Lecture 1. Computing: What's in a name?

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

April, 2019

Reading: CLRS Ch. 1, Ch. 3 (without the section on o, w notation)

1. Computer programming
   1.1 knowing the programming language syntax
   1.2 how to write good code
2. Algorithms and data structures
3. Complexity theory
4. Optimization
5. Numerical analysis
6. Systems/Computer architectures

1.2 Good code

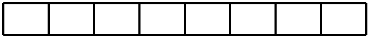
- does what it's supposed to do
- handles errors (you know what it's doing)
  Example: computing the variance
- does it efficiently (time and resources)
- easy to understand, debug, modify – reusing code

2 Algorithms and data structures

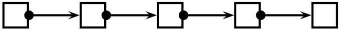
Examples of algorithms

- compute the mean of a sample
- compute the median
- shortest path in graph
- nearest neighbor, all neighbors within radius R

Data structures



a vector (array)



a linked list

Example: list of students in class, alphabetically sorted. Operations: add, drop, check if it's present in list, total number students.

3 Complexity

If size of the input is n, how many operations/seconds/kbytes does an algorithm need to compute the output?

Real world and theoretical measures of efficiency

- **number of operations** (theoretic)
- **memory** (theoretic)
- checkpointing
- cache hits
- disk access
- real time
- easy to modify

- worst case
- average case

Examples. Asymptotic maximum times for

- sorting $n \log n$
- max n
- median and k-th order statistic n
- shortest path in graph n
- maximum clique in a graph NP-hard
- longest common subsequence of k strings of length n NP-hard (n^k)
- minimum spanning tree n^2
- minimum spanning tree with degree $\leq k$ NP-hard
- finding the prime factors of an integer: assumed hard

4 Optimization Optima of functions (usually) over continuous domains, with constraints.
An easy optimization problem: $\min_x ax^2 + bx + c$ for $x \in (-\infty, +\infty)$ or $\min_x x^T Ax + b^T x + c$ for $x \in \mathcal{R}^p$.
Another easy optimization problem (logistic regression):

$$\max_{a,b} \prod_{i=1}^{n} \frac{e^{y_i(ax_i+b)}}{1 + e^{y_i(ax_i+b)}}$$

where $x_1, \ldots x_n$ are real numbers and $y_1, \ldots y_n \in \{-1, +1\}$.
A hard optimization problem (maxima of a kernel density estimate):

$$\max_x \frac{1}{n} \sum_{i=1}^{n} k(x_i, x)$$

where $k(x', x)$ is a positive symmetric function called the *kernel*.
5 Numerical analysis Algorithms for matrix computation. The behaviour of algorithms in the presence of rounding errors and how to make them stable.
Example: computing A^{-1} for a square matrix A. What happens when $\det A \to 0$?

This course:

- `python` programming
- algorithms and data structures
- applications in statistics (some possibilities)
  - simple: mean, median, covariance matrix, contingency tables
  - kernel density estimation
  - EM for mixtures
  - nearest neighbor and K-D trees
  - Hidden Markov Models
  - MCMC
  - ...