L2: Basics of Supervised Learning

Shan Wang

Lingnan College, Sun Yat-sen University

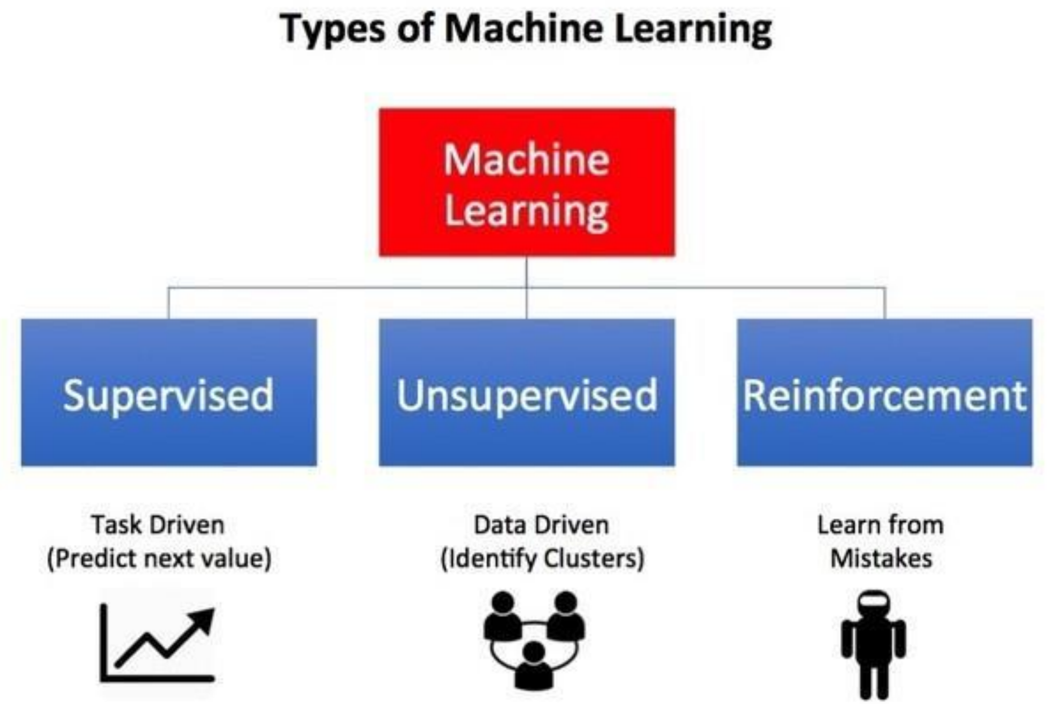
2020 Data Mining and Machine Learning LN3119
https://wangshan731.github.io/DM-ML/

Course outline

- Supervised learning
  - Linear regression
  - Logistic regression
  - SVM and kernel
  - Tree models
- Unsupervised learning
  - Clustering
  - PCA
  - EM
- Deep learning
  - Neural networks
  - Convolutional NN
  - Recurrent NN
- Reinforcement learning
  - MDP
  - ADP
  - Deep Q-Network

Last lecture

- What is machine learning
- Machine learning applications
- History of machine learning
- Classifications of machine learning



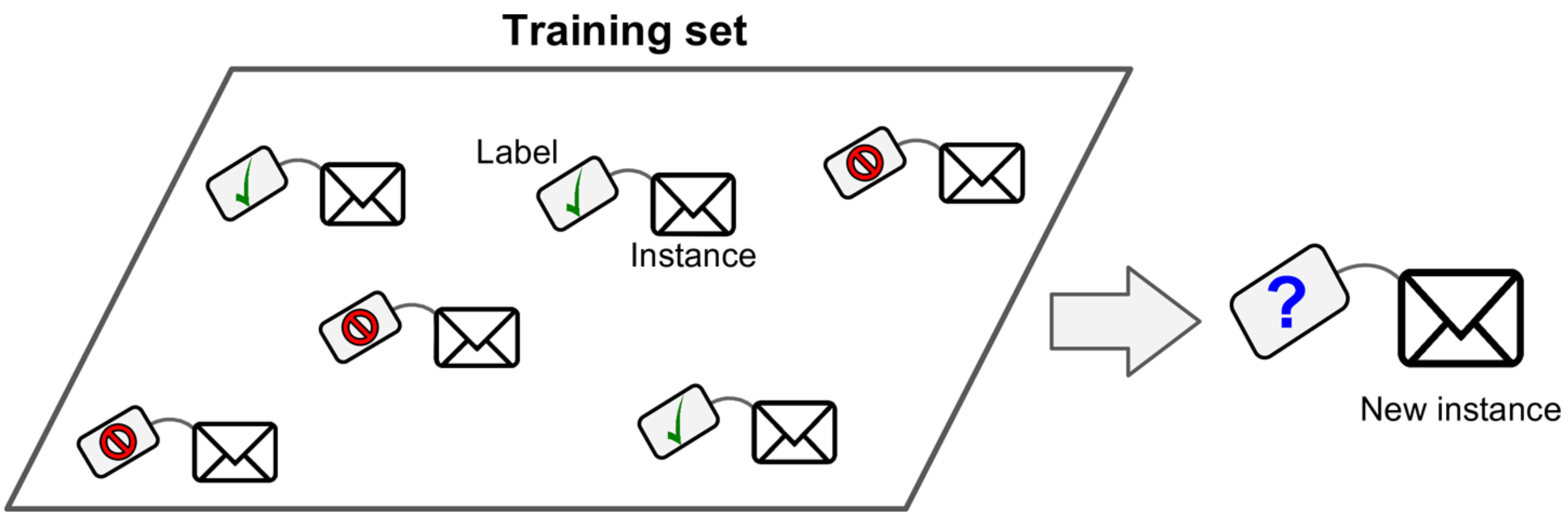
This lecture

- Basics of supervised learning
  - Learning process
  - Discriminative models and generative models
  - Machine learning three elements
    - Model
    - Strategy
    - Algorithm
  - Model evaluation
  - Model selection & Regularization
  - Cross validation

Reference: CS420, Weinan ZHANG (SJTU); 10-601, Mary MCGLOHON (CMU)

