CSE 547 /
STAT 548

4/4/25

Lecture 2

- sit front next week
- Canvas up ← syllabus
- web
  - L I
  - L II clustering

Lecture I – Big Data in Machine Learning

Marina Meilă

Department of Statistics
University of Washington

STAT 548/CSE 547
Spring, 2025

Big data and Machine Learning I

Big Data has implications for ML at many levels

- Storage
  - may not fit in local memory ← distibuted algo ← bring algs/computation to data
  - expensive/slow to move around
  - I/O expensive/slow

preserve data privacy – FEDERATED learning

- Access
  - serial/by block, not random ←
- Indexing
  - Preprocessing steps that allow faster access during
- Computing
  - Parallelization when possible
  - Automation of resource management (Hadoop, Spark)
- Algorithms
  - predominantly sub-quadratic, i.e. $\mathcal{O}(n), \tilde{\mathcal{O}}(n)$
  - sub-linear, i.e. $o(n)$ when possible – sampling, Stochastic Gradient Descent (SGD)

$n = 10^6$

$\binom{n}{2} = 10^{12} \cdot \frac{1}{2}$

Ex: find all x_i, x_j so that $\|x_i - x_j\| < \varepsilon$ ← $O(n^2)$

$i,j = 1:n$

Finding near neighbors

run time is $O(n)$ "grows like" n

$= r(n) = c_0 + c_1 \cdot n$

$\frac{r(n)}{n} \le$ constant $\leftarrow$ Ex: what is constant?

$\left[\frac{r(n)}{n} \longrightarrow \text{constant} < \infty \right]$
n large

$\tilde{O}(n)$

$r(n) = c \cdot n \log(n)$ (sorting) $\Leftrightarrow$ $r(n) \sim \tilde{O}(n)$

$\frac{r(n)}{n \cdot \text{poly}(\log(n))} \le$ const

Fact $\frac{\log^p(n)}{n} \longrightarrow 0$ for $n \to \infty$

$p = 1, 2, 3,$

$o(n)$ $r(n) = c \cdot \sqrt{n} \Rightarrow \frac{\sqrt{n}}{n} \longrightarrow 0$ $r(n) = o(n)$

neglij

$O(n^2) = \binom{n}{2} = \#\{\text{pairs } (i,j), i,j = 1:n, i<j\}$

Big data and Machine Learning II

$n \to \infty$

finite storage m

- Tasks
  - streaming
  - bandits ← model for advertising
  - on-line learning ← target of learning changes with time
  - approximate rather than exact solutions (e.g. nearest-neighbors)
- Statistical
  - new problems (streaming, bandits)
  - what is i.i.d. sampling anyways? (on-line learning)
  - approximation and sampling (e.g. how to sample from a data stream)
  - can ask more detailed questions – non-parametric statistics
  - more spurious patterns to find – validation without human intervention
  - often high dimension D

    D large $\Rightarrow$ NEED for large n
  - Solves curse of dimensionality? No.

$\boxed{\begin{matrix} w \\ n \end{matrix}}$

$D = \#\text{genes} \gg n = \#\text{patients}$

$y = \beta^T X + \varepsilon$

$X \in \mathbb{R}^{D \times n}$

$y \in \mathbb{R}^n$

$s = 5$ sparsity

$\beta_j = 0$ for all $j \notin S$

$|S| = s$

$d \cong s \log_2 D \ll n$

$\uparrow$

\# params

Parametric vs. non-parametric

$\Sigma \in \mathbb{R}^{D \times D} \rightarrow \dim = D + \frac{D(D+1)}{2}$

- $N(\mu, \Sigma)$
- Regression

$y = \beta^T x + \varepsilon \rightarrow \dim \beta = D$ indep. of n

A mathematical definition

- A model class $\mathcal{F}$ is **parametric** if it is finite-dimensional, otherwise it is **non-parametric**

In other words

- When we estimate a parametric model from data, there is a fixed number of parameters, (you can think of them as one for each dimension, although this is not always true), that we need to estimate to obtain an estimate $\hat{f} \in \mathcal{F}$.
- The parameters are meaningful.
  E.g. the β_j in logistic regression has a precise meaning: the component of the normal to the decision boundary along coordinate j.
- The dimension of β does not change if the sample size n increases.

$\mathrm{Var}(\hat{\beta}) \sim \frac{1}{n}$

Non-parametric models – Some intuition

- When the model is non-parametric, the model class $\mathcal{F}$ is a function space .
- The $\hat{f}$ that we estimate will depend on some numerical values (and we could call them parameters), but these values have little meaning taken individually .
- The number of values needed to describe $\hat{f}$ generally grows with n.
  Examples In the Nearest neighbor and kernel predictors, we have to store a ll the data points, thus the number of values describing the predictor f grows (linearly) with the sample size.
  Exercise Does the number of values describing f always grow linearly with the sample size? Does it have to always grow to infinity? Doe s it have to always grow in the same way for a given $\mathcal{F}$?
- Non-parametric models often have a **smoothness parameter**.
  Examples of smoothness parameters K in K-nearest neighbor, h the kernel bandwidth in kernel regression.
  To make matters worse, a smoothness parameter is not a parameter! More p recisely it is not a parameter of an $f \in \mathcal{F}$, because it is not estimated from the data, but a descriptor of the model class $\mathcal{F}$.
- We will return to smoothness parameters later in this lecture.

Parametric vs. non-parametric models

Infinite dim

Finite dimension

Parametric

- Linear, logistic regression
- Linear Discriminant Analysis (LDA)
- Neural networks (if not very large)

$n > p = \#$ params

- Naive Bayes
- CART with L levels

Decision trees

- Clustering by k-means, finite mixture models
- Spectral clustering of graphs

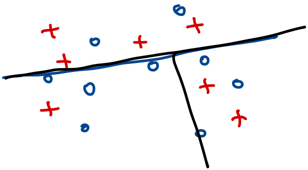
Non-parametric

- Nearest-neighbor classifiers and regressors
- Kernel (e.g. Nataraya-Watson) regression
- Monotonic regression
- Support Vector Machines
- Large, overparametrized neural networks

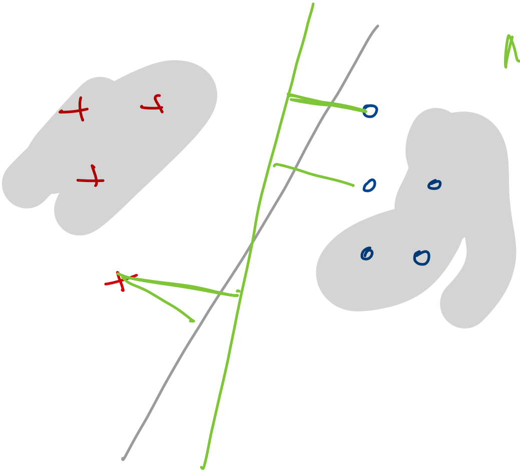
$p >> n$

- Gaussian processes
- Dirichlet Process Mixture Models (DPMM)
- Clustering by level sets or mode-finding
- Affinity based clustering
- Manifold learning

- L NOT fixed



DT can perfectly classify for L large enough



Max Margin Classifier

Hyperplane

This course

High-dimensional data	Graph data	Machine Learning	Infinite data
Nearest-neighbor finding	Network analysis	Clustering	Data streams
Dimension reduction	Graph clustering	Kernel and k-nearest neighbor regression and classification	Apps
	Models for networks		Parallel processing
		SVM	[Recommender systems]
		Stochastic gradient descent	

Leetcode

- Similarities btw code problems

parse-
• trees of code → abstract tree → vectorize

preorder traversal