Optimization and Machine Learning SI151

Lu Sun

School of Information Science and Technology

ShanghaiTech University

March 2, 2020

Today:

- Introduction to machine learning and optimization
- Course logistics
- Overview of supervised learning I

Readings:

- The Element of Statistical Learning, Chapters 1 and 2
- Pattern Recognition and Machine Learning, Chapter 1

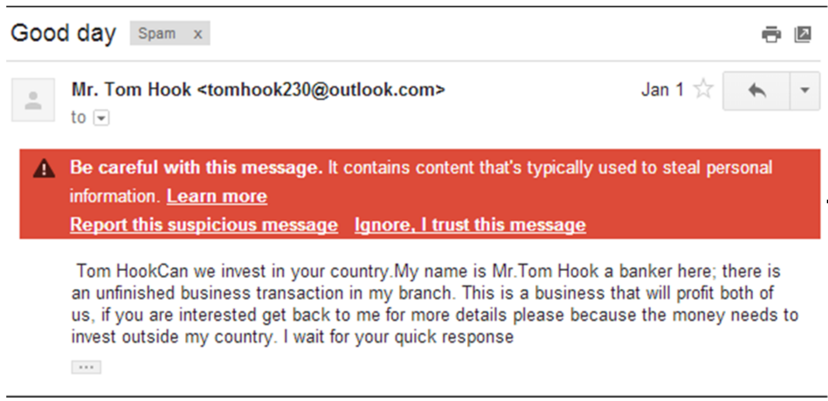
Introduction to Machine Learning and Optimization

Machine Learning

“Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.”

------Wikipedia

ML: Study of algorithms that
- improve their performance P
- at some task T
- with experience E



spam
vs
email

Well-defined ML task: <P, T, E>

Optimization in Machine Learning

Optimization problems arise in nearly everything we do in Machine Learning.
Typically, practical ML problems are solved mathematically by

translate

Conceptual idea

into

$$P: \min_{x \in D} f(X)$$

Optimization problem

Solution

This course: how to solve P.

Learning to Detect Spam Emails

- Data:
  - 4601 email messages
  - Each is labeled by email (+) or spam (-)
  - The relative frequencies of the 57 most commonly occurring words and punctuation marks in the message
- Classify:
  - label future messages email (+) or spam (-)
- Supervised learning problem on categorical data:

  Binary classification problem

Table: Words with largest difference between spam and email shown.

	spam	email
george	0.00	1.27
you	2.26	1.27
your	1.38	0.44
hp	0.02	0.90
free	0.52	0.07
hpl	0.01	0.43
!	0.51	0.11
our	0.51	0.18
re	0.13	0.42
edu	0.01	0.29
remove	0.28	0.01

Learning to Detect Spam Emails

- Examples of rules for prediction:
  - If (%george<0.6) and (%you>1.5)
    then `spam`
    else `email`
  - If (0.2·%you-0.3·%george)>0
    then `spam`
    else `email`
- Tolerance to errors:
  - Tolerant to letting through some spam
    (false positive)
  - No tolerance towards throwing out email
    (false negative)

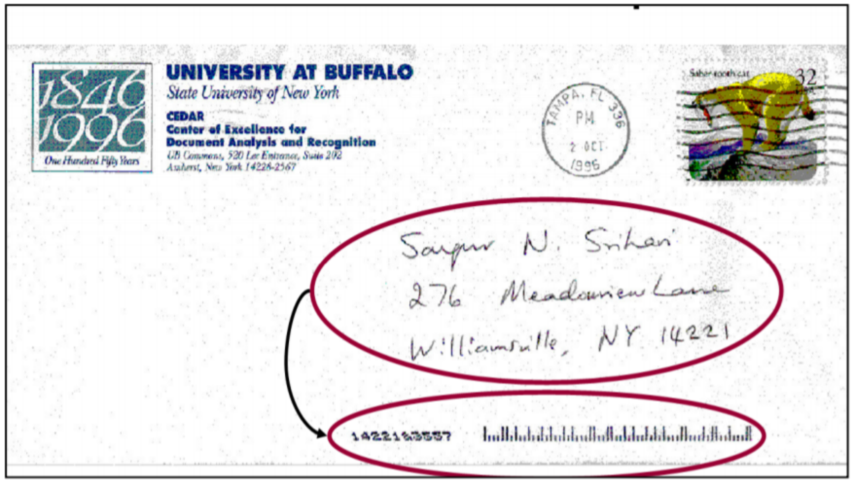
Table: Words with largest difference between spam and email shown.

	spam	email
george	0.00	1.27
you	2.26	1.27
your	1.38	0.44
hp	0.02	0.90
free	0.52	0.07
hpl	0.01	0.43
!	0.51	0.11
our	0.51	0.18
re	0.13	0.42
edu	0.01	0.29
remove	0.28	0.01

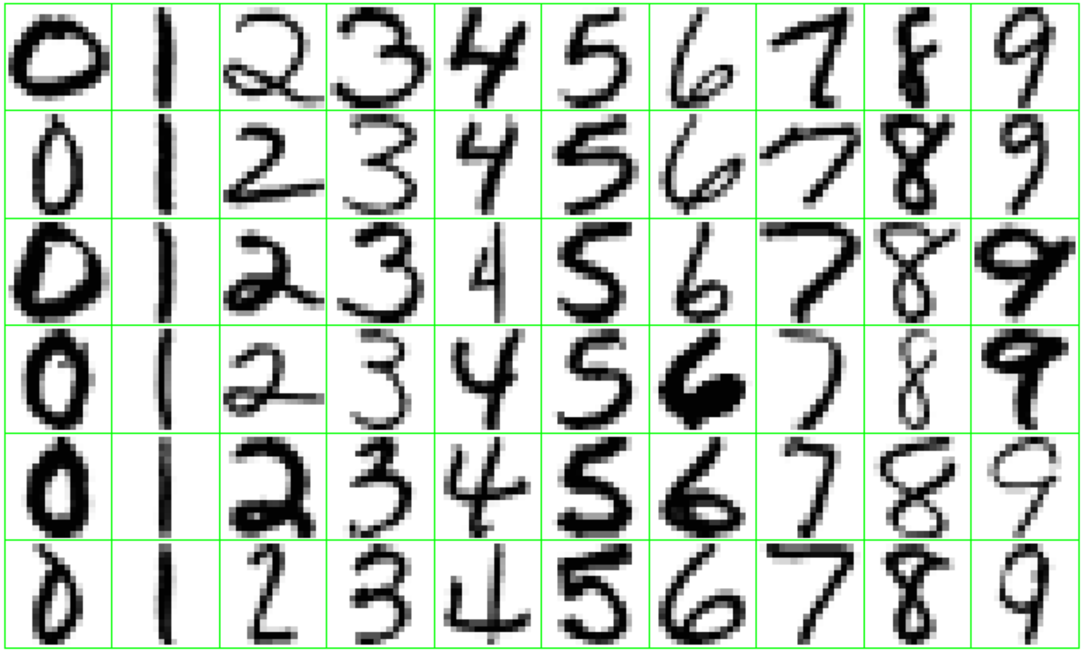
Learning to Recognize Handwritten Digits

Data: images are single digits 16x16 8-bit gray-scale, normalized for size and orientation