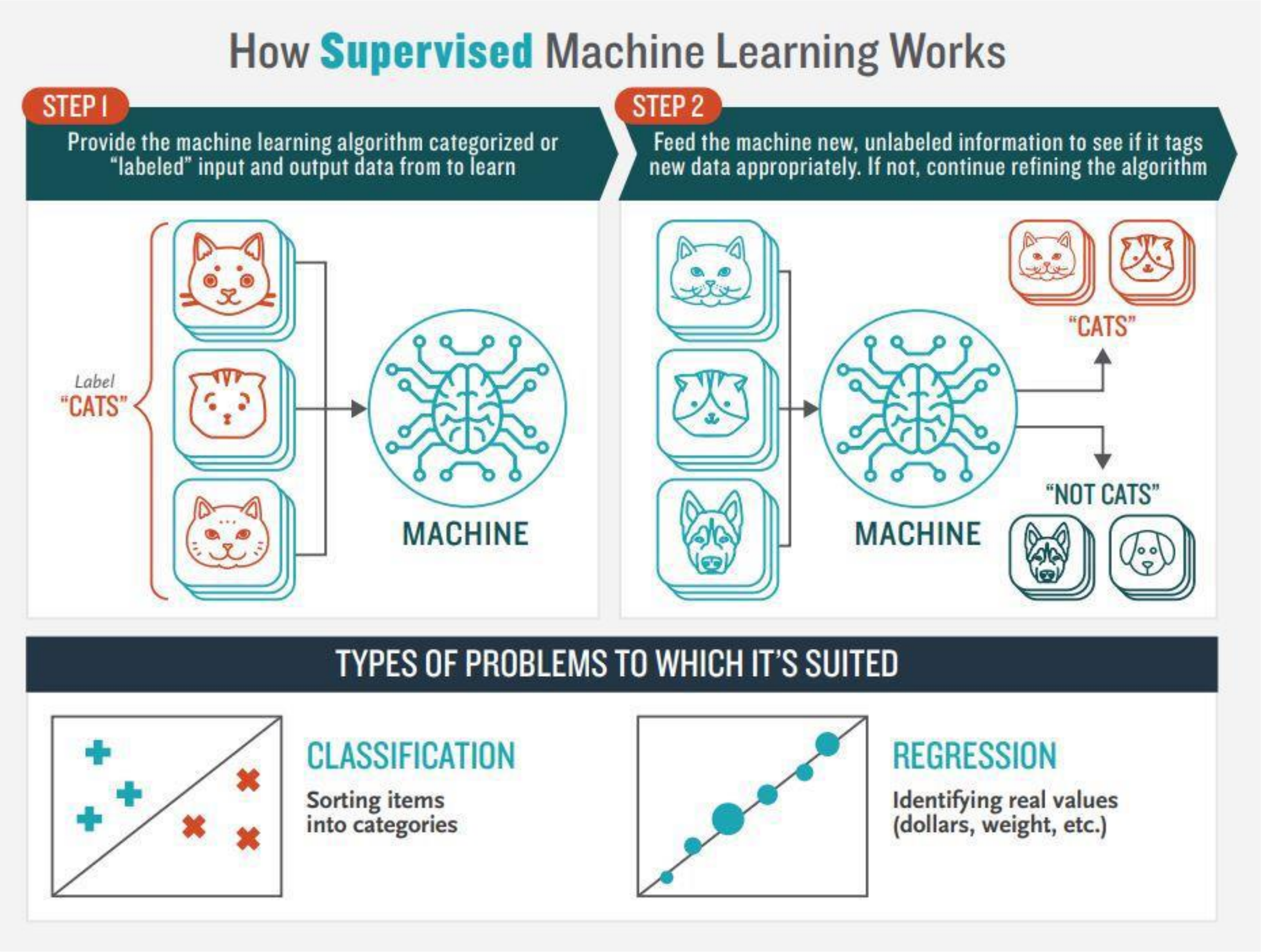
Supervised learning

- Learning a function that maps an input to an output based on example input-output pairs

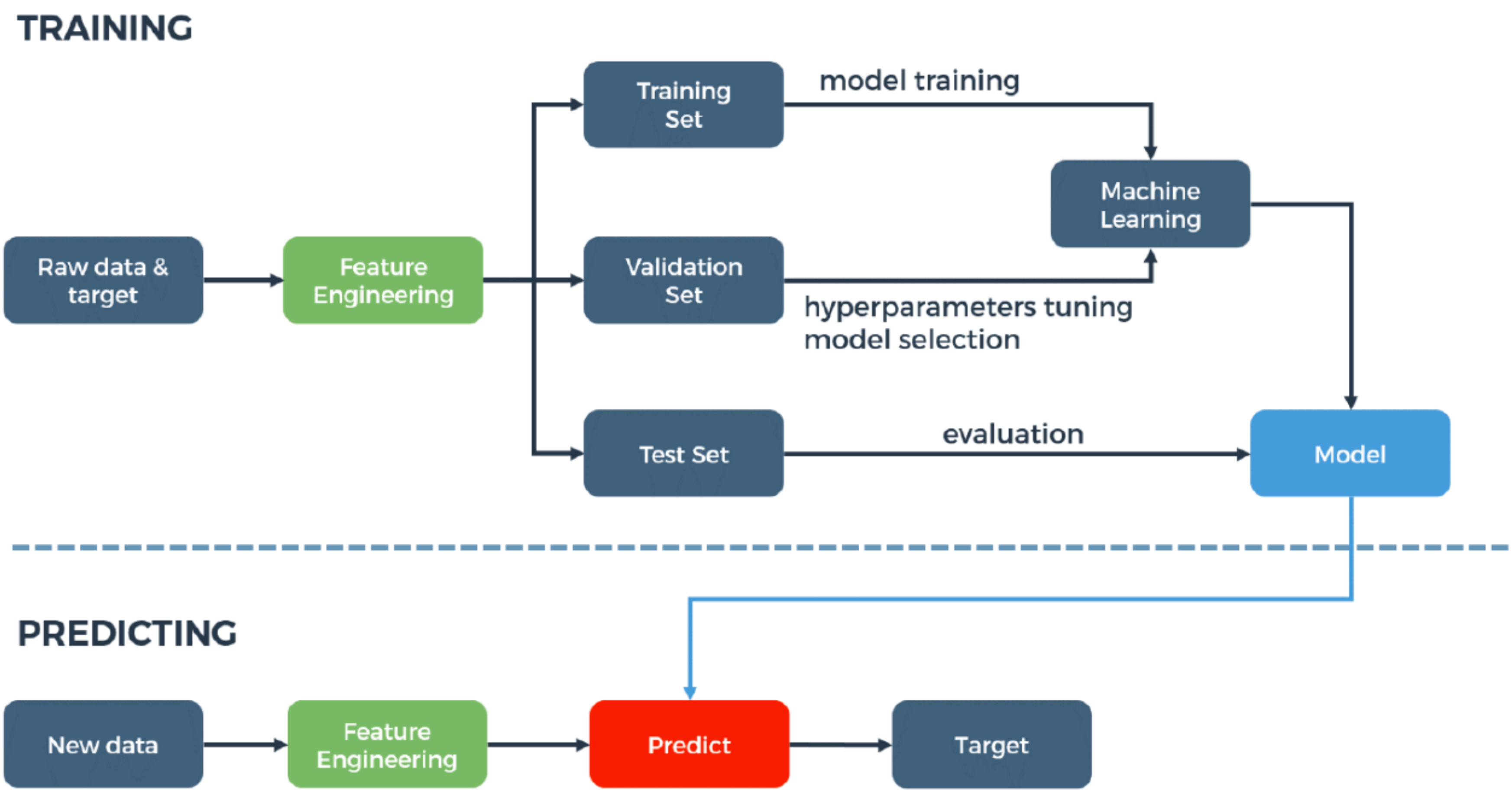


Learning process

How supervised learning works



Supervised learning process



- Basic assumption: there exist the same patterns across training, test and new data

