Neural Networks: A Theory Lab

Goal: Introducing the notion of universality

Outline:

- Intro
- Lab
- Theory

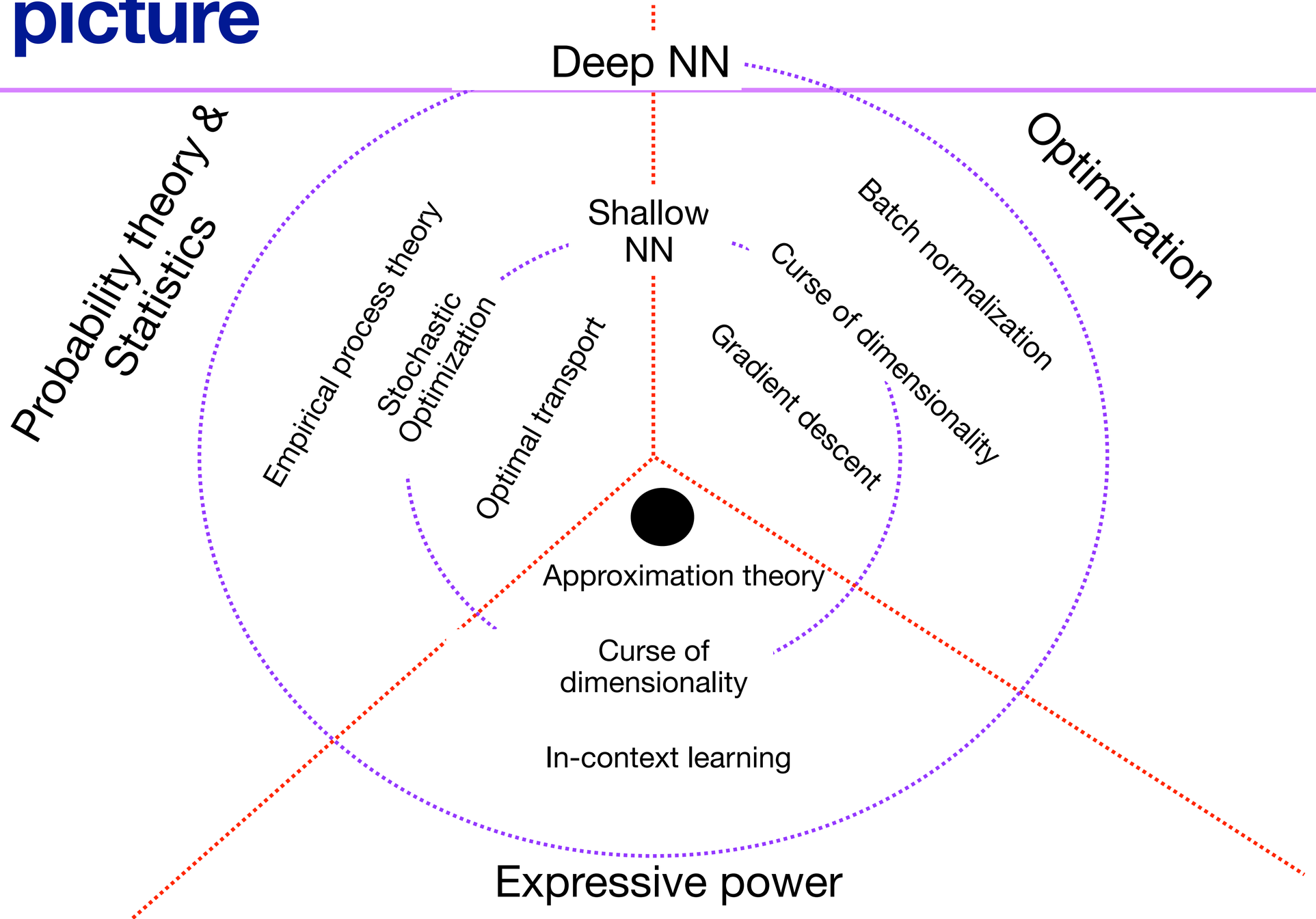
Hadi Daneshmand

University of Virginia

Before starting the lecture

- Canvas is published
  - You can submit assignments and project reports in Canvas.
- I will record lectures from today on and post it on Canvas
- Please interrupt me for **questions**

Big picture


Deep NN
Probability theory & Statistics
Optimization
Shallow NN
Empirical process theory
Stochastic Optimization
Optimal transport
Gradient descent
Curse of dimensionality
Batch normalization
Approximation theory
Curse of dimensionality
In-context learning
Expressive power

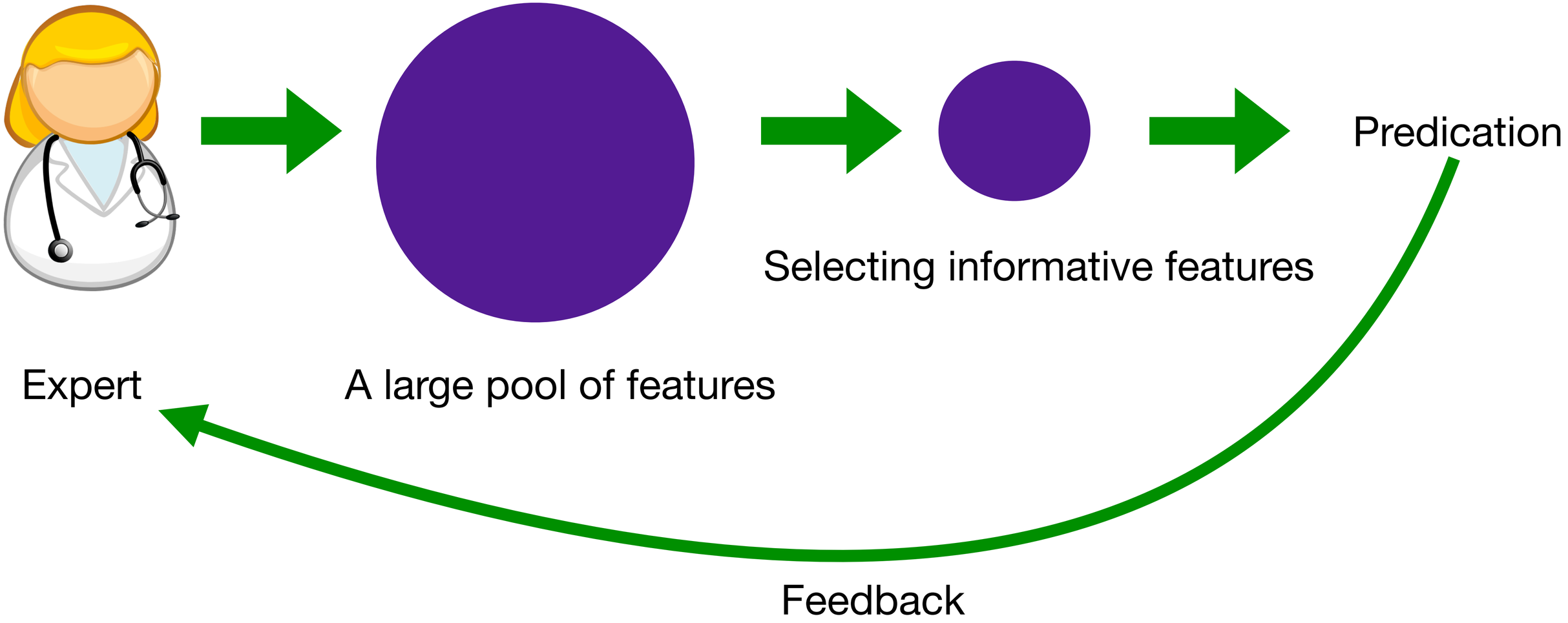
Why neural networks?