Classify: newly written digits



https://cedar.buffalo.edu/~srihari/CSE574/Chap1/1.1%20ML-Overview.pdf



- Non-binary classification problem
- Low tolerance to misclassifications

Learning to Diagnose Prostate Cancer

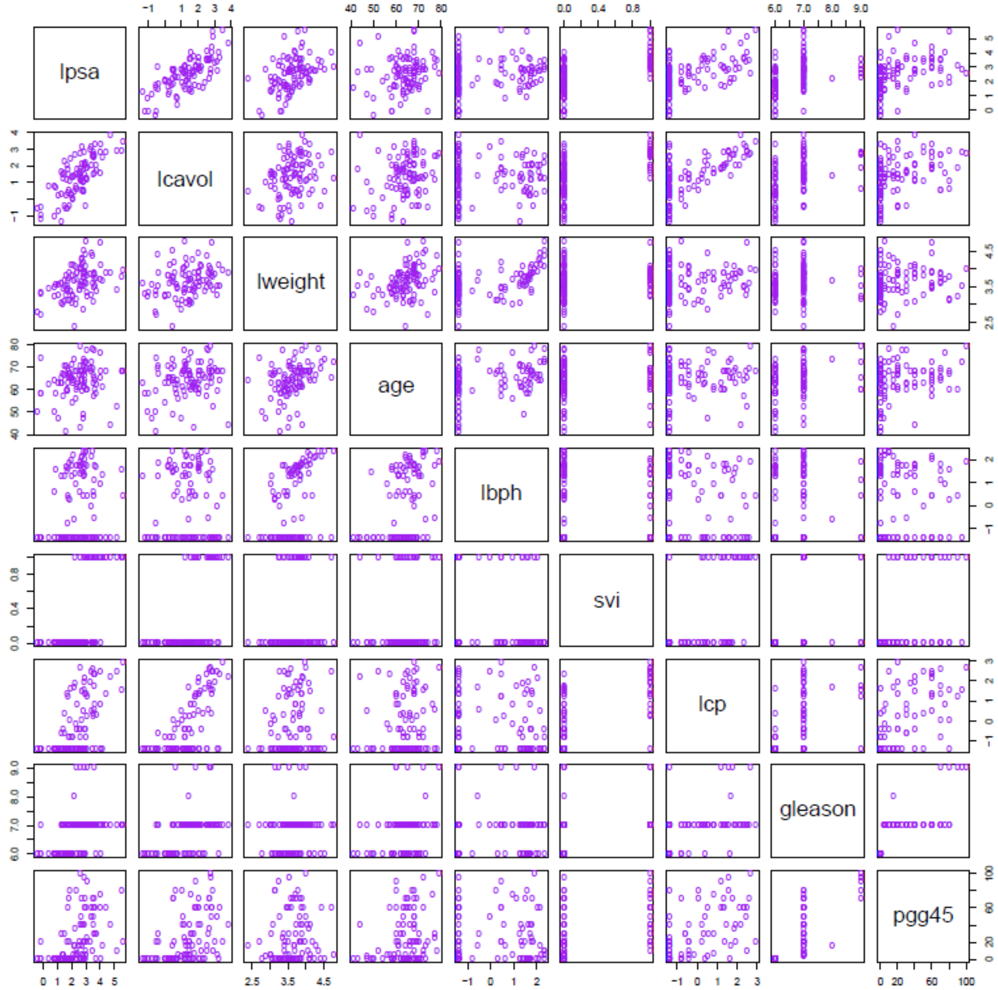
- Data (by Stamey et al. 1989):
  - Given:

`lcavol`	log cancer volume
`lweight`	log prostate weight
`age`	age
`lbph`	log benign hyperplasia amount
`svi`	seminal vesicle invasion
`lcp`	log capsular penetration
`gleason`	gleason score
`pgg45`	percent gleason scores 4 or 5

  - Predict:

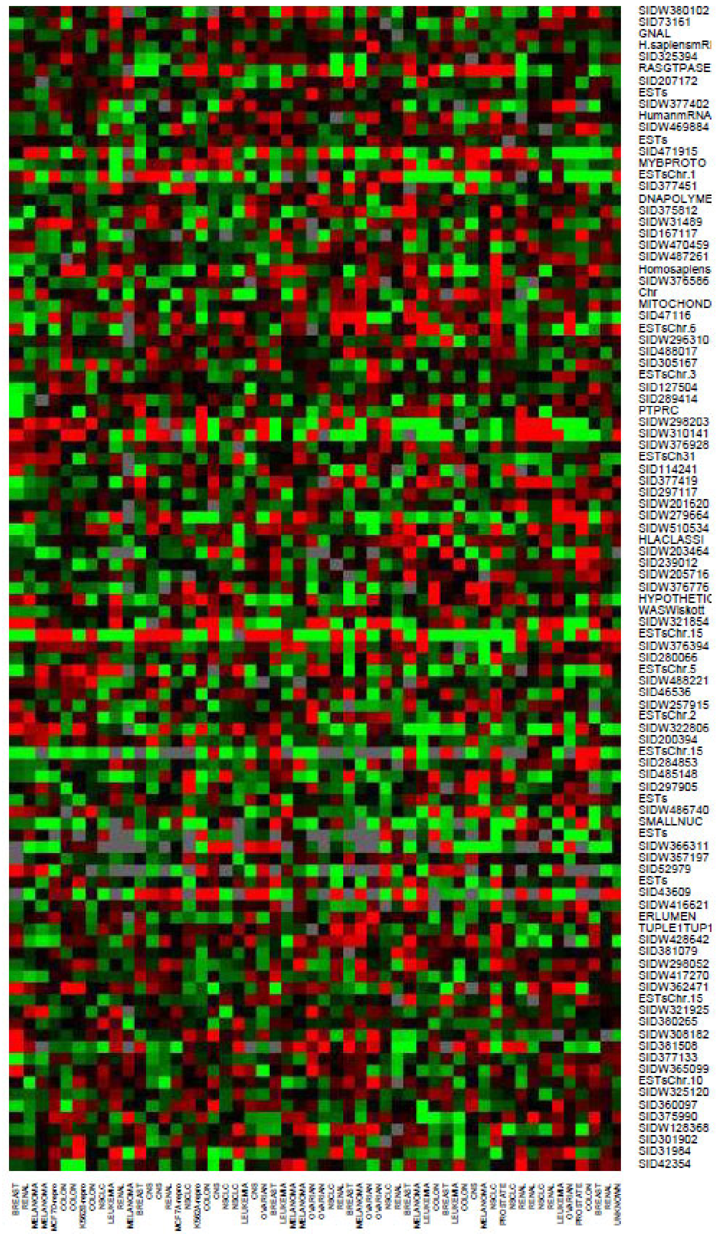
`lpsa`	log of prostate specific antigen

- Supervised learning problem on quantitative data: Regression problem.