Discriminative models and generative models

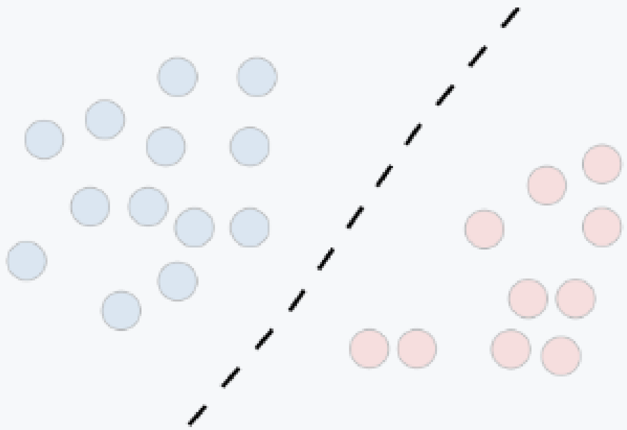
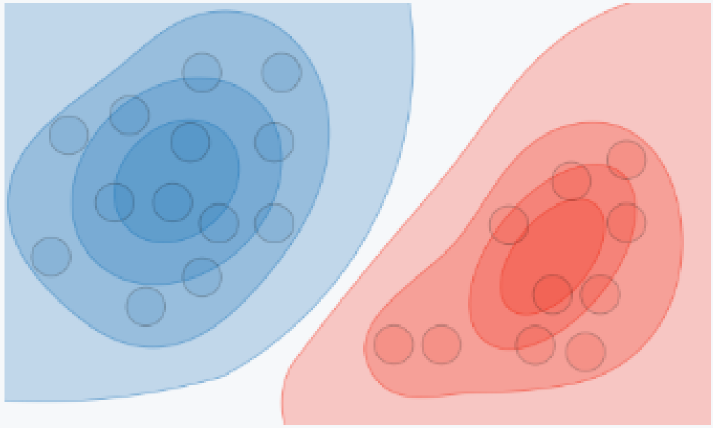
What is discriminative model

- Modeling the dependence of unobserved variables on observed ones
- a.k.a. conditional models
- Directly estimate: $p_\theta(y|x)$

What is generative model

- Modeling the joint probabilistic distribution of data
- i.e., modeling $p_\theta(x, y)$
- Then do conditional inference
  - $p_\theta(y|x) = \frac{p_\theta(x,y)}{p_\theta(x)} = \frac{p_\theta(x,y)}{\sum_{y'} p_\theta(x,y')}$

Discriminative vs. generative

	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

ML THREE elements

Model

Model

- Spaces
  - Input space (feature space) X, output space (labeled space) Y
- Training data
  - Sample S of size N drawn i.i.d. from $X \times Y$ according to distribution D:
  - $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$
- Hypothesis set: $F \subseteq Y^X$ (mappings from X to Y)
  - Space of possible models, e.g. all linear functions
  - Depends on feature structure and prior knowledge about the problem

ML THREE elements

Strategy

Strategy

- Objective
  - Find a good hypothesis $f \in F$
- What is a good f
  - A f with small generalization error
- Loss function: $L: Y \times Y \rightarrow \mathbb{R}$
  - $L(\hat{y}, y)$: loss of predicting $\hat{y}$ when the true output is y
    - Binary classification: $L(\hat{y}, y) = 1_{\hat{y} \neq y}$
    - Regression: $L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$
- Generalization error
  - $R(f) = \mathbb{E}_{(\boldsymbol{x}, y) \sim D}[L(f(\boldsymbol{x}), y)]$
- Empirical error
  - $\hat{R}(f) = \frac{1}{N} \sum_{i=1}^{N} L(f(\boldsymbol{x}_i), y_i)$

Generalization error bound

- Finite hypothesis set F
- Generalization error bound
  - For any function $f \in F$, with probability no less than $1-\delta$, it satisfies

$$R(f) \leq \hat{R}(f) + \epsilon(d, N, \delta)$$

Where

$$\epsilon(d, N, \delta) = \sqrt{\frac{1}{2N}(\log d + \log \frac{1}{\delta})}$$

- N: number of training instances
- d: number of functions in F

Bonus question: How to prove it?

Hint: Hoeffding Inequality

Generalization error bound - Hint

Lemma: Hoeffding Inequality

Let $X_1, X_2, \ldots, X_n$ be bounded independent random variables $X_i \in [a, b]$, the average variable *Z* is

$$Z = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Then the following inequalities satisfy:

$$P(Z - \mathbb{E}[Z] \geq t) \leq \exp\left(\frac{-2nt^2}{(b-a)^2}\right)$$

$$P(\mathbb{E}[Z] - Z \geq t) \leq \exp\left(\frac{-2nt^2}{(b-a)^2}\right)$$

Maximum likelihood estimation

- Maximum likelihood estimation
  - We know $x_1, x_2, \dots, x_N \sim N(\mu, \sigma^2)$, how to know μ?
  - Set up likelihood equation: $P(\boldsymbol{x}|\mu, \sigma^2)$, and find μ to maximize it.
  - If $x_1, x_2, \dots, x_n$ are independent

$$\mathcal{L}(\boldsymbol{x}) = P(\boldsymbol{x}|\mu, \sigma^2) = \prod_{i=1}^{N} P(x_i|\mu, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

  - Taking the log likelihood (we get to do this since log is monotonic) and removing some constants:

$$\log(\mathcal{L}(\boldsymbol{x})) \propto \sum_{i=1}^{N} -(x_i - \mu)^2$$

  - FOC:

$$\mu = \frac{1}{N} \sum x_i$$

About MLE

- Maximum likelihood estimator:

$$\theta = \operatorname{argmax} P(\boldsymbol{x}|\theta)$$

where

- $P(\boldsymbol{x}|\theta)$ is the joint **probability density function** of observations $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$
- **Frequentist**: only believe the data
- It is almost unbiased
- If we have enough data, it is great

Maximum A Posterior

- What if you have some ideas about your parameter?
- Bayes' Rule

$$P(\theta|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\theta)P(\theta)}{P(\boldsymbol{x})} = \frac{P(\boldsymbol{x}|\theta)P(\theta)}{\sum_{\Theta} P(\theta,\boldsymbol{x})\,P(\boldsymbol{x})} = \frac{P(\boldsymbol{x}|\theta)P(\theta)}{\sum_{\Theta} P(\boldsymbol{x}|\theta)P(\theta)}$$

- Maximum A Posterior

$$\theta = \operatorname{argmax} P(\theta|\boldsymbol{x}) = \operatorname{argmax} \frac{P(\boldsymbol{x}|\theta)P(\theta)}{\sum_{\Theta} P(\boldsymbol{x}|\theta)P(\theta)}$$

- Equivalent to maximize the numerator $P(\boldsymbol{x}|\theta)P(\theta)$
- Different from MLE:
  - Assume there is a **prior** distribution $P(\theta)$
  - We have some knowledge about the parameter

About MAP

- Example
  - There are two bags
    - Bag A: 50% Green balls+ 50% Red balls
    - Bag B: 100% Red balls
  - If you consecutively pick two red balls from one bag, which bag is most likely?
  - MLE
    - A: $P(\boldsymbol{x}|\theta) = 0.25$; B:$P(\boldsymbol{x}|\theta) = 1$; so B
  - MAP – we know get bag A with 0.9, get bag B with 0.1
    - A: $P(\boldsymbol{x}|\theta)P(\theta) = 0.25 * 0.9 = 0.225$
    - B: $P(\boldsymbol{x}|\theta)P(\theta) = 1 * 0.1 = 0.1$
    - So A
- If the prior is uniform distribution
  - We do not have knowledge about the parameter
  - MAP=MLE

ML THREE elements

Algorithm

Algorithm

- Objective
  - Find a good hypothesis $f \in F$ with small generalization error
- Solve $\min_{f} \hat{R}(f)$
  - An optimization problem
    - Analytical solution
    - Gradient method
    - Heuristics

Model evaluation

Confusion matrix

- Confusion Matrix
  - TP – True Positive ; FP – False Positive
  - FN – False Negative; TN – True Negative

	Predicted class		
Actual class		Class=yes	Class=no
	Class=yes	TP	FN
	Class=no	FP	TN

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

- Any limitation?