- Neural networks are designed to replace feature engineering
- What is feature engineering?
- We explain feature engineering via an example

Heart disease predication

- Problem: we want to predict whether a person in future have a heart disease
- Question: what information about the person will help for the predication?
- Ideas?
- Can I use the above information to predict another disease?

Conventional feature engineering

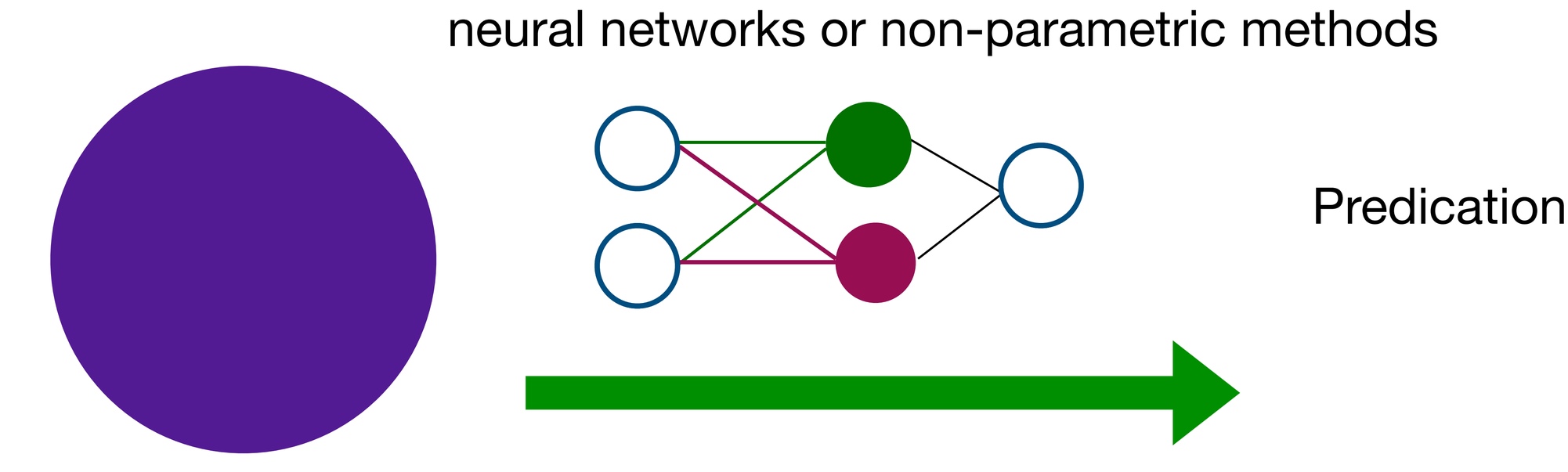


Challenges of feature engineering

- Labor-intensive
- Requires domain knowledge
- It is task dependent

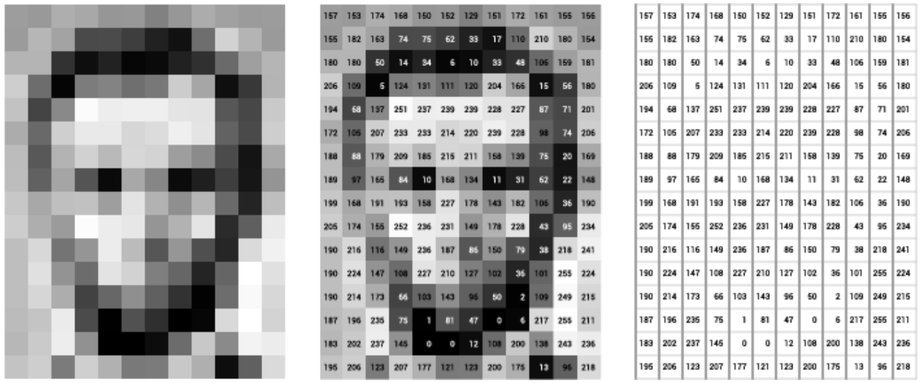
Deep Learning Goal

- Deep learning automates feature extraction (representation learning)



Abstraction of predication

- Image processing example



https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html

A face

Large Pool of Features: x

Output: $y = f(x)$

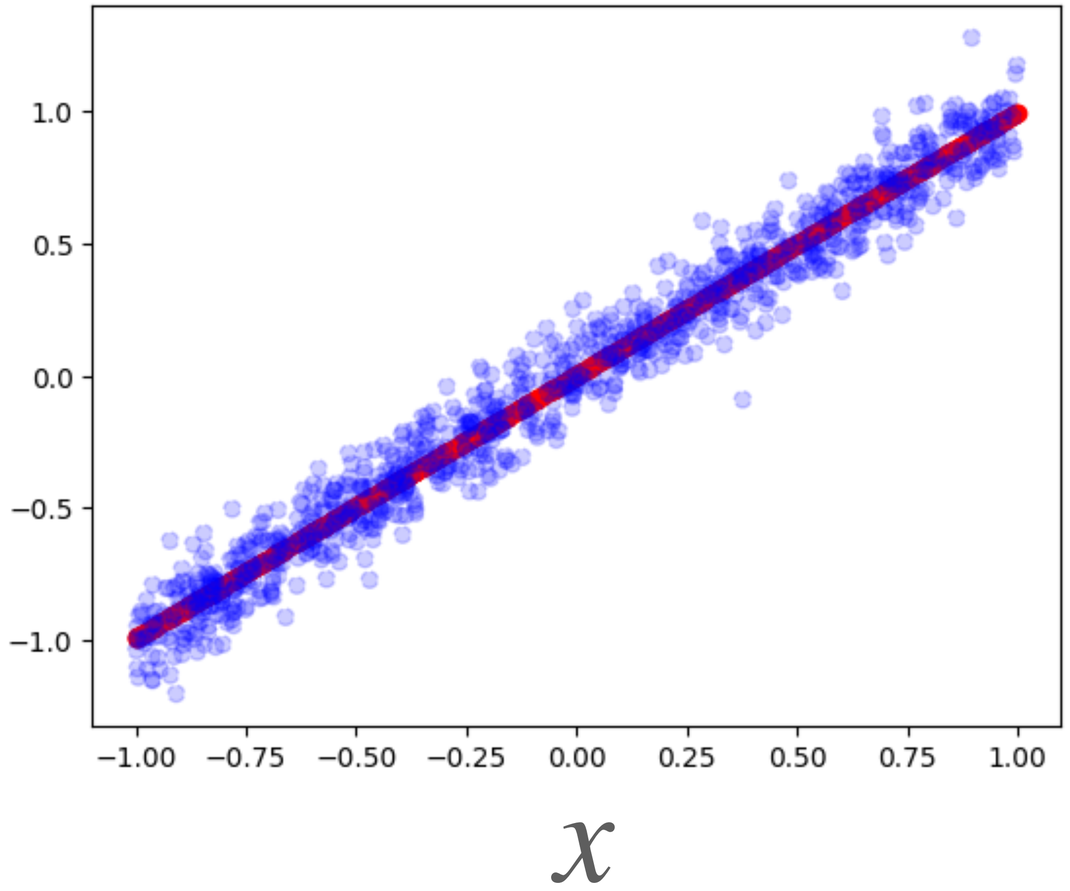
Regression problem

- Given $(x_1, y_1), \ldots, (x_n, y_n) \implies (x_{n+1}, ?)$
- Where $y_i = f(x_i)$ where function f is unknown