Learning to Analyze DNA Data

- Data:
  - Color intensities signifying the abundance levels of mRNA for a number of genes (6830) in several (64) different cell states (samples).
  - Red: over-expressed gene
  - Green: under-expressed gene
  - Gray: gene with missing values
  - Black: normally expressed gene (according to some predefined background)
- Questions:
  1. Which genes show similar expression over the samples – Unsupervised learning
  2. Which samples show similar expression over the genes – Unsupervised learning
  3. Which genes are highly over or under expressed in certain cancers – Supervised learning

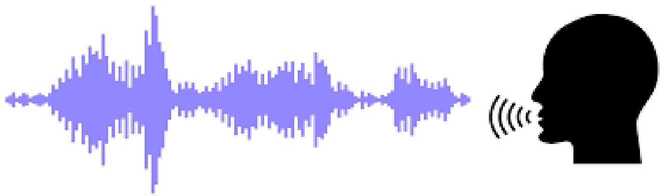


genes (100)

samples (64)

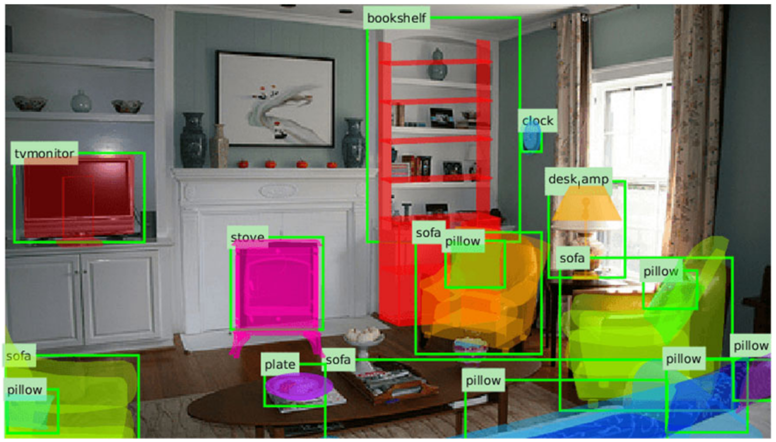
Machine Learning – Practice

Text analysis

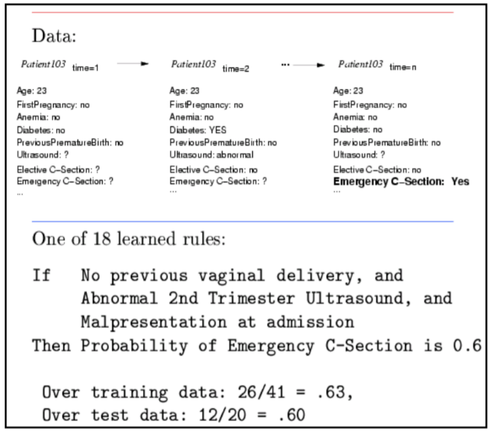


Speech recognition

Control learning



Object recognition



Mining databases

- Logistic regression
- SVM
- Neural networks
- Hidden Markov models
- Reinforcement learning
- Bayesian methods
- ……

Machine Learning – Theory

PAC Learning Theory
(by Leslie Valiant, 1984)

examples (m)

hypothesis complexity (H)

failure probability (δ)

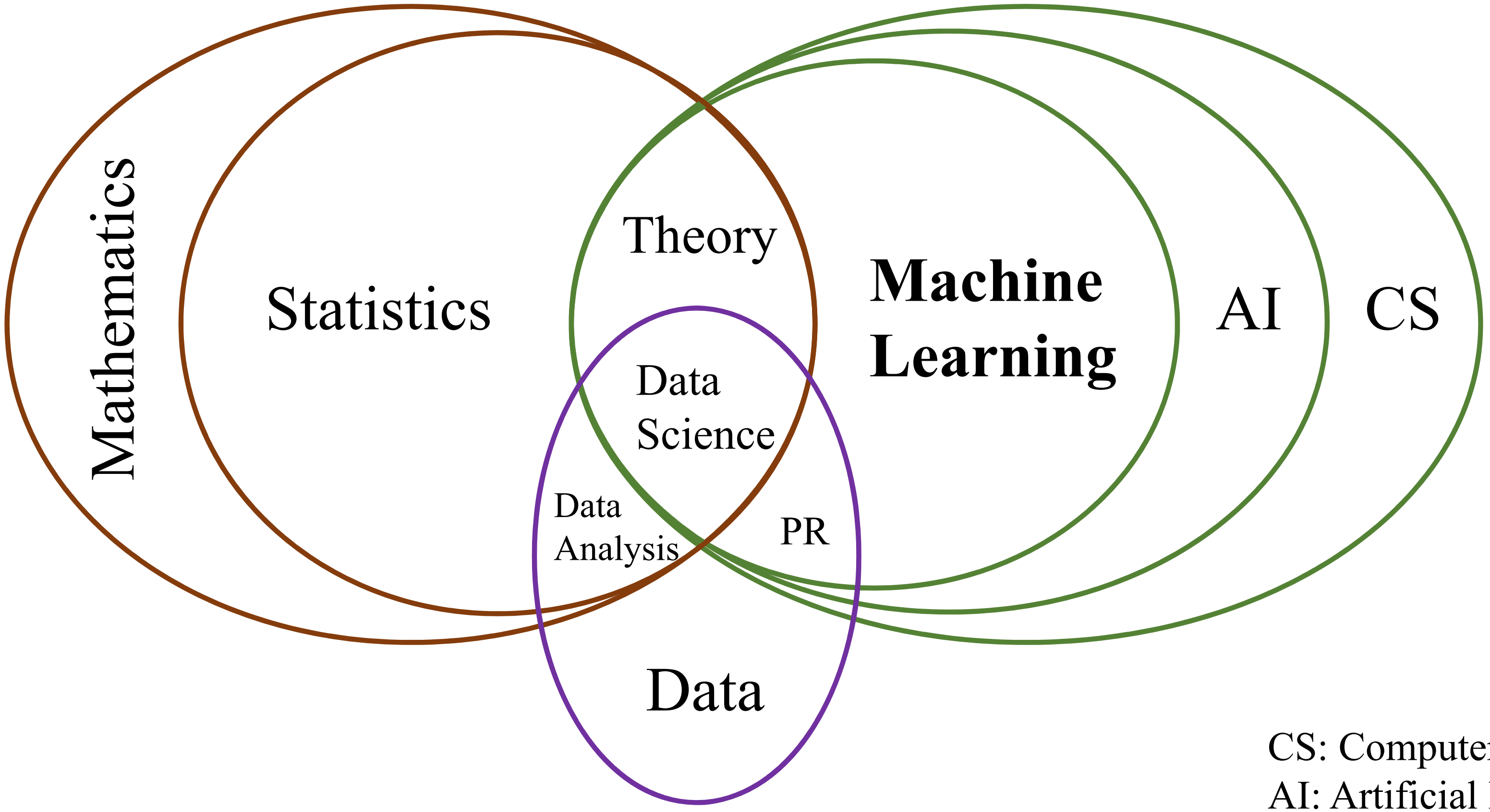
error rate (ε)

$$m \geq \frac{1}{\varepsilon}\left(ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

PAC: Probably Approximately Correct

Other theories for
- Reinforcement learning
- Semi-supervised learning
- ……

Venn Diagram on the Relationship of ML and Statistics



CS: Computer Science
AI: Artificial Intelligence
PR: Pattern Recognition

What You Will Learn in This Course

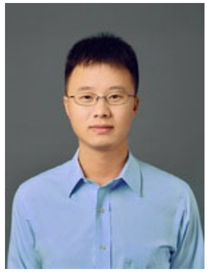
- The primary machine learning and optimization algorithms
  - Ridge regression, lasso, logistic regression, SVM, neural networks, graphical models, unsupervised learning, reinforcement learning…
  - Convex optimization, gradient methods, proximal methods, ADMM, …
- Underlying statistical and computational theory
- Enable to apply the algorithms to solve practical problems
- Enough to read and understand related research papers.