Accuracy measure

- Limitation of accuracy
  - Consider a 2-class problem
    - Number of Class 0 examples = 9990
    - Number of Class 1 examples = 10
  - If a “stupid” model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %

- The accuracy is misleading because the model does not detect any example in class 1

Other measures

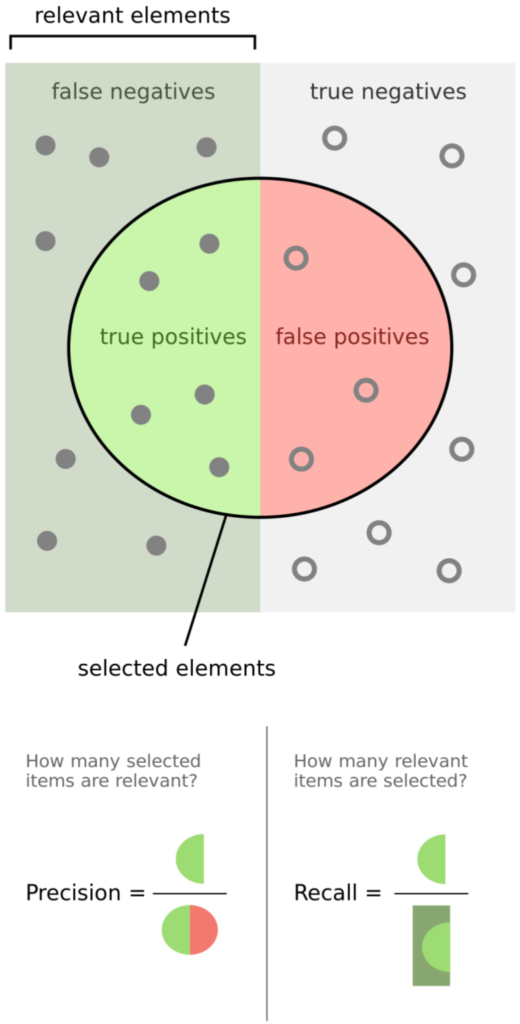
- Cost-sensitive measures

		Predicted class	
		Class=yes	Class=no
Actual class	Class=yes	TP	FN
	Class=no	FP	TN

$$Precision\ (p) = \frac{TP}{TP + FP}$$

$$Recall\ (r) = \frac{TP}{TP + FN}$$

$$F1 - measure\ (F) = \frac{2rp}{r + p} = \frac{2TP}{2TP + FP + FN}$$



- F1 measure is best if there is some sort of balance between precision (p) & recall (r)

Example

- A school is running a machine learning primary diabetes scan on all students
  - Diabetic (+) / Healthy (-)
  - False positive
    - a false alarm
  - False negative
    - Predict a diabetic student as a healthy student
    - Worse!
- Precision
  - How many of those who we labeled as diabetic are actually diabetic?
- Recall
  - Of all the students who are diabetic, how many of those we correctly predict?

Which to choose?

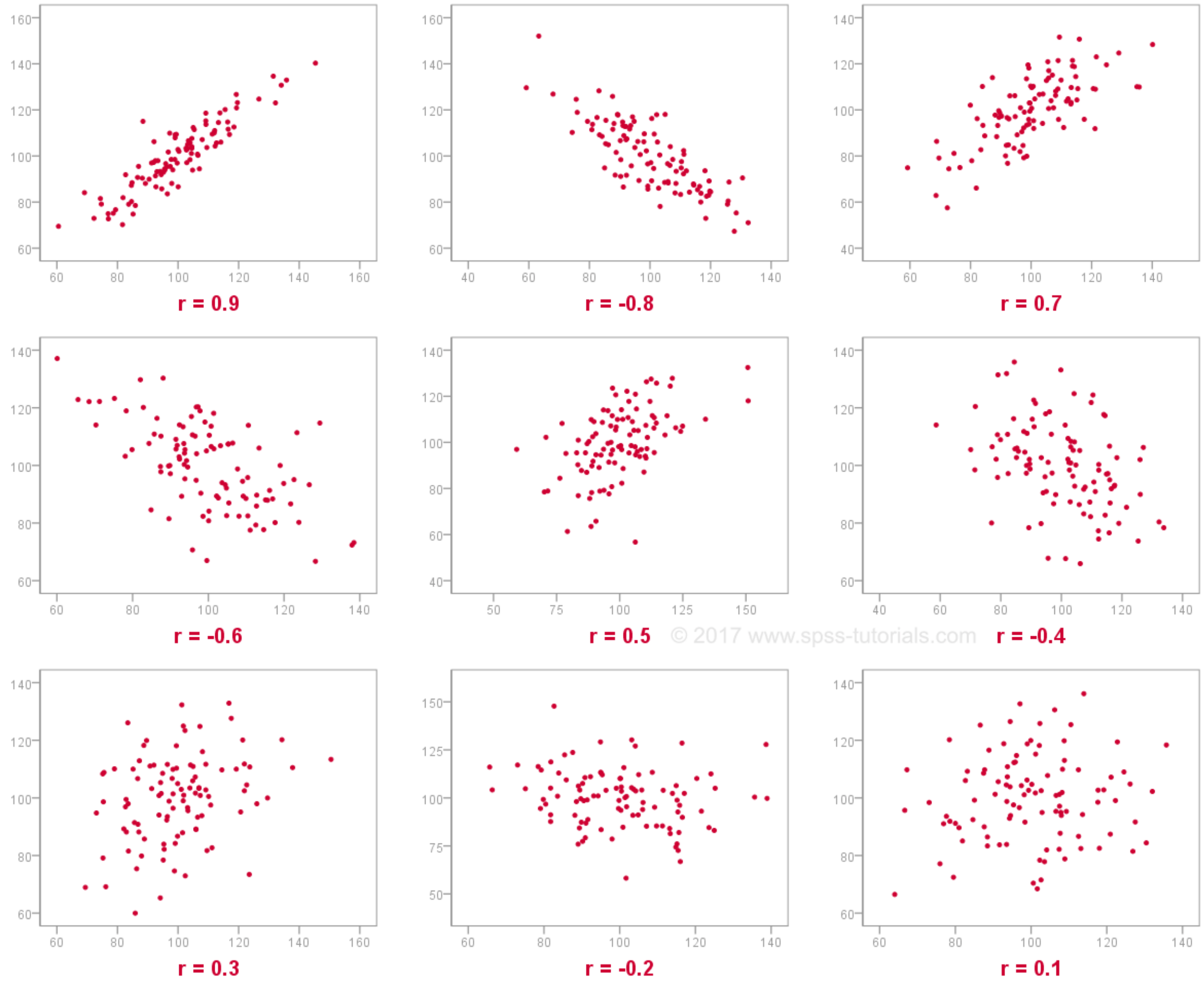
- Accuracy
  - FN & FP counts are close
  - FN & FP have similar costs
- F1 measure
  - costs of FP and FN are different
  - uneven class distribution
- Recall
  - FP is far better than FN
  - e.g., diabetes
- Precision
  - FN is far better than FP
  - e.g., spam emails