Recall linear regression

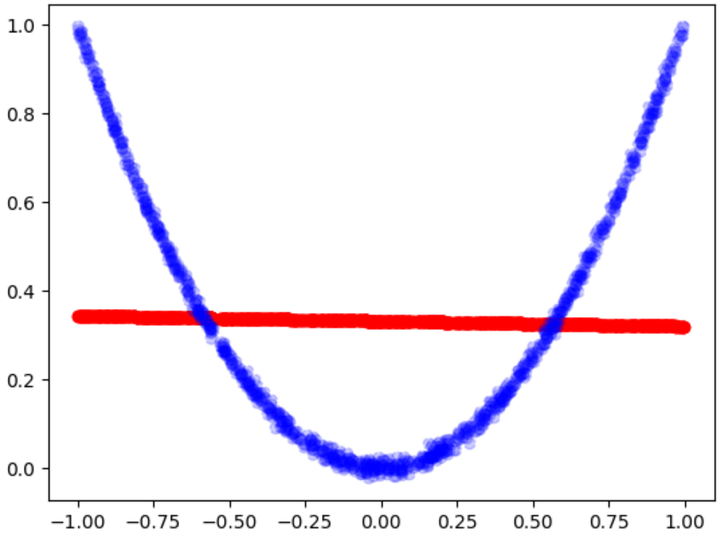
- Recall the example of linear regression that you learned about in Intro to ML course

$$y = f(x)$$

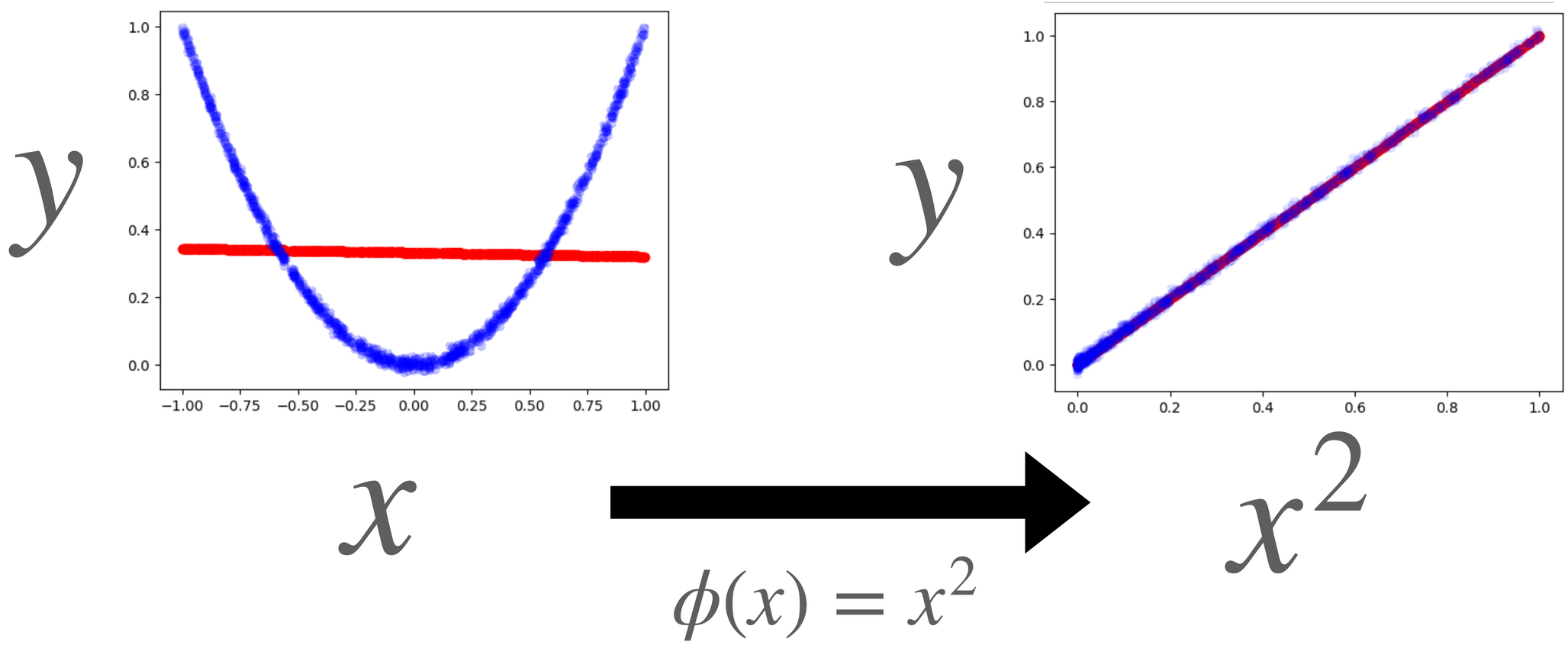


Recap: linear regression

- Given: $x_1, x_2, \ldots, x_n \in \mathbb{R}^d, y_1, y_2, \ldots, y_n \in \mathbb{R}$
- Find $w \in \mathbb{R}^d$ such that $\sum_{i=1}^{n} (x_i^\top w - y_i)^2$ is minimized.
- Challenge of linear regression: approximating non-linear functions



Question: How to solve the non-linearity challenge



A conventional method in statistics

- You won't believe: It is the standard approach in **statistics** to **manually** design a non-linear transformation of x

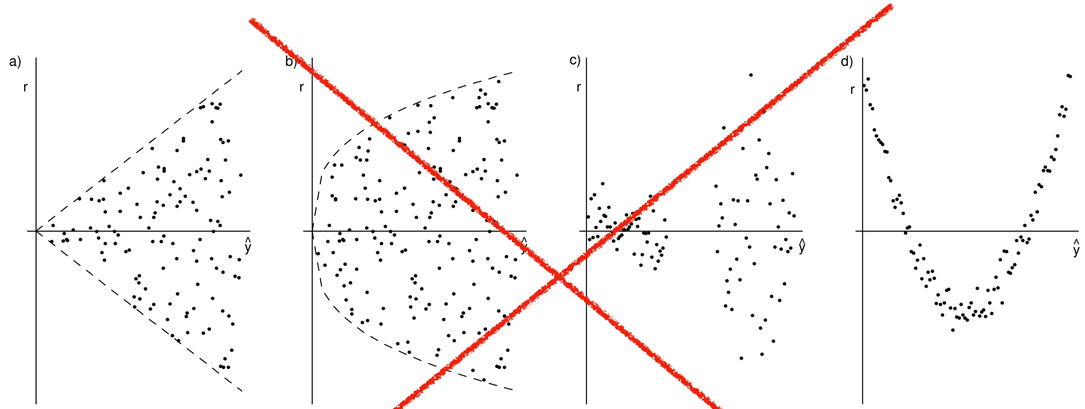


Figure 1.5: a) linear increase of standard deviation, b) nonlinear increase of standard deviation, c) 2 groups with different variances, d) missing quadratic term in the model.