Course Logistics

Optimization and Machine Learning SI151

Faculty

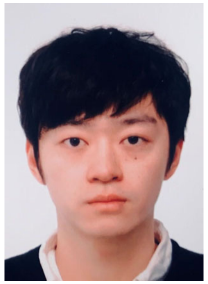
Lu Sun (孙 露)
sunlu1



Yuanming Shi (石 远明)
shiym

Teaching Assistants

Xin Deng
(邓 鑫)
dengxin1



Shuhao Xia
(夏 舒豪)
xiashh

Weikai Xu
(许 惟锴)
xuwk

Xiangyu Yang
(杨 翔宇)
yangxy3

Yuyan Zhou
(周 雨嫣)
zhouyy1

Email address = account_name + @shanghaitech.edu.cn

Optimization and Machine Learning SI151

General information

- Time: Mon. & Wed., 10:15-11:55am
- Online: Blackboard & Piazza
- 16 weeks (64 credit hours)
- Machine learning in weeks 1-12; convex optimization in weeks 13-16

Grading

- Homework: 30%
- Course project: 30%
- Final exam: 40% (tentative)

Optimization and Machine Learning SI151

Highlights

- Please write your HW, project and exam in English;
- For late HW or project, the score will be exponentially decreased;
- Once any plagiarism or cheating is confirmed, relevant assignments or exams will receive 0 points;
- Some may fail if they don't work hard.

Important:

Every online course has one simple question.
Please find and answer it within 10 mins after the course.
It would not be scored, but treated as an attendance record.

Optimization and Machine Learning SI151

Recommended textbooks

- **The Elements of Statistical Learning: Data Mining, Inference and Prediction**, Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman
- **Pattern Recognition and Machine Learning**, Christopher Bishop
- **Convex Optimization**, Stephen Boyd and Lieven Vandenberghe

Some useful online resources

- CMU, machine learning course

http://www.cs.cmu.edu/~ninamf/courses/601sp15/lectures.shtml

- MIT, machine learning course

http://www.ai.mit.edu/courses/6.867-f04/index.html

- Stanford, convex optimization course

https://web.stanford.edu/~boyd/cvxbook/

- CMU, convex optimization

https://www.stat.cmu.edu/~ryantibs/convexopt/

Overview of Supervised Learning I

--- Variable Types and Terminology

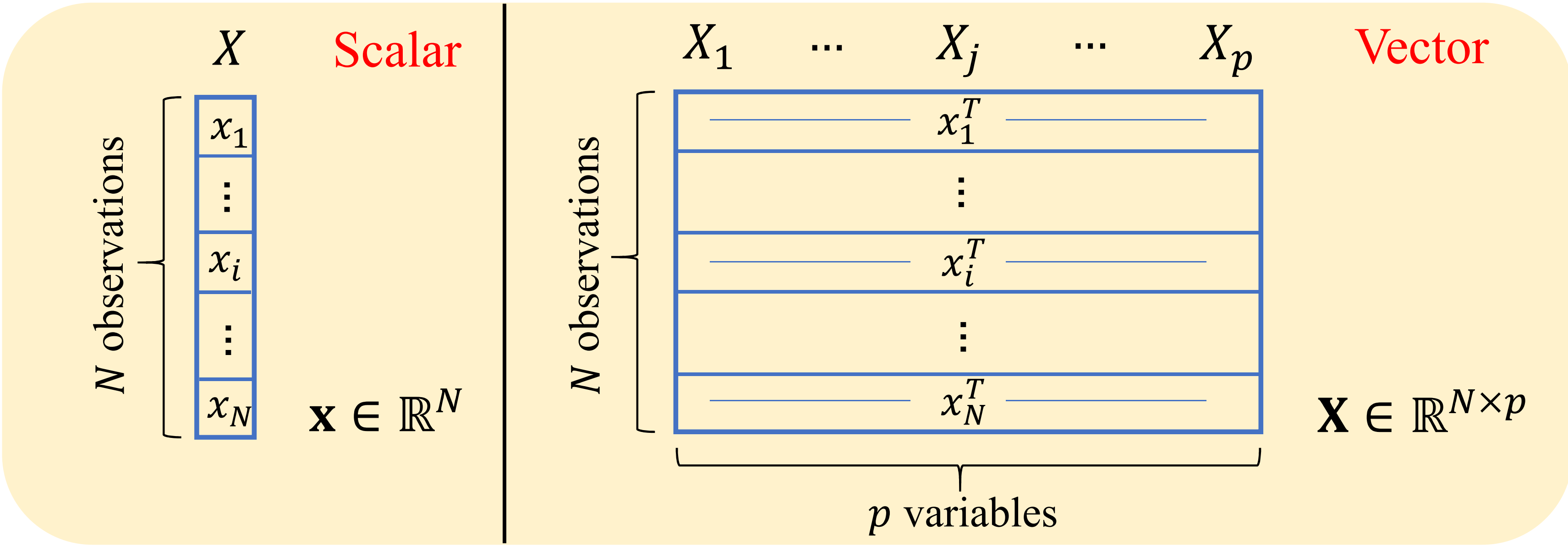
Variable Types and Terminology

Input: a variable X. If X is a vector, its j-th element is X_j

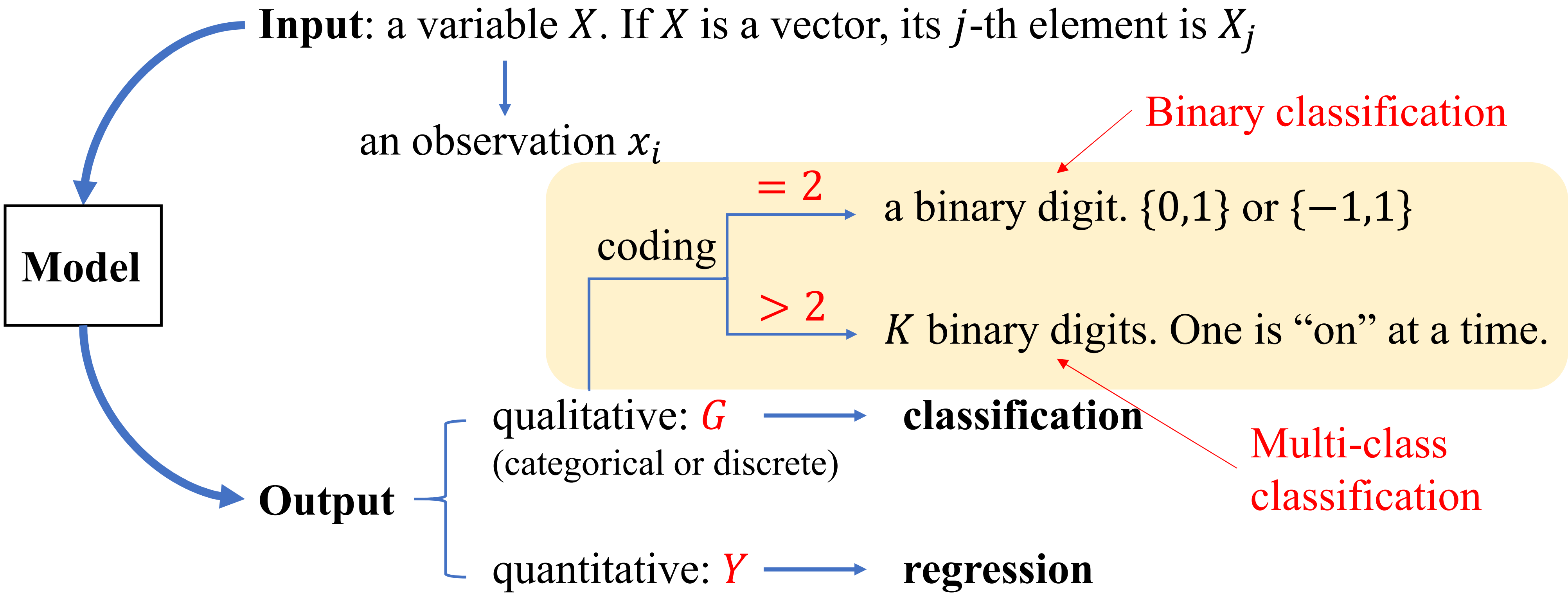
an observation x_i
(scalar or vector)

Typically, we use i to denote the index of observations, while use j to denote the index of variables.