Correlation

- Pearson
  - measures the linear association between continuous variables

$$r_{XY} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Correlation (cont.)



Correlation (cont.)

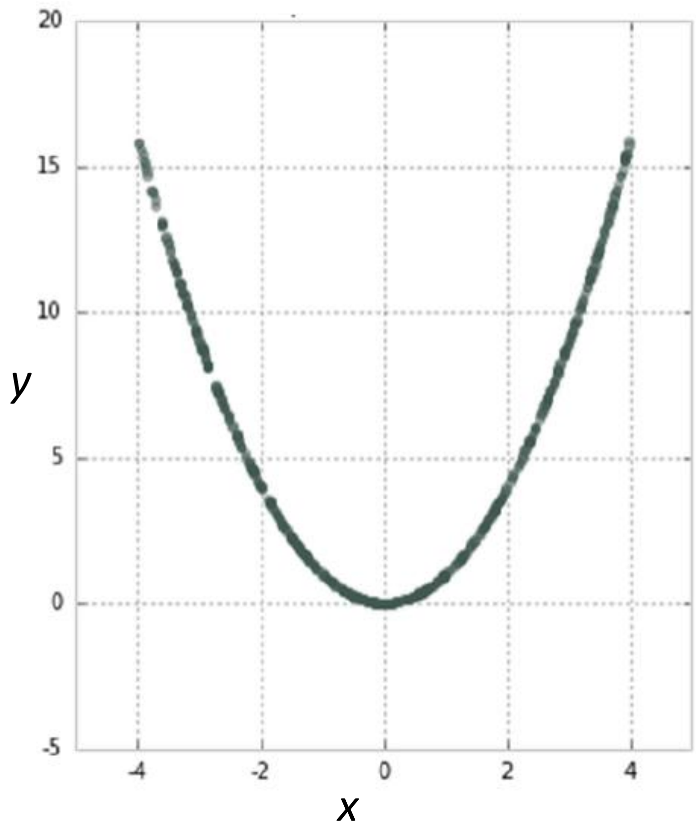
- Pearson correlation
  - measures the linear association between continuous variables

$$r_{XY} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

- Any limitation?

Correlation (cont.)

- Limitation of Pearson correlation
  - Only linear correlation can be detected.
  - Clearly, there are some relationship between *x* and *y*, but the correlation is only 0.02.



R-squared

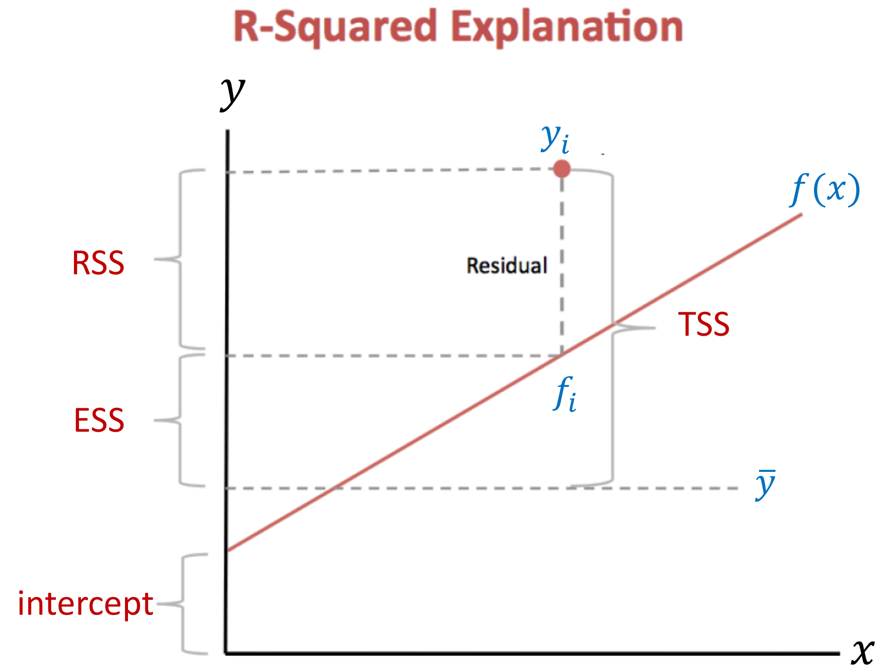
- Coefficient of determination (R^2)
  - measures how much of the residue can be explained by the regression line

Total Tum of Squares $TSS = \sum(y_i - \bar{y})^2$
(Total variance)

Explained Sum of Squares $ESS = \sum(f_i - \bar{y})^2$
(Explained variance)

Residual Sum of Squares $RSS = \sum(f_i - y_i)^2$
(Unexplained variance)

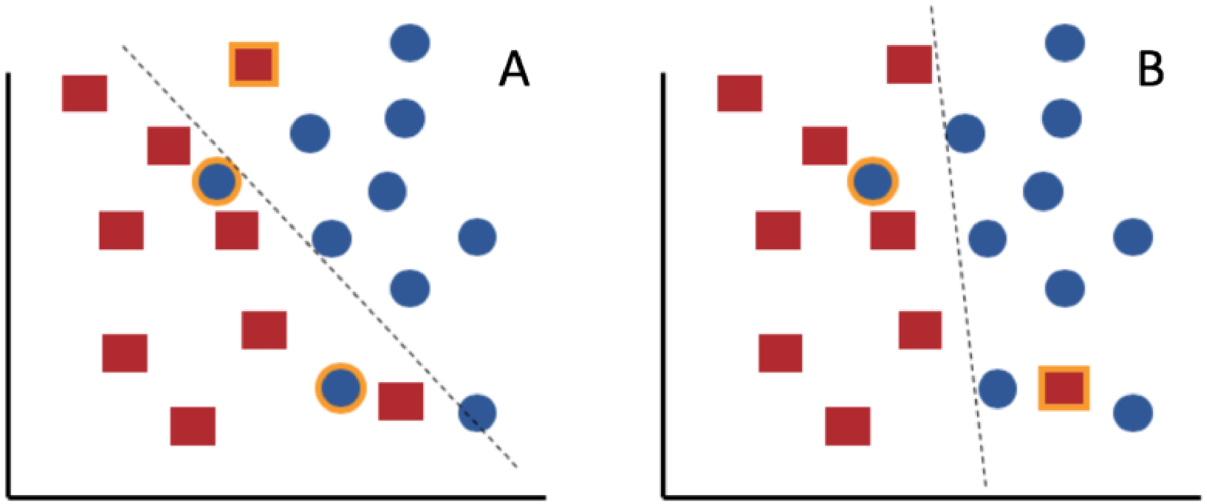
$$R^2 = \frac{explained\ variance}{total\ variance} = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$