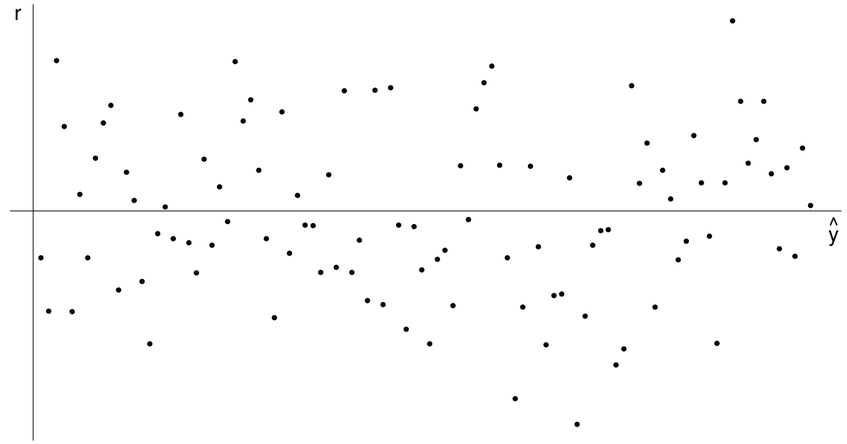
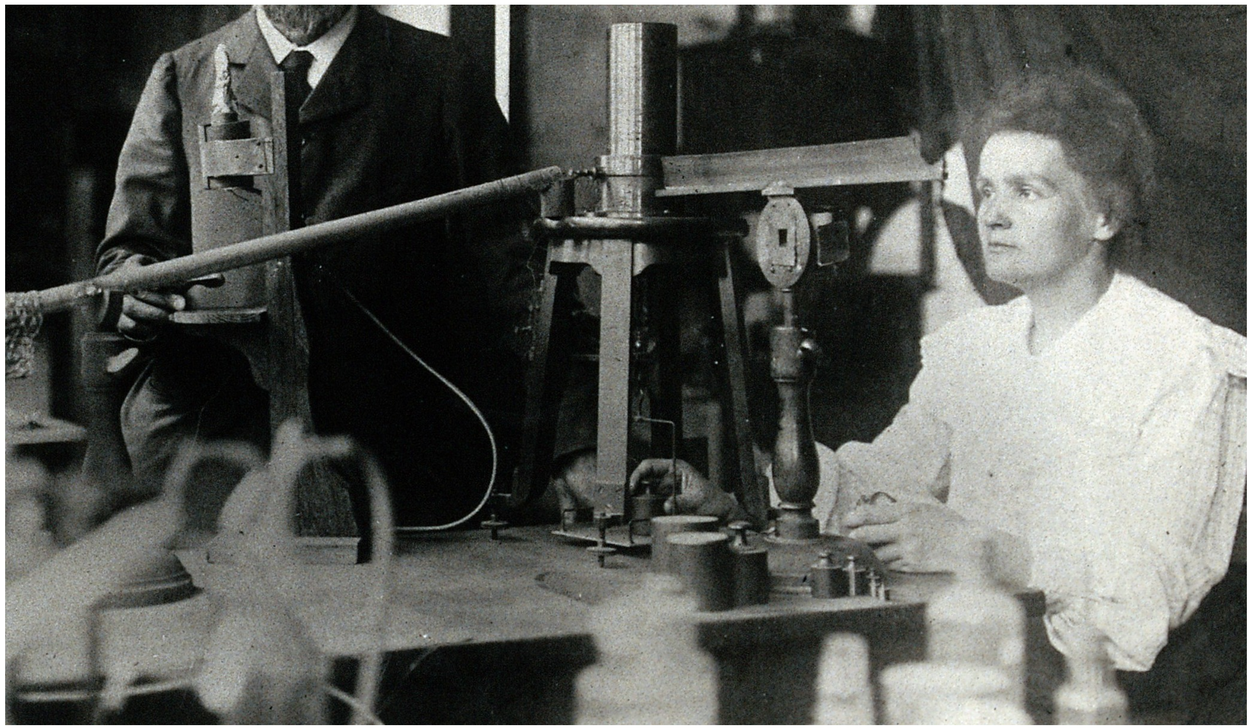


Figure 1.4: Ideal Tukey-Anscombe plot: no violations of model assumptions.

- Taken from my lecture notes in computational statistics course

- A brief intro
- Lab
- Theory



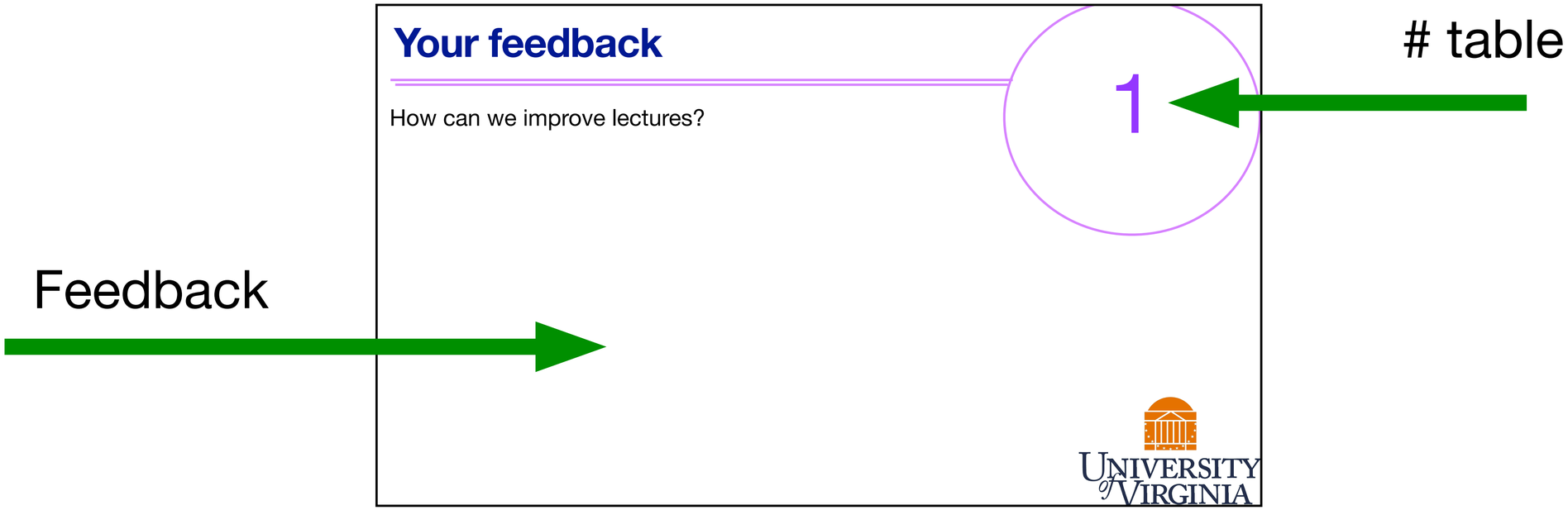
wikipedia: Marie Curie

Lab

Get ready for hands-on group activity

Group assignment

- Please join the table with the number on the paper

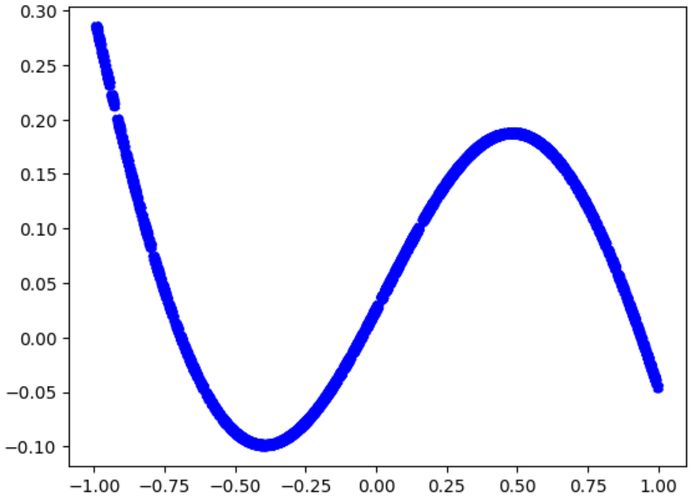


Work in groups

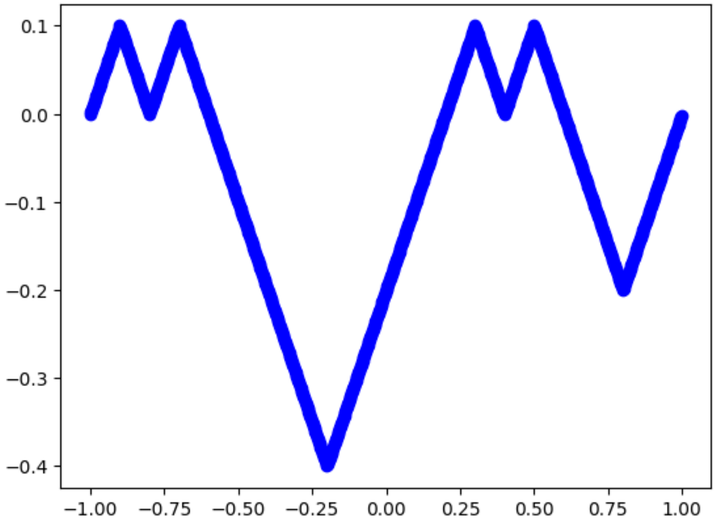
- Goal: We want to design **universal** non-linear features
- **Universal** features can approximate various y s

image: Flaticon.com'