Model



Variable Types and Terminology



Variable Types and Terminology

Input: a variable X. If X is a vector, its j-th element is X_j

an observation x_i

Model

Main question of this course:
Given the value of an input vector X,
make a good prediction $\hat{Y}$ of the output Y.

Output

- qualitative: G (categorical or discrete) → **classification**
- quantitative: Y → **regression**

Overview of Supervised Learning I

--- Least Squares and Nearest Neighbors

Simple Approach 1: Least Squares

- Given inputs:

$$X^T = (X_1, X_2, \dots, X_p)$$

- Predict output Y via the model

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

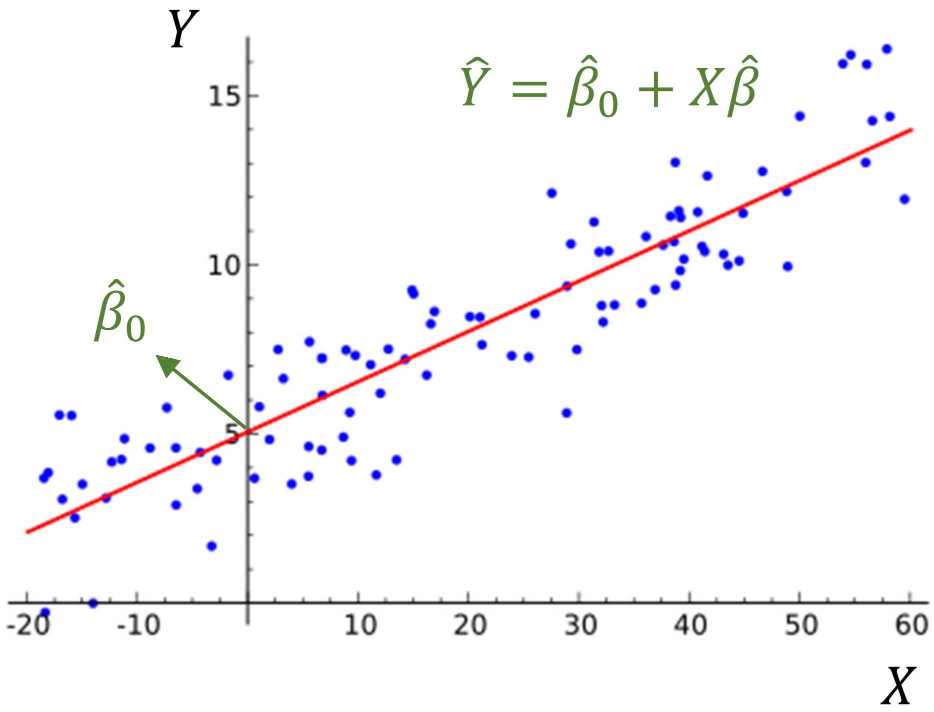
$\hat{\beta}_0$: bias or intercept

- Include the constant variable 1 in X

$$\hat{Y} = X^T \hat{\beta}$$

- Here $\hat{Y}$ is a scalar. If the output $\hat{Y}$ is K-vector, then $\hat{\beta}$ is a $p \times K$ matrix of coefficients.

Multi-output regression



Simple Approach 1: Least Squares

- Given inputs:

$$X^T = (X_1, X_2, \dots, X_p)$$

- Predict output Y via the model

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

$\hat{\beta}_0$: bias or intercept

- Include the constant variable 1 in X

$$\hat{Y} = X^T \hat{\beta}$$

- Here $\hat{Y}$ is a scalar. If the output $\hat{Y}$ is K-vector, then $\hat{\beta}$ is a $p \times K$ matrix of coefficients.

- In the $(p+1)$-dimensional input-output space, $(X, \hat{Y})$ represents a hyperplane
- If the constant is included in X, then the hyperplane goes through the origin

$$f(X) = X^T \beta$$

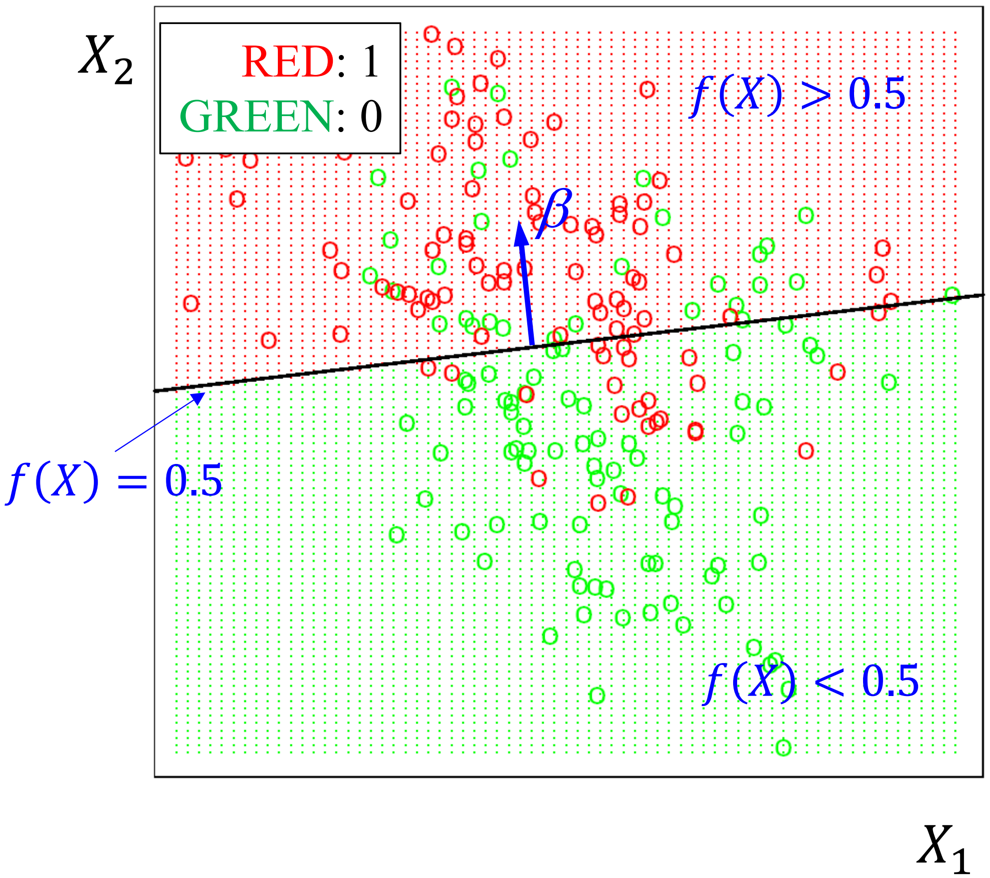
is a linear function
- Its gradient

$$\boxed{f'(X) = \beta}$$

is a vector that points in the steepest uphill direction.

For the derivatives of vectors and matrices, please refer to:
- **The Matrix Cookbook**. Kaare Brandt Petersen and Michael Syskind Pedersen

Simple Approach 1: Least Squares



- In the $(p + 1)$-dimensional input-output space, $(X, \hat{Y})$ represents a hyperplane
- If the constant is included in X, then the hyperplane goes through the origin

$$f(X) = X^T \beta$$

is a linear function
- Its gradient

$$f'(X) = \beta$$

is a vector that points in the steepest uphill direction.