Model selection and regularization

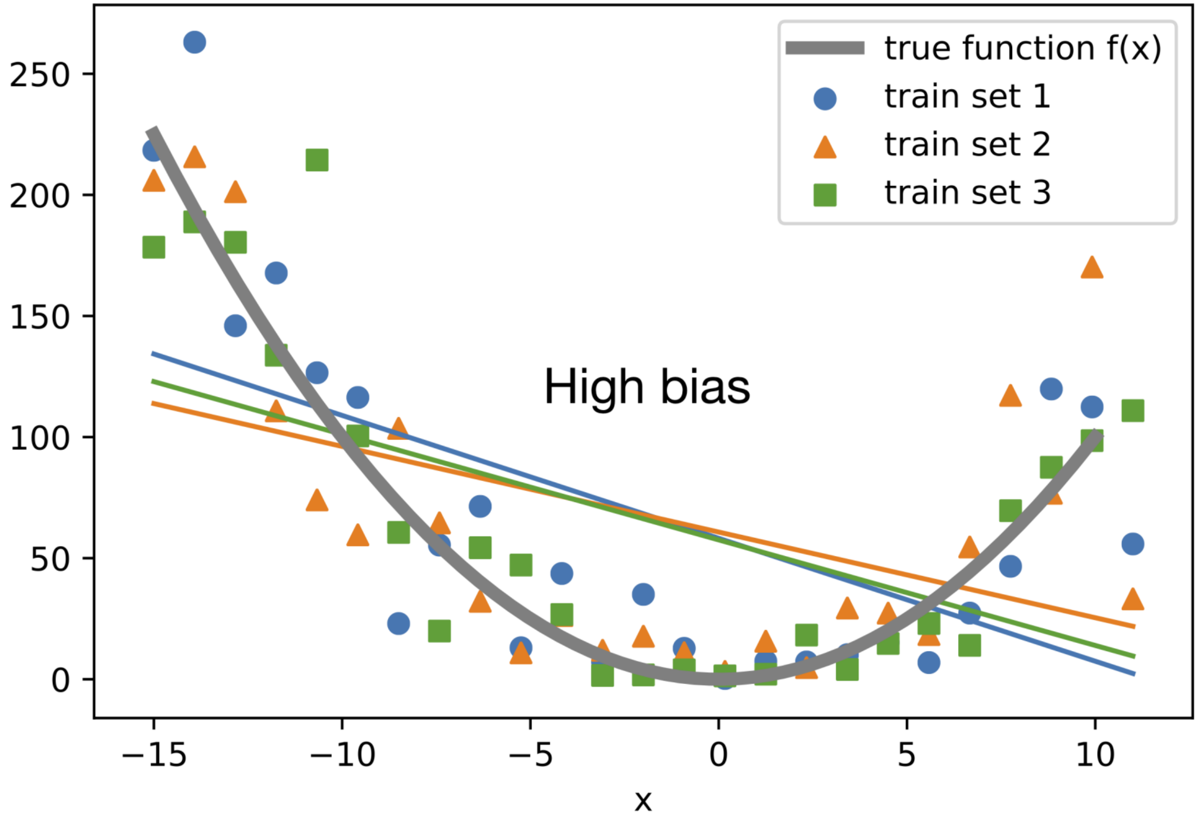
Model selection criteria

- Maximize accuracy? (i.e., minimize error rate)
  - Error rate =1 – accuracy
- Which one?