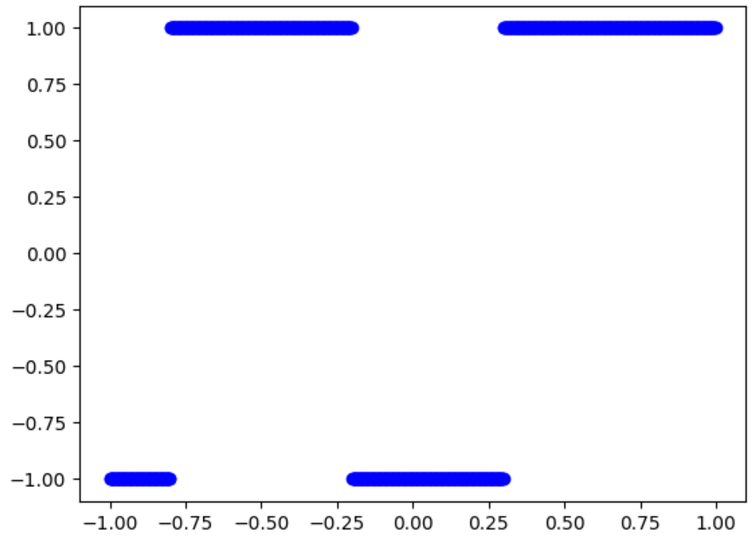
y



x



x



x

Your task (8 mins)

- Given x, design features $\phi_1(x), \ldots, \phi_n(x)$ such that $y \approx \sum_{i=1}^{n} w_i \phi_i(x)$
- $\phi_i : \mathbb{R} \to \mathbb{R}, x \in \mathbb{R}$

Scan to Start

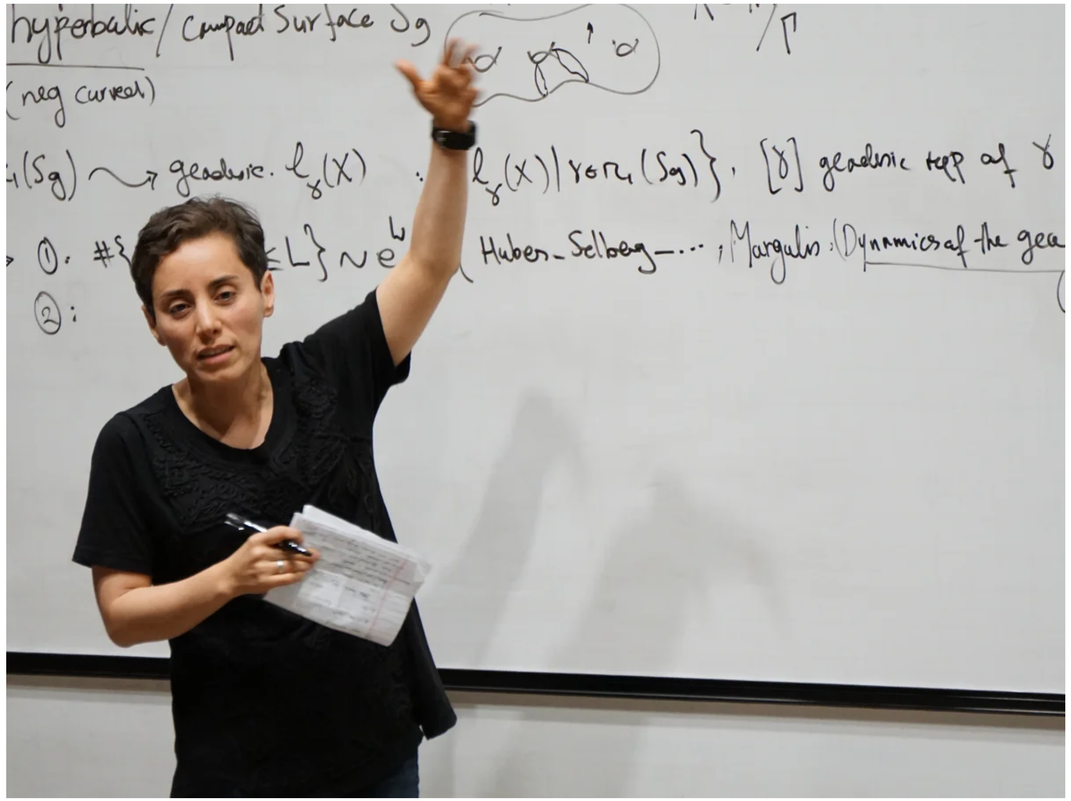
https://shorturl.at/ABb6H

https://colab.research.google.com/drive/1M-nRBdhg1XJiV8sy4wMkUsfPJLKKSCB0?usp=sharing

- You can search or use ChatGPT

Ideas?