Simple Approach 1: Least Squares

- Training procedure:
  Method of *least-squares*
- N = #observations
- Minimize the *residual sum of squares*

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$$

Or equivalently,

$$\begin{aligned}\mathrm{RSS}(\beta) &= (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) \\ &= \|\mathbf{y} - \mathbf{X}\beta\|_2^2\end{aligned}$$

- This quadratic function always has a
  global minimum, but it may not be unique.

Note: for an arbitrary vector $\boldsymbol{a}$, we have the squared ℓ_2-norm $\|\boldsymbol{a}\|_2^2 = \boldsymbol{a}^T \boldsymbol{a}$.

Simple Approach 1: Least Squares

- Training procedure:
  Method of *least-squares*
- N = #observations
- Minimize the *residual sum of squares*

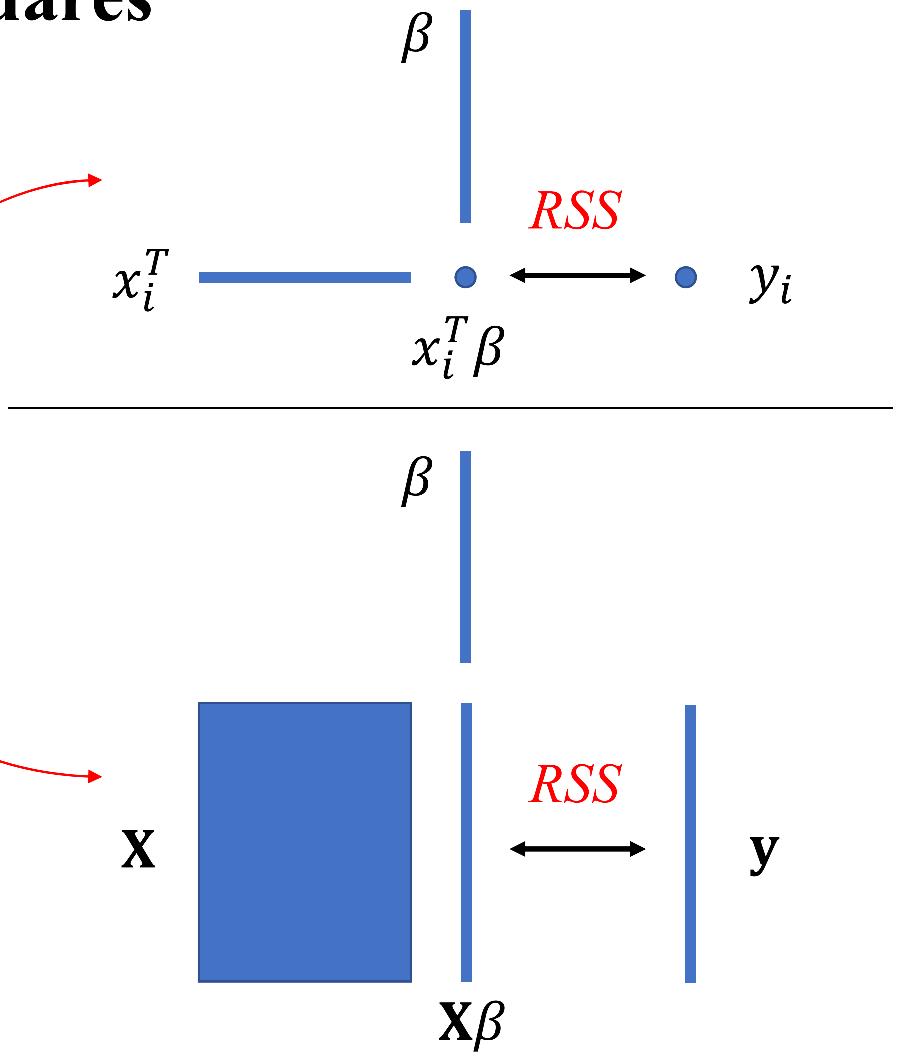
$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} \boxed{(y_i - x_i^T\beta)^2}$$

Or equivalently,

$$\mathrm{RSS}(\beta) = \boxed{(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)}$$
$$= \|\mathbf{y} - \mathbf{X}\beta\|_2^2$$

- This quadratic function always has a global minimum, but it may not be unique.

Q: What is the difference among x_i, x_i^T, $\mathbf{x}$, X and $\mathbf{X}$?



Simple Approach 1: Least Squares

- Training procedure:
  Method of *least-squares*
- N = #observations
- Minimize the *residual sum of squares*

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$$

Or equivalently,

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$
$$= \|\mathbf{y} - \mathbf{X}\beta\|_2^2$$

- This quadratic function always has a global minimum, but it may not be unique.
- Differentiating w.r.t. β yields the *normal equations*

$$\mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta) = 0$$

- If $\mathbf{X}^T\mathbf{X}$ is nonsingular, then the unique solution is

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- The fitted value at an arbitrary input x_0 is

$$\hat{y}(x_0) = x_0^T \hat{\beta}$$

- The entire fitted surface is characterized by $\hat{\beta}$.

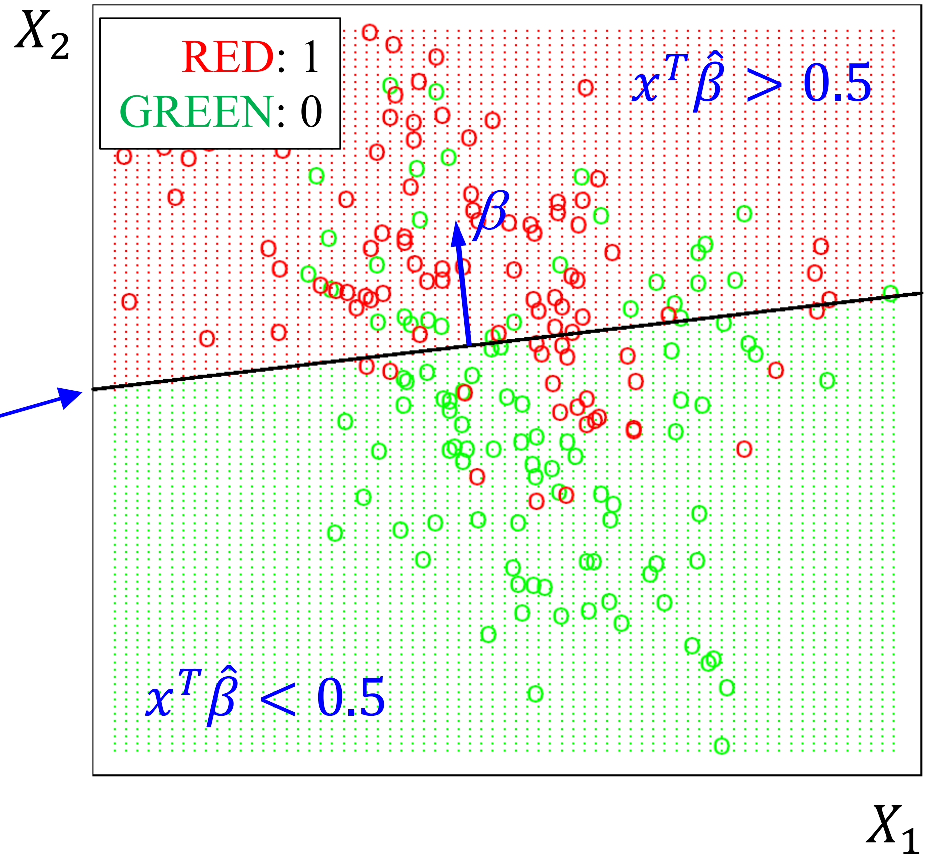
Simple Approach 1: Least Squares

Example:

- Data on two inputs X_1 and X_2.
- Output variable has values GREEN (coded 0) and RED (coded 1).
- 100 points per class.
- Regression line is defined by

$$x^T\hat{\beta} = 0.5.$$

- Easy but many misclassifications if the problem is not linear.



