CS5339 Lecture Notes #0: Introduction

Jonathan Scarlett

March 30, 2021

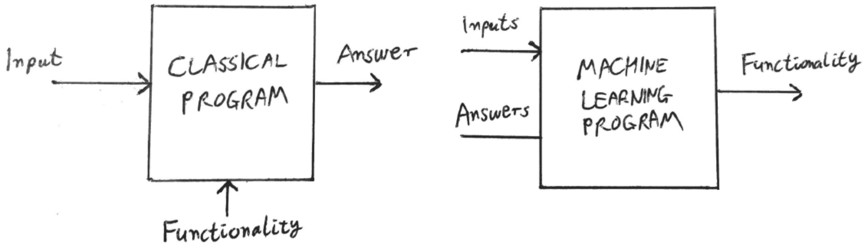
Useful references:

- Supplementary notes lec1a.pdf
- Chapters 1 and 2 of Bishop's "Pattern Recognition and Machine Learning" book
- Chapters 1 and 2 of "Understanding Machine Learning" book[1] [2]
- Lecture notes from MIT[3] or UIUC[4] – in the later reference lists, only the former will be included.

1 Introduction

Problems.

- Broad goal of machine learning: Use data to learn some "functionality" from data, rather than the "classical" approach of explicitly programming that functionality:



- Throughout the course, a "data set" is a collection of pairs $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, $\dots$, $(\mathbf{x}_n, y_n)$, with n in total. In more concise notation, $\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^n$.
  - $\mathbf{x}_t \in \mathbb{R}^d$ has various names: Input vector, covariates, feature vector, independent variables, etc.
  - $y_t \in \mathbb{R}$ also has various names: Output variable, target, label, dependent variable, etc.

  In some problems, the labels y_t are absent (such problems are called "unsupervised learning" problems).

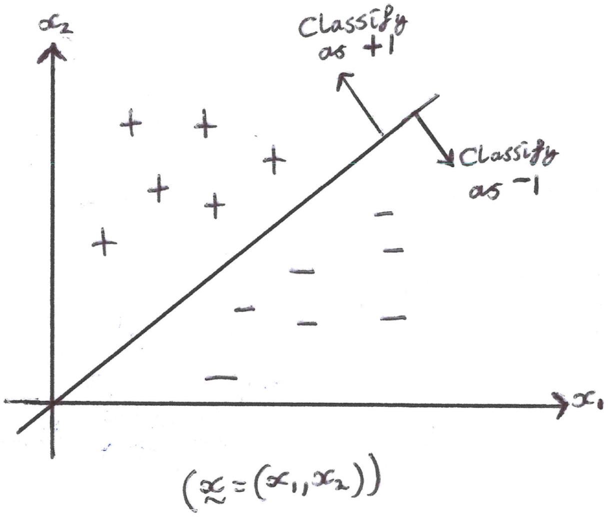
[1] http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf

[2] Bishop's book is the more standard choice for those new to the field, whereas this book (by Shalev-Shwartz and Ben-David) is more on the mathematical side, which may be preferred by some. A caveat to choosing this one is that the presentation is "backwards" compared to the course – it first covers statistical learning theory, and then algorithms.

[3] http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/lecture-notes/

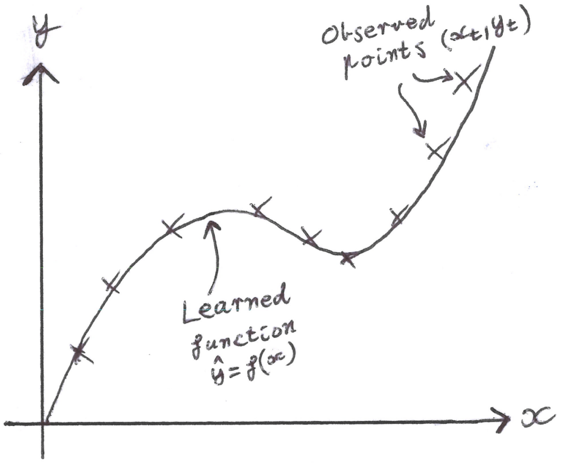
[4] http://mjt.cs.illinois.edu/courses/ml-s19/

- Classification: Given a data set $\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^n$ with $y_t \in \{+1, -1\}$, learn a "good" mapping $\hat{y} = f(\mathbf{x})$:



  - e.g., $y = 1$ if $\mathbf{x}$ is a spam email
  - e.g., $y = 1$ if $\mathbf{x}$ is my fingerprint
  - e.g., $y = 1$ if customer with details $\mathbf{x}$ will like the product
  - e.g., $y = 1$ if patient with details $\mathbf{x}$ has a disease
  - ...

- Regression: Given a data set $\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^n$ with $y_t \in \mathbb{R}$, try to learn a "good" mapping $\hat{y} = f(\mathbf{x})$:

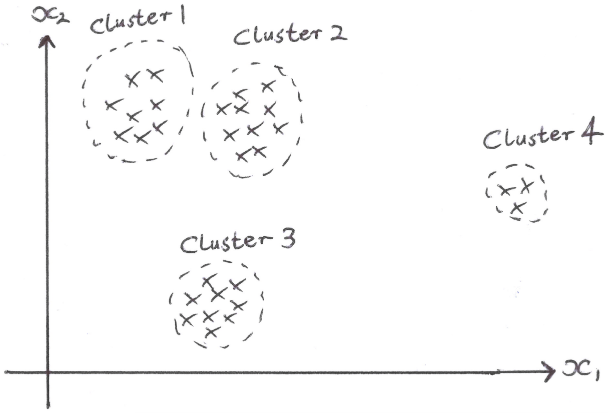


y
Observed points (x_t, y_t)
Learned function $\hat{y} = f(x)$
x

  - e.g., y is the next stock price, $\mathbf{x}$ contains a number of previous prices
  - e.g., y is the concentration of oil at location $\mathbf{x}$
  - e.g., y is the number of sales of a product represented by $\mathbf{x}$

- e.g., y measures the effectiveness of a medicine represented by $\mathbf{x}$
- ...