- Intro
- Lab
- Theory



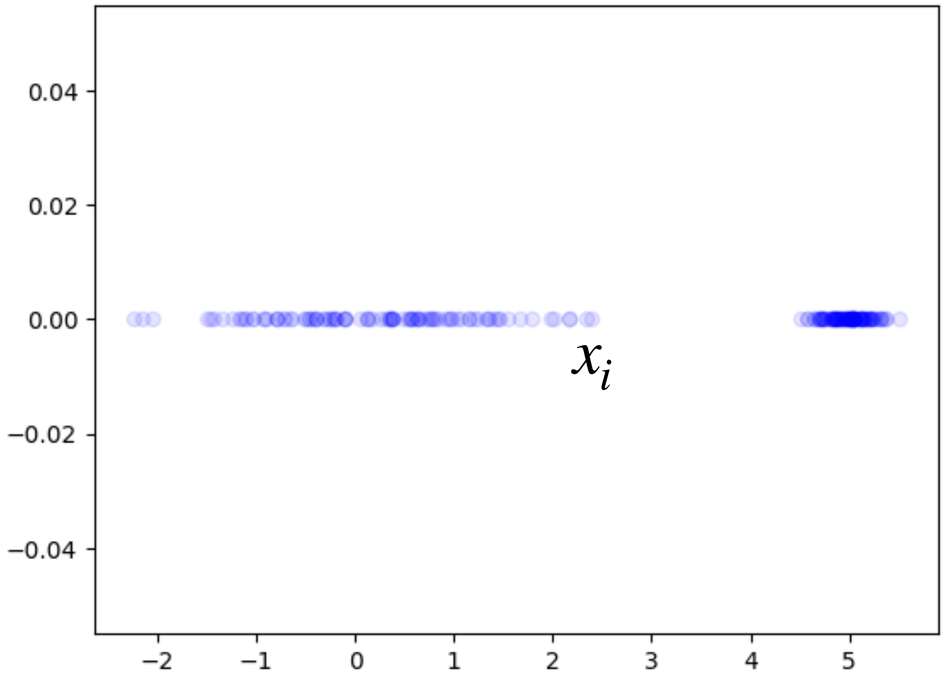
theguardian: Maryam Mirzakhani

Theory

Get ready for math

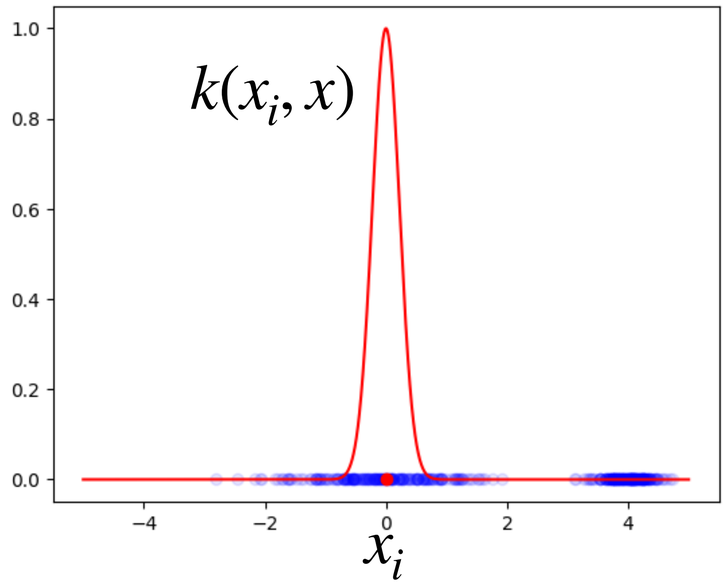
Density estimation problem

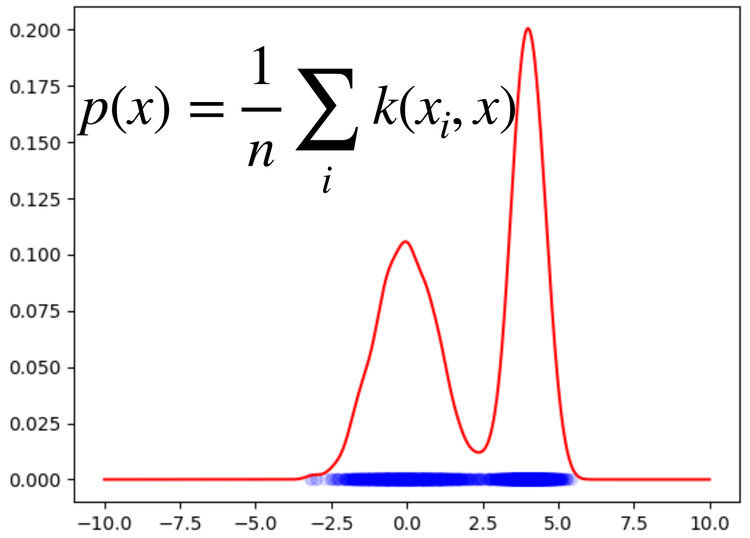
- Given i.i.d. samples $x_1, \ldots, x_n$, what is the probability density of the underlying distribution? (Pioneered by Tukey in 1977)



Kernel density estimation

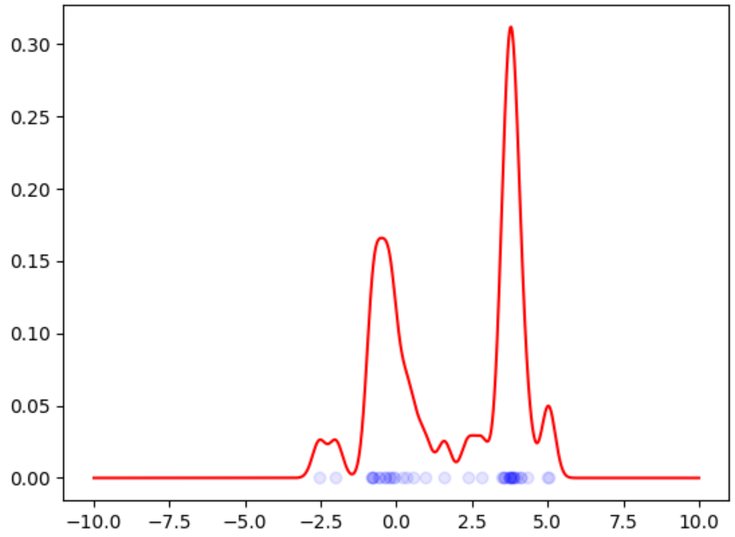
- Main idea is putting **bumps** on each point and compute the average
- **Bumps** are kernels $k(x, x_i) = \exp(-0.5\|x - x_i\|^2/\sigma^2)$



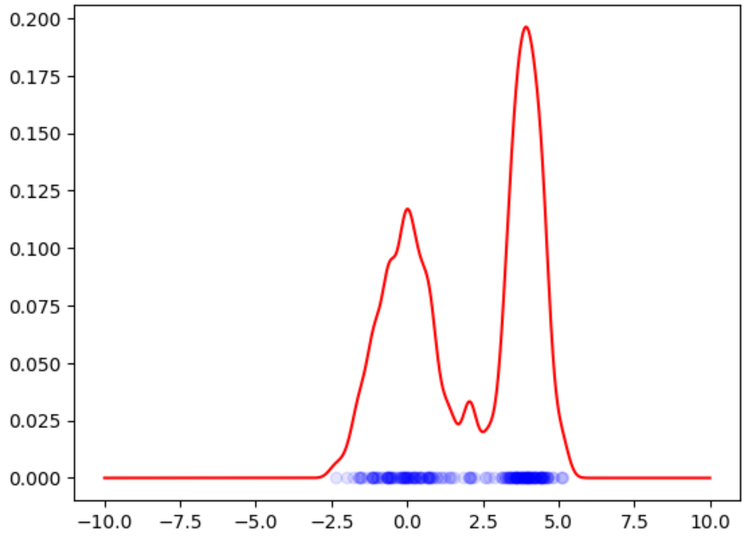


Asymptotic density estimation

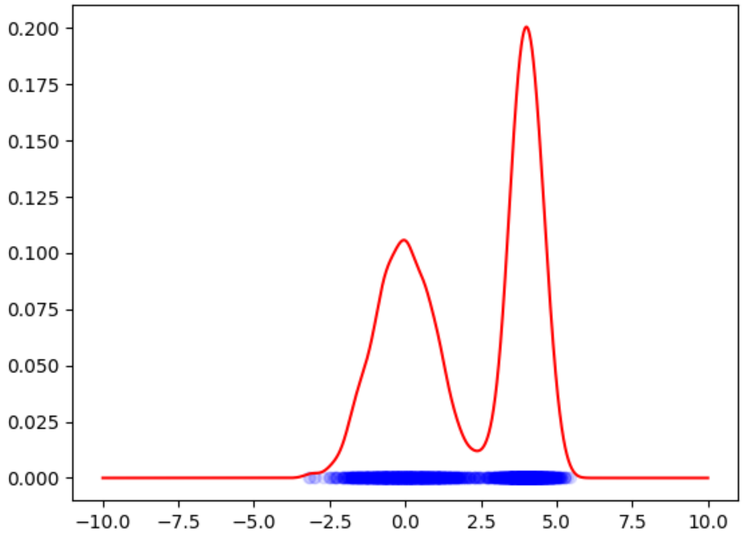
- For many distributions, the kernel density estimation converges to the true density as the number of samples goes to infinity



