $\cup_i E_i = \Omega$ and ii) $E_i \cap E_j = \emptyset$ unless i = j
  - Let A be another event over Ω. Prove that $P(A) = \sum_i P(A|E_i) \cdot P(E_i)$
  - What happens if A is independent of the E_i's?

Topics: axioms of probability, law of total probability, independence

Checklist/5

- Assume we roll a die
  - What is the expected value of the outcome?
- Assume we throw two dice in a row
  - What is the expectation of the sum?
  - How do you compute it?
- Assume X_i: $\Omega \rightarrow \Re$ are real variables over the same sample space
  - What can we *always* say about $E[\sum_i E_i]$?

Topic: expectation and its properties

Checklist/6

- Assume we roll a die
  - What is the variance of the outcome?
- Assume we throw two dice in a row
  - What is the variance of the sum?
  - How do you compute it?
- Assume $E_1 ... E_n$ are events over the same sample space
  - Is $\mathbf{var}(\cup_i E_i) = \sum_i \mathbf{var}(E_i)$ in general?
  - If not, when does the equality above hold?

Topic: expectation, variance and their properties

Checklist/7

- Assume I perform an experiment that can either succeed with probability p or fail with probability 1 – p
  - How is the corresponding indicator variable called?
- Assume I repeat the experiment n times *independently* and I record the number X of successes
  - How is the corresponding random variable called?
  - What is E[X] ?
- **Puzzle:** assume I have an undirected graph G = (V, E) and assume I sample one edge u.a.r. What is the probability that I sample a specific vertex u ?

Topic: basic probability distributions

Is there more?

- Yes, potentially a lot that is used
  - Concentration inequalities
    - E.g.: bound $P(|X – E[X]| > \varepsilon E[X])$
    - Markov, Chebyshev, Chernoff, Azuma …
- Stochastic processes
  - Essentially Markov chains
- Basics of statistics
- We try to review them **if** needed

References

- A quick tour of the very basics of (discrete) probabilty theory can be found here
- Another one from a course I previously taught can be found here
- The first four chapters of the following book cover most of your needs:
  - Mitzenmacher, Michael, and Eli Upfal. Probability and computing: Randomized algorithms and probabilistic analysis. Cambridge University Press, 2005
- Probably, your textbook from your introductory course in probability will suffice too
- Refer to instructor for further info

Spectra, spectra everywhere!

Linear algebra: what you may need

- Matrices and vectors recur often
  - Basic operations: products of matrices, scalar products, cosine
  - Norms
  - Ranks
- Spectral analysis
  - Mostly the basics: the introductory material of Chapter 11 of the book should suffice for most needs

Warming up

A programming starter/1

- Movielens dataset:
  - <UID>::<MID>::<Rating>::<Timestamp>

```
2::1207::4::978298478
2::1968::2::978298881
2::3678::3::978299250
2::1244::3::978299143
2::356::5::978299686
2::1245::2::978299200
2::1246::5::978299418
2::3893::1::978299535
2::1247::5::978298652
3::3421::4::978298147
3::1641::2::978298430
3::648::3::978297867
3::1394::4::978298147
3::3534::3::978297068
```

A programming starter/2

- Simple solution:
  - Dictionary with MID as key
  - Count no. users that
- Receive file as input
- Return ID of most popular movie (number of views)

```
2::1207::4::978298478
2::1968::2::978298881
2::3678::3::978299250
2::1244::3::978299143
2::356::5::978299686
2::1245::2::978299200
2::1246::5::978299418
2::3893::1::978299535
2::1247::5::978298652
3::3421::4::978298147
3::1641::2::978298430
3::648::3::978297867
3::1394::4::978298147
3::3534::3::978297068
```

A programming starter/3

```
def most_popular(file):
    movies = {}
    f = open(file, "r")
    for line in f:
        record = line.split("::")
        if record[1] in movies:
            movies[record[1]] += 1
        else:
            movies[record[1]] = 1
    f.close()
    movie_list = list(movies.items())
    movie_list = sorted(movie_list, key = lambda x: x[1], reverse = True)
    return movie_list[0]

file = "../spark-1.4.1-bin-hadoop2.6/data/MyData/Movielens/ratings.dat"
print(most_popular(file))
```

A simple task, computationally demanding if dataset is large
Can we parallelize it?

Is “big” really big/1

- You are given 1 TB of 32bit integers to sort
  - How long does it take?
    - Fetching each item directly from disk
    - Being a bit smarter
- Variant of an example considered in the advanced programming course by Demetrescu

Is “big” really big/2

execute typical instruction	1 ns
fetch from new disk location (seek)	8,000,000 ns
read 1MB sequentially from disk	20,000,000 ns
send packet US to Europe and back	150 ms = 150,000,000 ns

Some useful times

- Assume efficient sorting algorithm (O(n log n) cost) → n log n for simplicity
- Assume 4 GB RAM

Is big really big/3

- If we assume fetching each value from disk:
  - Time ≈ 8ms * nlog n with n = #integers to sort
  - We have $n = 2^{40}/4 = 2^{38}$
  - Back of the envelope calculation gives you 1000 years approximately
  - Note that actual computing times are negligible

Is big really big/4

- We read the file in chunks of 4GB each → $2^{40}/2^{32} = 256$ chunks
- We sort each chunk in main memory using an efficient algorithm → Time for such a sort: ≈ $2^{28} * 28 * 1ns \approx 7.5s$
- We merge the sorted chunks by reading them in parallel from disk and performing buffering
- We need 20ms to transfer 1MB from disk to RAM
- We essentially must read the 256 sorted chunks (1 TB total) and write the result back to disk

Just to have a first feeling: assuming buffering allows us to neglect seek times (not true in fact) and neglecting computational costs (reasonable) merge cost is dominated by two file transfers from and to disk, 1 TB each

$$\text{Time} \approx 2^{40}/2^{20} * 20ms \approx 6hrs$$

Maybe, we can make it with in 1 day

Data collection traps

- Assume we have a political election in country C with two parties A and B
- Assume the ideal case in which we collect a sample of Facebook users (representative of the totality of country C's Fb users), 60% of which like party A
- What can we say about the accuracy of this predictor of the election's outcome?
- Elaborate on the factors affecting the accuracy of our estimate

Data analysis traps

- Consider a city with 1000000 inhabitants and assume each of them knows an average of 20 people on a personal basis
- Assume that, on any given day, a generic person A dreams of one of her acquaintances B with probability 1/10000
- Assume further that, on any given day, a generic person B calls any of her acquaintances B with probability 1/1000
- Work out the expected number of acquaintance pairs (A, B) that have the following experience over 1 year span
  - A dreams of B on any given day and B calls A the day after

Data interpretation traps

- Sleeping with one's shoes on is strongly correlated with waking up with a headache.
- Therefore, sleeping with one's shoes on causes headache