Difference between Statistics and Machine Learning

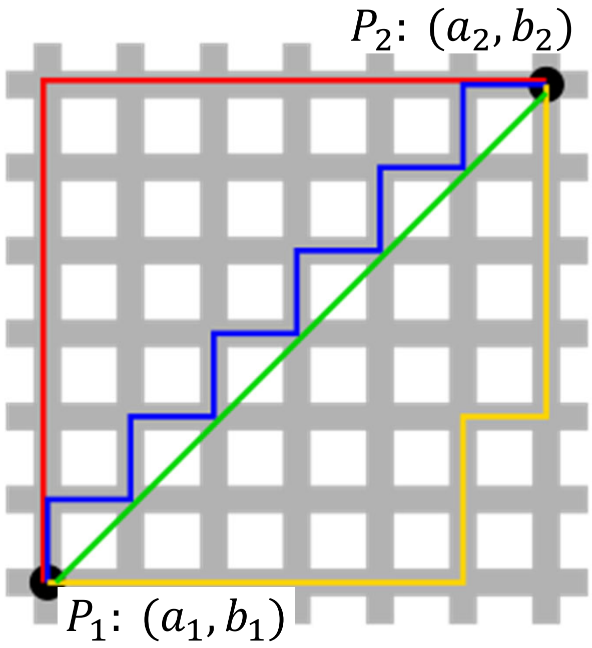
Symbol	Statistics	Machine Learning
X	variable, covariable predictor independent variable	feature attribute
Y	response dependent variable	label
x_i	observation data point	example instance
β	weights coefficients	parameters
$f(\cdot)$	model	learner

Simple Approach 2: Nearest Neighbors

- Use observations in the training set closest to the given input.

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i.$$

- $N_k(x)$ is the set of the k closest points to x is the training sample
- Average the outcome of the k closest training sample points



$\ell_1(P_1, P_2)$
$= |a_2 - a_1| + |b_2 - b_1|$

$\ell_2(P_1, P_2)$
$= \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2}$

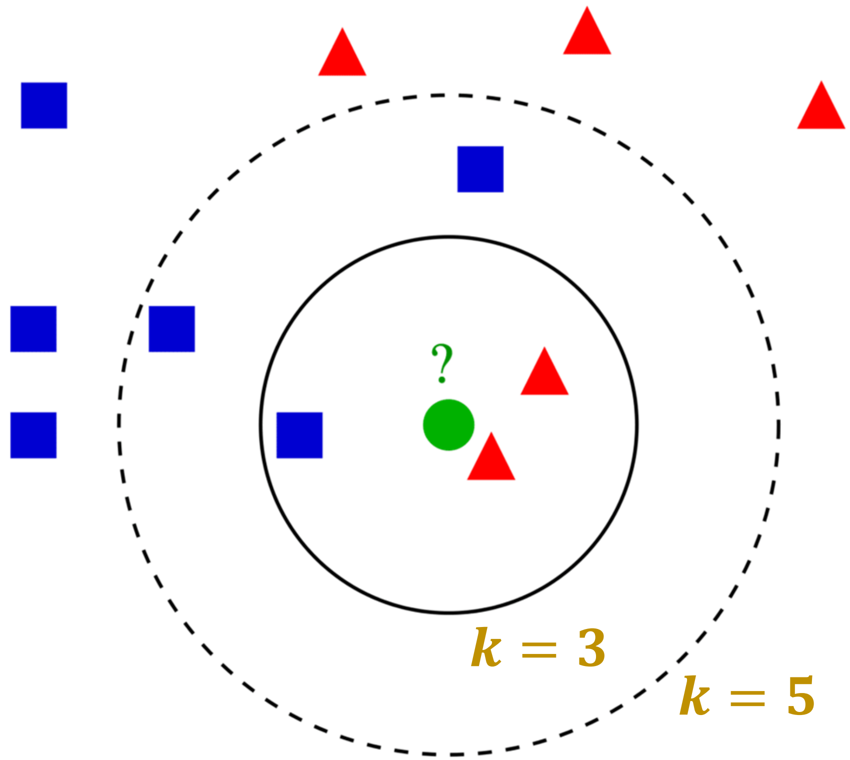
Taxicab geometry ($\boldsymbol{\ell_1}$) versus **Euclidean distance ($\boldsymbol{\ell_2}$)** : In taxicab geometry, the red, yellow, and blue paths all have the same shortest path length of 12. In Euclidean geometry, the green line has length $6\sqrt{2} \approx 8.49$ and is the unique shortest path.

Simple Approach 2: Nearest Neighbors

- Use observations in the training set closest to the given input.

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i.$$

- $N_k(x)$ is the set of the k closest points to x is the training sample
- Average the outcome of the k closest training sample points



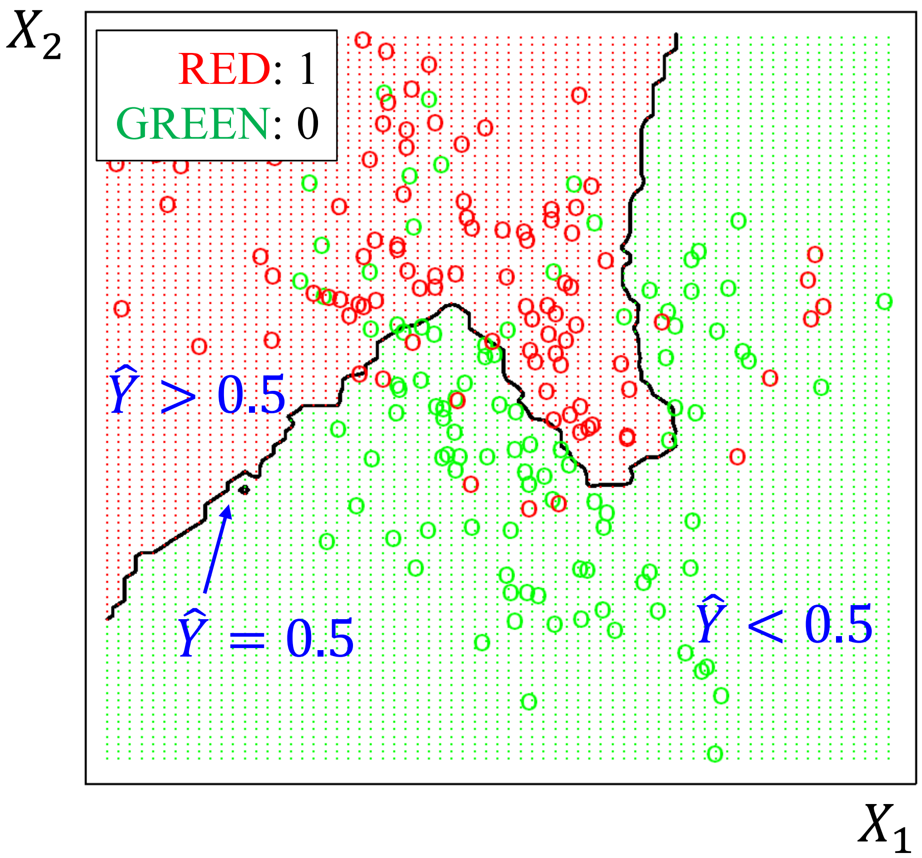
Simple Approach 2: Nearest Neighbors

- Use observations in the training set closest to the given input.