20 samples



100 samples

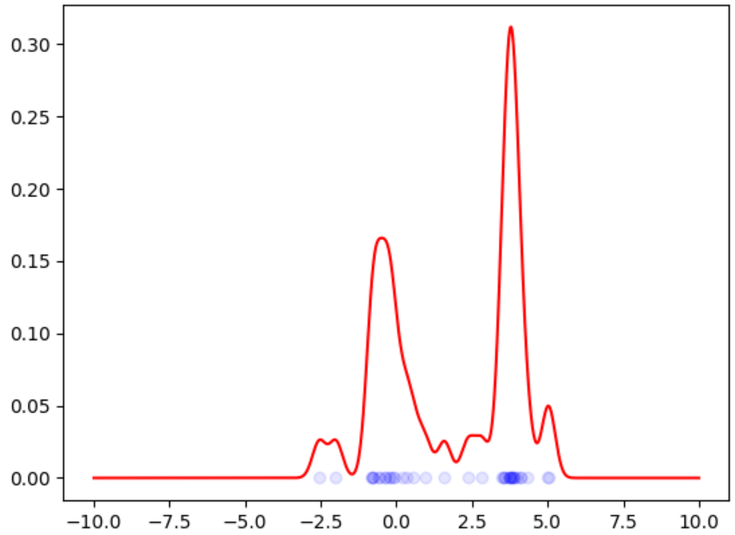


1000 samples

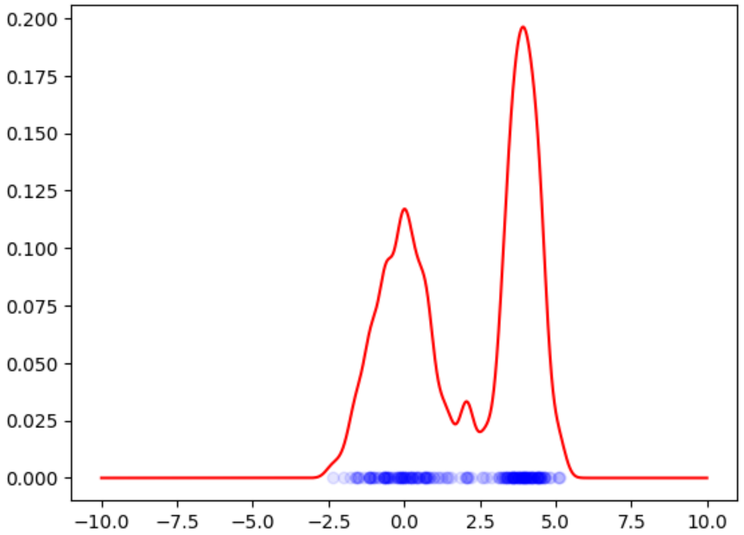
Group activity (3mins)

image: Flaticon.com'

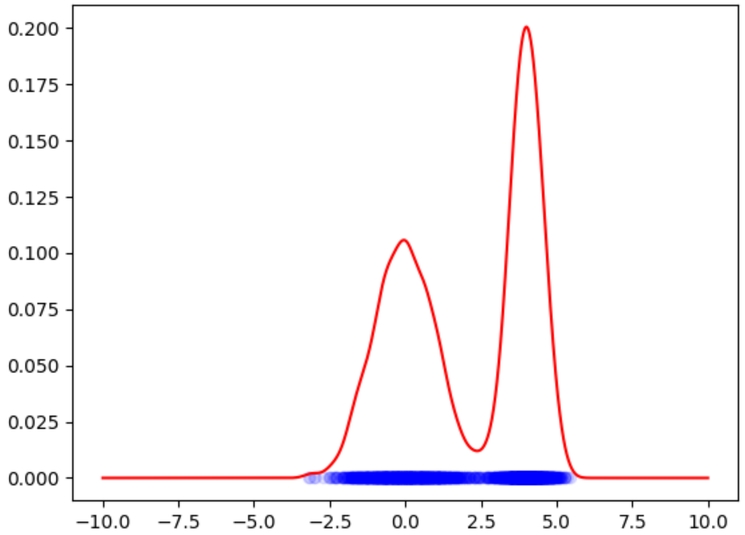
- When kernel density estimation become exact with an infinite sample size?



20 samples



100 samples

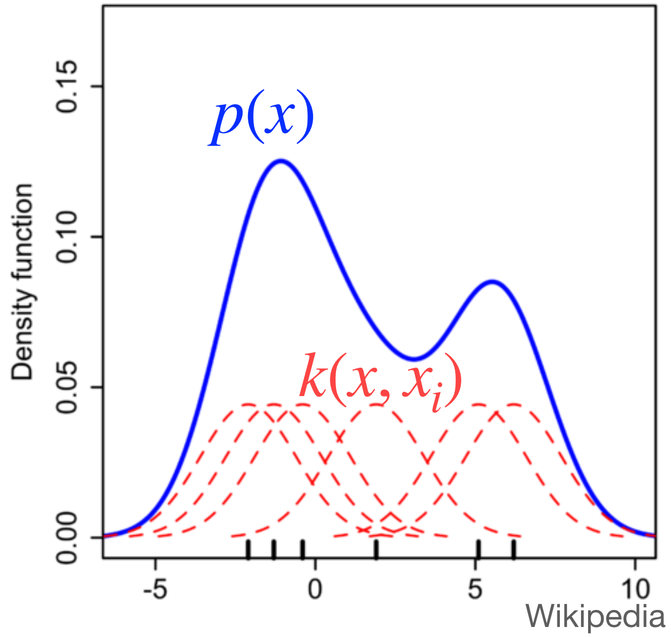


1000 samples

$$p(x) = \frac{1}{n} \sum_i k(x_i, x)$$

When does error vanishes with sample size?

- $p(x) = \lim_{n\to\infty, y_i \sim p(y)} \frac{1}{n} \sum_i k(y_i, x) = \int k(y, x) p(y) dy$: reproducing property
- Under the reproducing property, the error of kernel estimation vanishes as $n \to \infty$



Kernel regression and representer theorem

- Kernel regression: $f^* = \arg\min_f \sum_i (f(x_i) - \underbrace{y_i}_{p(x_i)})^2 + \lambda \|f\|^2$

  subject to f obeys the reproducing property $f(x) = \int_{-\infty}^{\infty} k(y, x) f(y) dy$

- **Representer theorem [Schölkopf et al. (2001)]:** $f^* = \sum_i \alpha_i k(x, x_i)$ for $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$

- As long as: k is positive semi-definite, namely $\sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \geq 0$ holds for all $x_i, x_j \in \mathbb{R}^d$, $\alpha_i \in \mathbb{R}$

Revisiting the task (7 mins)

- Given x, design features $\phi_1(x), \ldots, \phi_n(x)$ such that $y \approx \sum_{i=1}^{n} w_i \phi_i(x)$

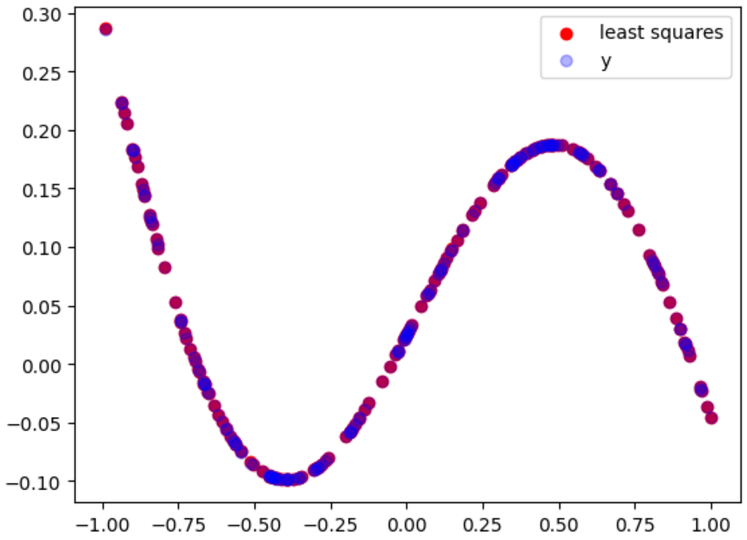
Scan to Start

https://shorturl.at/ABb6H

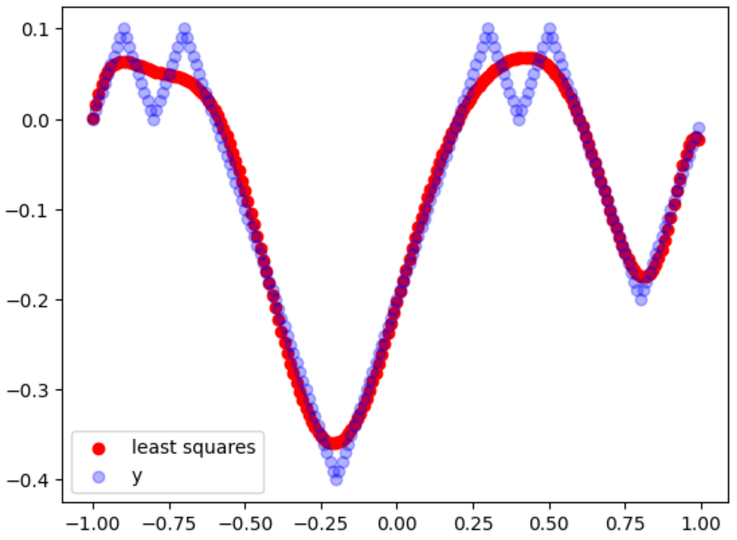
https://colab.research.google.com/drive/1M-nRBdhg1XJiV8sy4wMkUsfPJLKKSCB0?usp=sharing