Bias

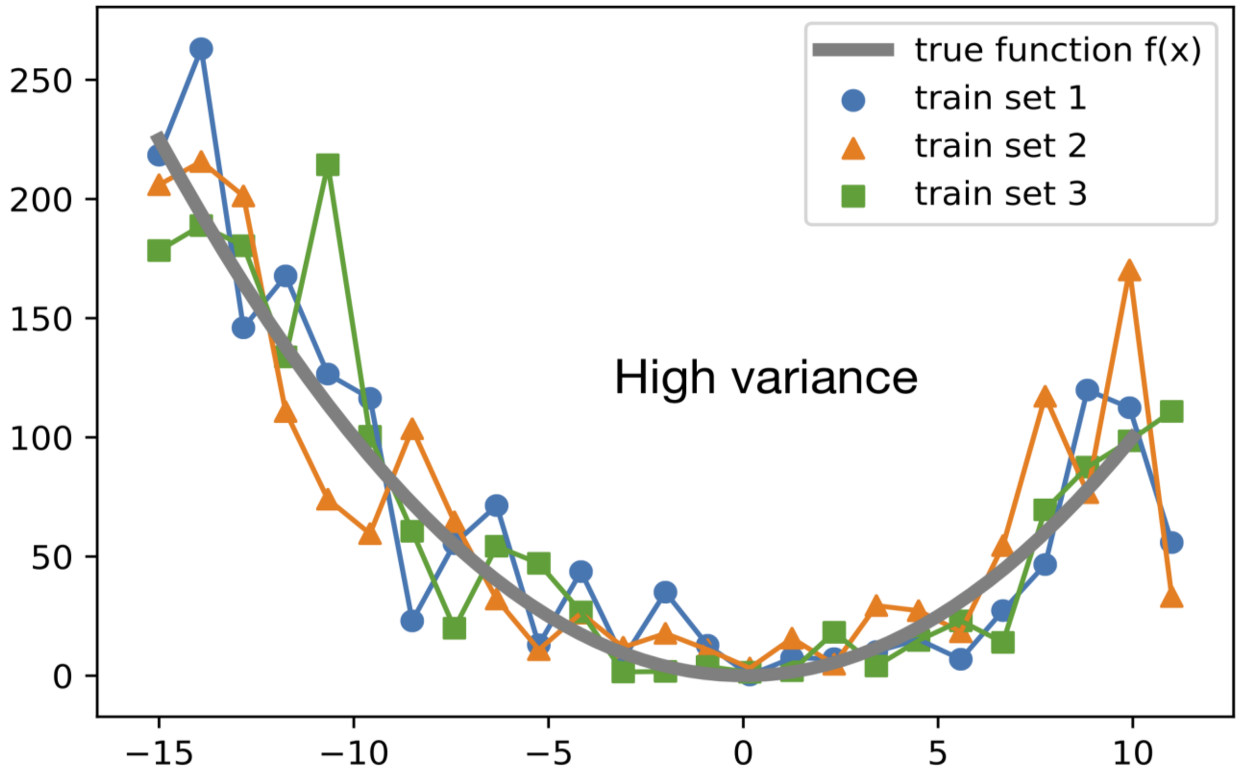
- Bias= $y - \mathrm{E}[\hat{y}]$



- High bias → underfitting

Variance

- Variance= $\mathrm{E}[(\hat{y} - \mathrm{E}[\hat{y}])^2]$



- High variance → overfitting

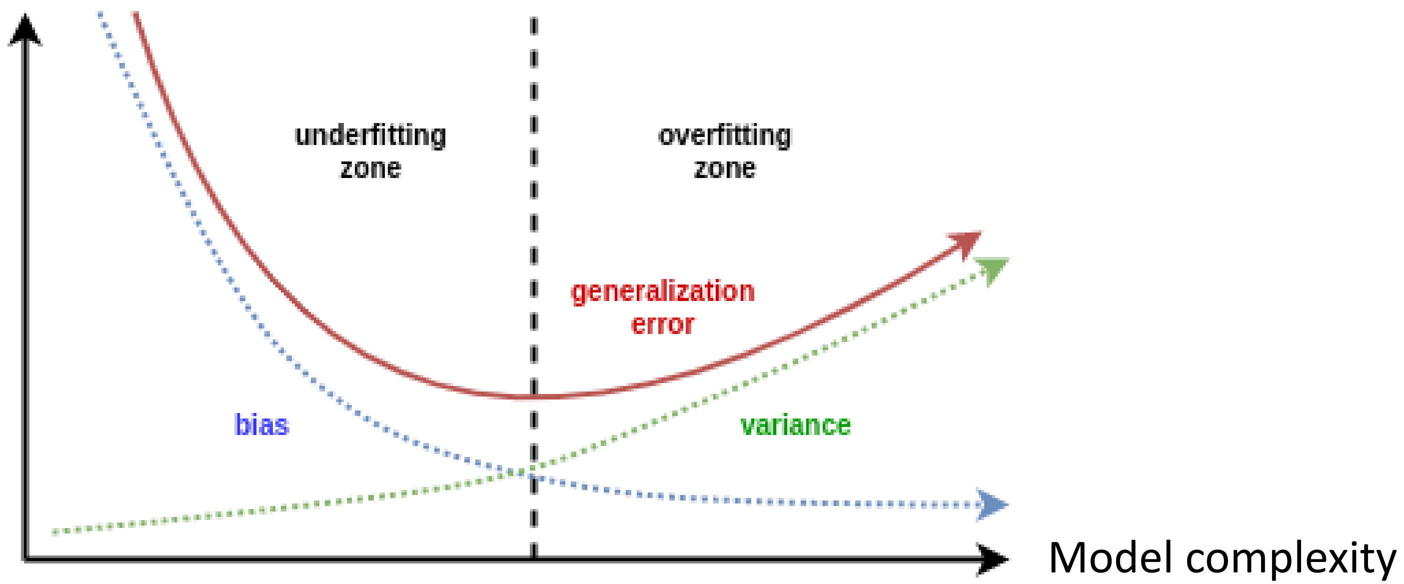
Bias vs. Variance

- Bias – variance decomposition

$$E[S] = E[(y - \hat{y})^2]$$

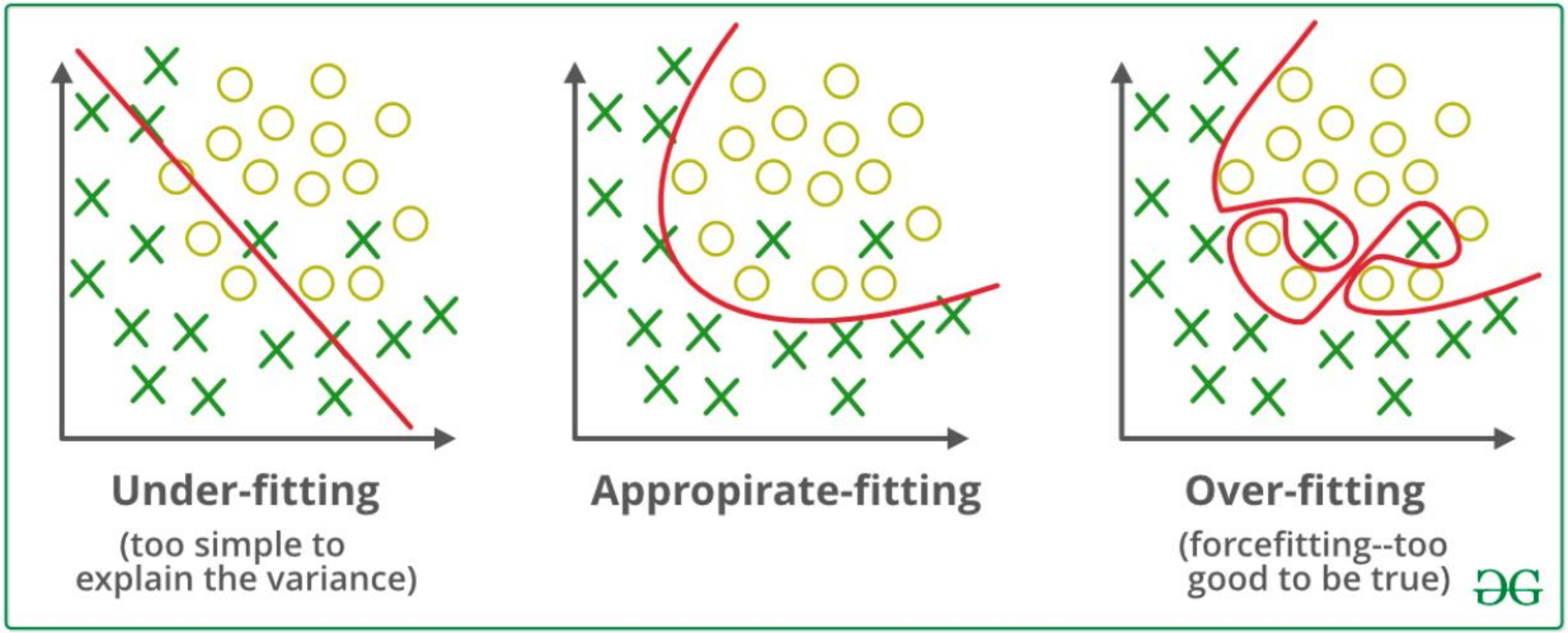
$$\begin{aligned} E[(y - \hat{y})^2] &= (y - E[\hat{y}])^2 + E[(E[\hat{y}] - \hat{y})^2] \\ &= [\mathrm{Bias}]^2 + \mathrm{Variance}. \end{aligned}$$

- Bias – variance trade-off



Over-fitting and under-fitting

- Fitting



Occam's Razor

- Principle of Occam's Razor
  - Suppose there exist two explanations for an occurrence.
  - The one that requires the least assumptions is usually correct.