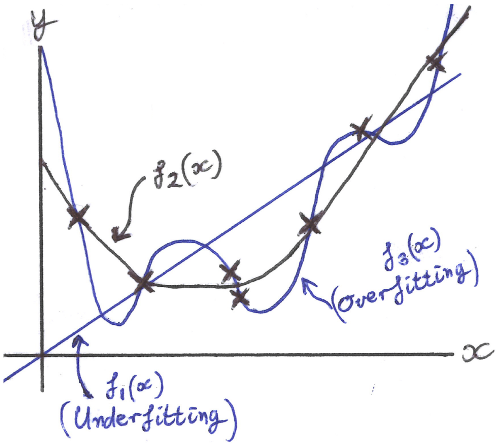
- Clustering: Given an *unlabeled* data set $\mathcal{D} = \{\mathbf{x}_t\}_{t=1}^n$, group the inputs into "similar" clusters:



- e.g., groups of "similar" users in a social network
- e.g., groups of "similar" documents such as news reports
- ...

Challenges.

- Over-fitting and under-fitting. Richer models are not always a good idea:



- Choosing a good model ...or even class of models
  - e.g., fit a polynomial of degree 2? 5? 100? Or fit something other than a polynomial?
- Data representation.
  - e.g., represent email with $\mathbf{x}$ being purely a string of characters
  - e.g., represent email with $\mathbf{x}$ having j-th entry x_j = (#times the t-th word occurs)

- Computational complexity.
  - The "best" methods often require the most computation (often too much to be feasible)
  - Particularly relevant for large data sets (even performing [constant $\times n \log n$] computations might be infeasible)
- Curse of dimensionality.
  - Recall that the input $\mathbf{x}$ is in $\mathbb{R}^d$. Often d can be very large (e.g., number of English words).
  - *Learning issues*: Even if $\mathbf{x} \in \{-1,1\}^d$ and $y \in \{-1,1\}$, there are 2^{2^d} different functions mapping $\mathbf{x}$ to y. One needs to assume/exploit structure that ensures a lower "effective dimensionality".
  - *Computational issues*: Even performing [constant $\times\, d \log d$] computations might be infeasible
  - *Geometric issues*: Our intuition of 2D/3D geometry falls apart in high dimensions

Goals of this course.

- Students are expected to learn modern machine learning methods covering the above problems, understand how/why/when they work, and be able to derive/explain the underlying mathematical theory.
- The course will **not** introduce you to state-of-the-art algorithms or cutting-edge applications; instead, we focus on some of the fundamental concepts and ideas that are required to properly understand those (and to potentially build on them and improve them!).

Topics not covered.

- Machine learning is such a huge field that the breadth and depth of a single course must be limited.
- Classical/fundamental topics not covered (unless maybe we have spare time):
  - Decision trees (+ random forests)
  - Active learning
  - Graphical models
  - Approximate Bayesian methods (e.g., naive Bayes, Monte Carlo, variational methods)
  - Reinforcement learning
  - PCA and other dimensionality reduction methods
  - ...
- Advanced topics not covered:
  - Deep learning
  - Generative models
  - Convergence analysis for optimization
  - Semi-supervised learning
  - Fairness/interpretability
  - Privacy/security

- Bandit and ranking problems
- Gaussian process methods[5]
- Dataset Shift
- Meta-learning (learning to learn)
- Matrix factorization
- Distributed learning algorithms
- Information-theoretic limits[6]
- ...

2 Background Material

I suggest brushing up on the following if any of it is unfamiliar. If you are not sure where to find suitable reading material, please feel free to ask me. Some of it is covered in the optional tutorial 00-Intro_Tutorial.pdf (available on LumiNUS).

Probability.

- Basic operations (conditioning, marginal distribution, averaging)
- Probability mass function, probability density function
- Independence, conditional independence
- Covariance matrix, law of total variance
- Law of large numbers, central limit theorem
- Bayes' rule

Linear Algebra.

- Basic operations (matrix multiplication, transpose, inverse, inner product, norm)
- Positive (semi)definite matrices
- Eigenvalues and eigenvectors
- Gradient $\nabla f(\mathbf{x})$ and Hessian $\nabla^2 f(\mathbf{x})$ of a multivariate function
- Some basic matrix calculus (don't worry if you don't know it)

Others.

- Calculus (basic derivatives and integrals)
- Basic limits (e.g. $\left(1+\frac{c}{n}\right)^n \to e^c$)
- Big-O notation[7]
- Taylor expansions

[5] I have some slides on this topic available at `https://www.comp.nus.edu.sg/~scarlett/gp_slides/` if you are interested.

[6] Again, see `https://www.comp.nus.edu.sg/~scarlett/it_data_slides/` for some slides.

[7] See the Wikipedia page `https://en.wikipedia.org/wiki/Big_O_notation` or the blog post `https://jeremykun.com/2011/06/14/big-o-notation-a-primer/`