$$\phi_i(x) = e^{-0.5\|x - x_i\|_2^2}$$

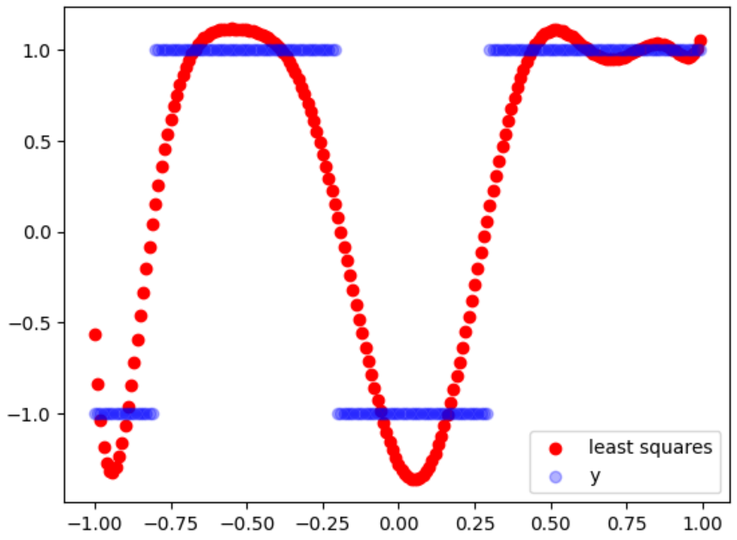
My results



Good approximation



Reasonable approximation



Poor approximation

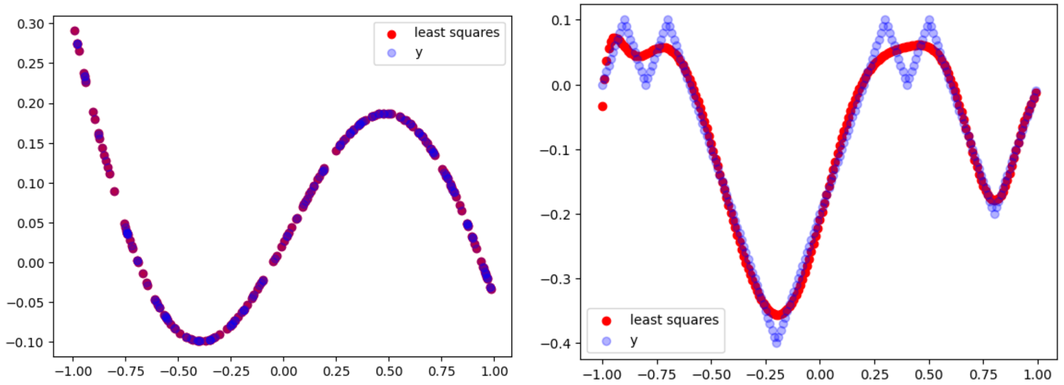
Summary

Theory

$$f^* = \sum_i \alpha_i k(x, x_i)$$

Representer theorem

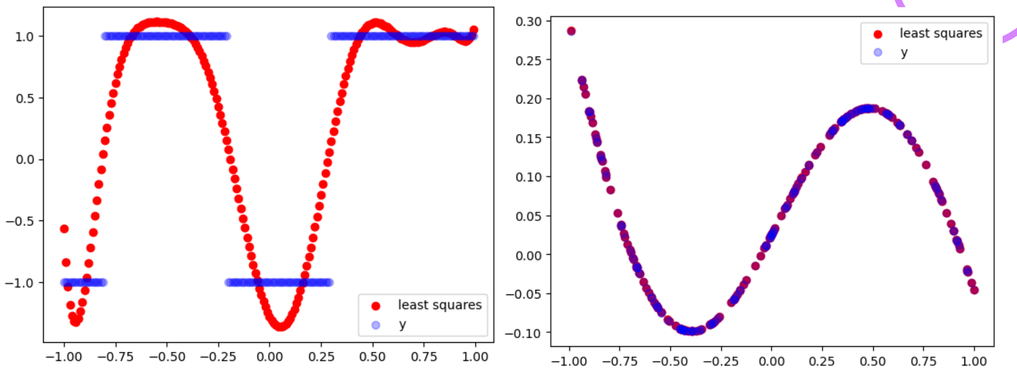
Observation



Kernel methods provide universal features

Group activity (2mins)

- For which functions, the estimation is good?



Recall

- Kernel regression: $f^* = \arg\min_f (f(x_i) - \underbrace{y_i}_{p(x_i)})^2 + \lambda \|f\|^2$

subject to f obeys the reproducing property $f(x) = \int_{-\infty}^{\infty} k(y,x)p(y)dy$

- Representer theorem: $f^* = \sum_i \alpha_i k(x, x_i)$ for $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$

Group activity (3mins)

- What is the accuracy of kernel methods on ImageNet dataset?

3 issue with kernels

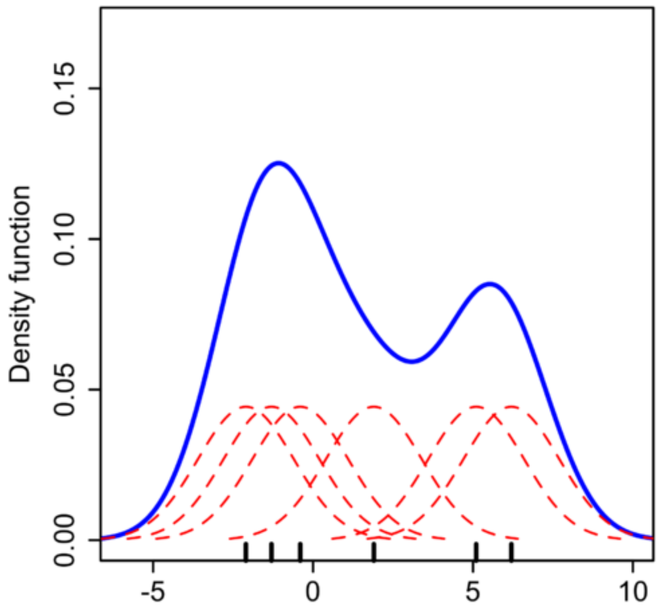
- Computational issue: Computational complexity of $O(\text{samplesize}^2)$
- Function approximation issue: **Curse of dimensionality**
- Choosing and designing proper kernels requires domain knowledge

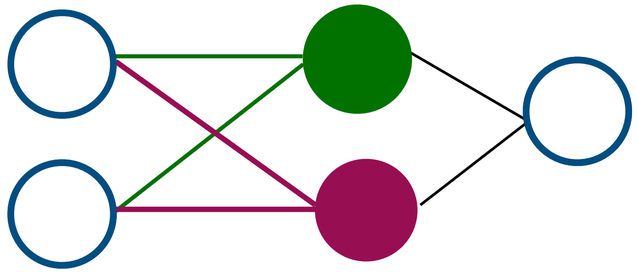
Machine learning universal tools

Kernel methods

Neural Nets

There is a bridge between these two





Thank you very much!