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i.$$

- $N_k(x)$ is the set of the k closest points to x is the training sample
- Average the outcome of the k closest training sample points
- Fewer misclassifications

15-nearest neighbors averaging



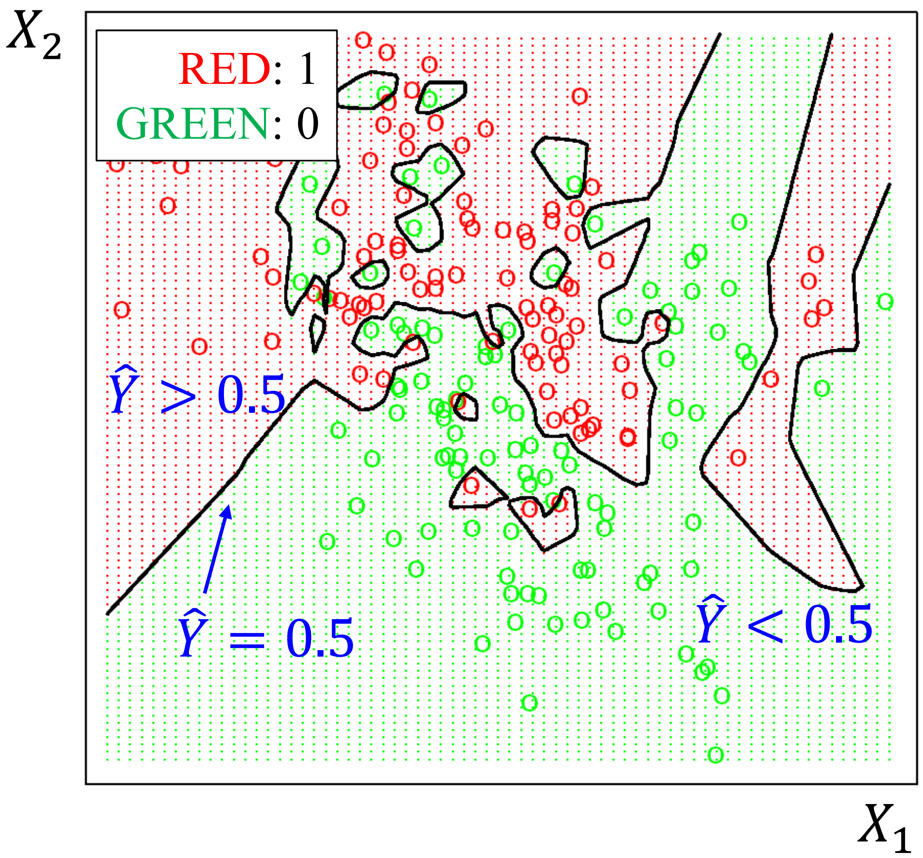
Simple Approach 2: Nearest Neighbors

- Use observations in the training set closest to the given input.

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i.$$

- $N_k(x)$ is the set of the k closest points to x is the training sample
- Average the outcome of the k closest training sample points
- No misclassifications: overtraining

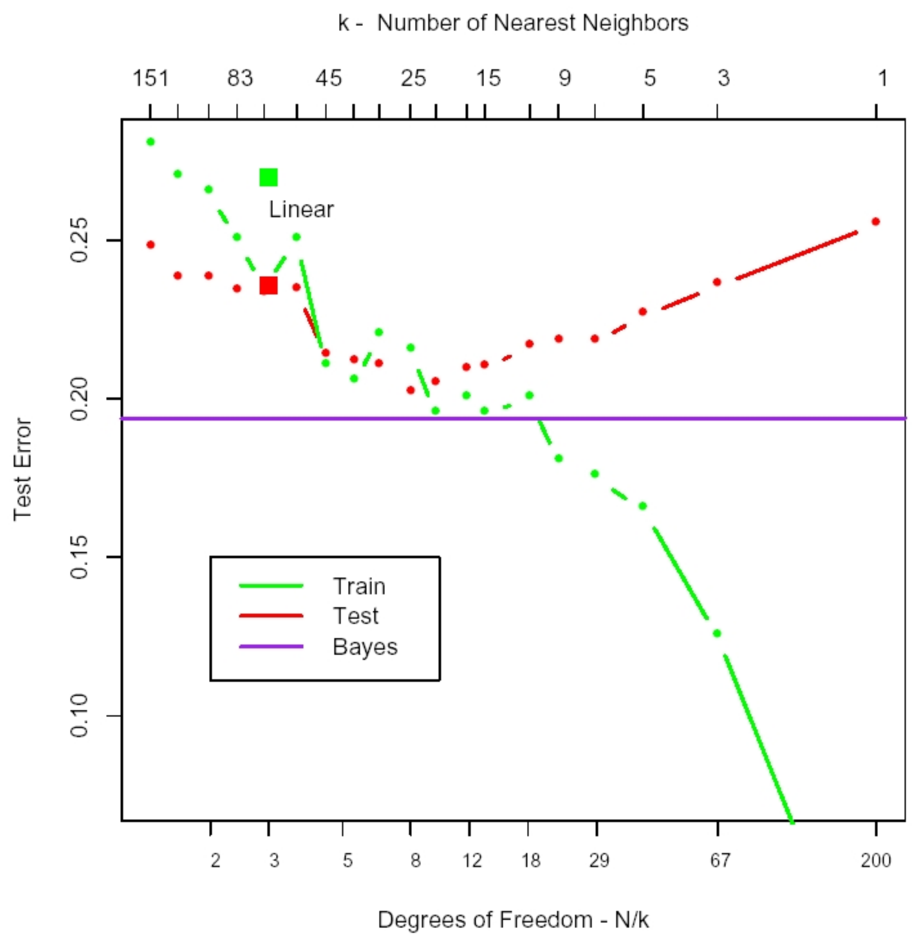
1-nearest neighbors averaging



Comparison of the Two Approaches

Mixture of the two scenarios:

- Step 1: Generate 10 means m_k from the bivariate Gaussian distribution $N((1,0)^T, \mathbf{I})$ and label this class GREEN. Likewise, label the class RED.
- Step 2: For each class, generate 100 observations as follows:
  - For each observation, pick an m_k at random with probability 0.1;
  - Generate a data point according to $N(m_k, \mathbf{I}/5)$.



Classification results on 10,000 testing examples.

Comparison of the Two Approaches

Least squares	**k-nearest neighbors**
p parameters (p = #variables)	$\frac{N}{k}$ parameters (k: hyperparameter) (N = #observations)
Low variance (robust)	High variance (not robust)
High bias (strong assumption)	Low bias (mild assumption)
Good for Scenario 1: Training data in each class generated from a two-dimensional Gaussian, the two Gaussians are independent and have different means.	Good for Scenario 2: Training data in each class generated from a mixture of 10 low-variance Gaussians, with means again distributed as Gaussian.

Variants of the Simple Approaches

- Kernel methods: use weights that decrease smoothly to zero with distance from the target point, rather than the 0/1 cutoff used in nearest-neighbor methods
- In high-dimensional spaces, some variables are emphasized more than others
- Local regression fits linear models (by least squares) locally rather than fitting constants locally
- Projection pursuit and neural network models are sums of nonlinearly transformed linear models