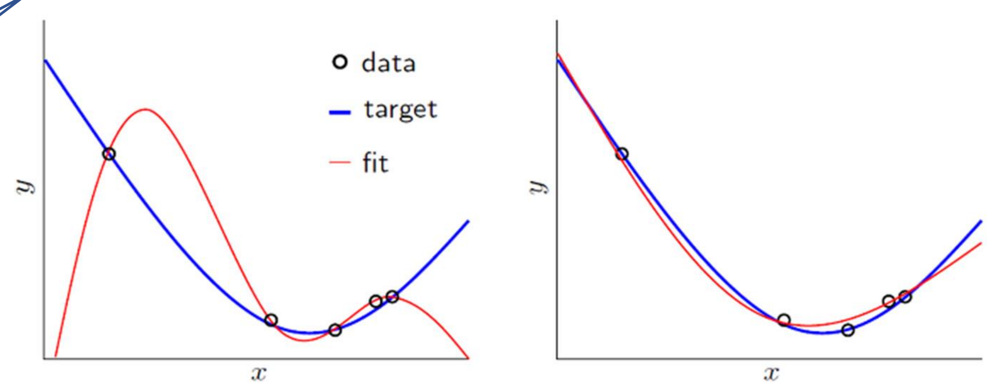
How regularization works

- Regularization
  - Add a penalty term of the parameters to prevent the model from overfitting the data
- Recall empirical risk minimization(ERM):
  - $f = \mathrm{argmin}_{h \in H} \hat{R}(f)$
  - It can be over-optimized (overfitting)
- With regularization

Complexity of f

  - $f = \mathrm{argmin}_{f \in F} \hat{R}(f) + \lambda \Omega(f)$

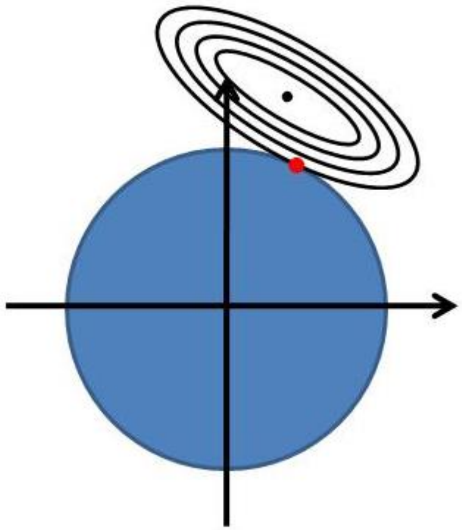
Regularization parameter



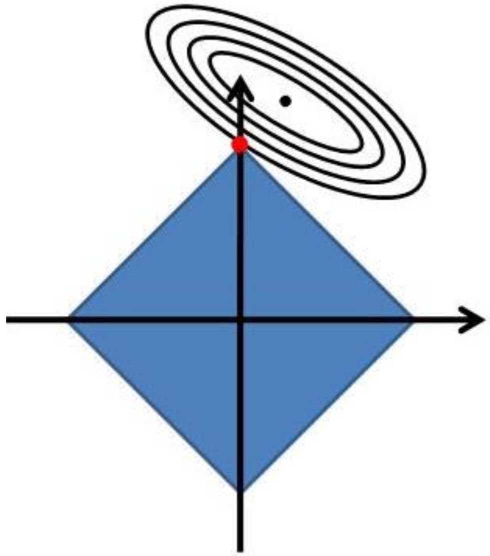
(a) without regularization (b) with regularization

L1-norm and L2-norm regularization

- L2-norm (Ridge):
  - $\Omega(f = ax + b) = a^2 + b^2$

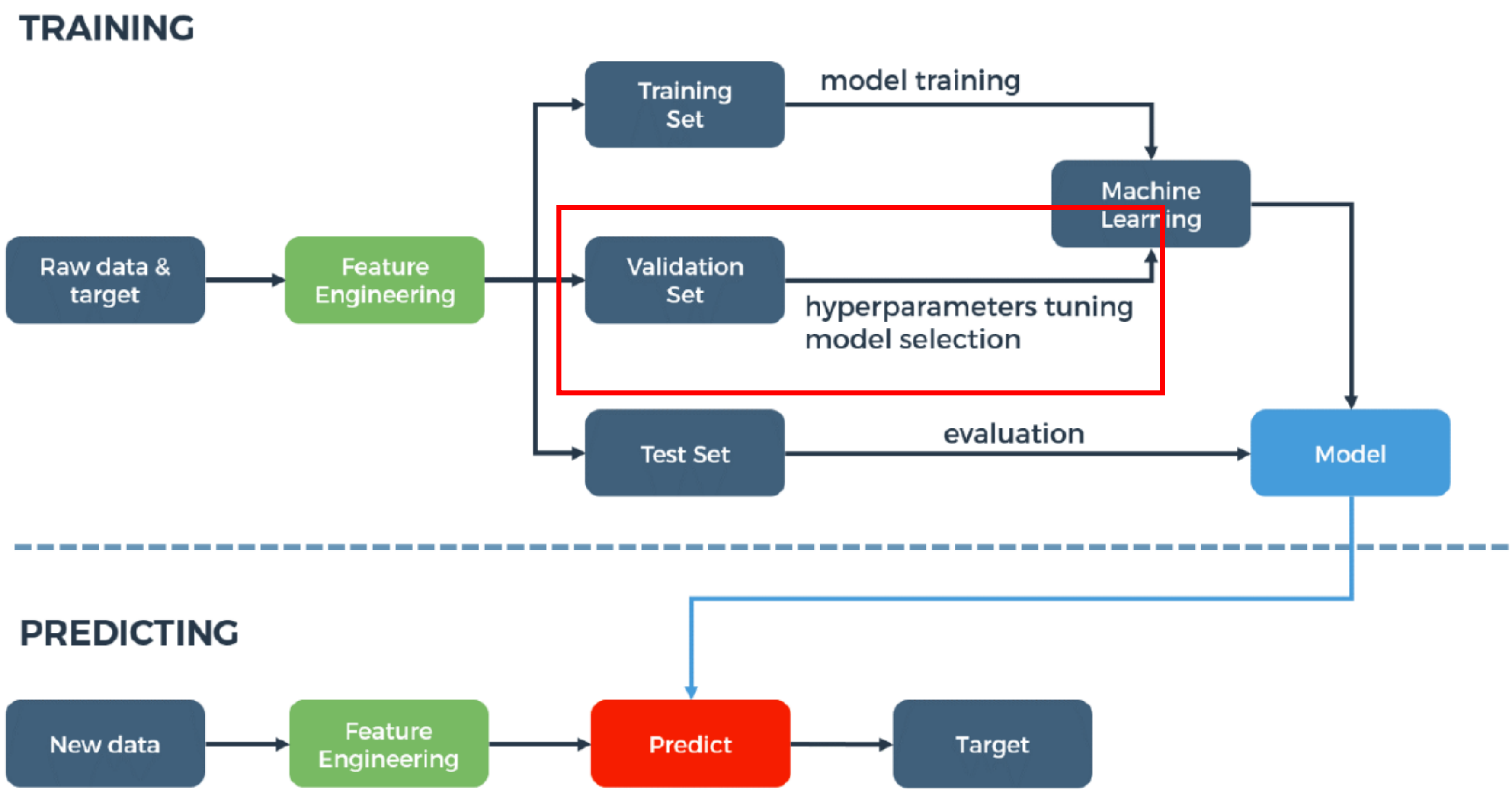


- L1-norm (Lasso):
  - $\Omega(f = ax + b) = |a| + |b|$



Cross validation

Supervised learning process



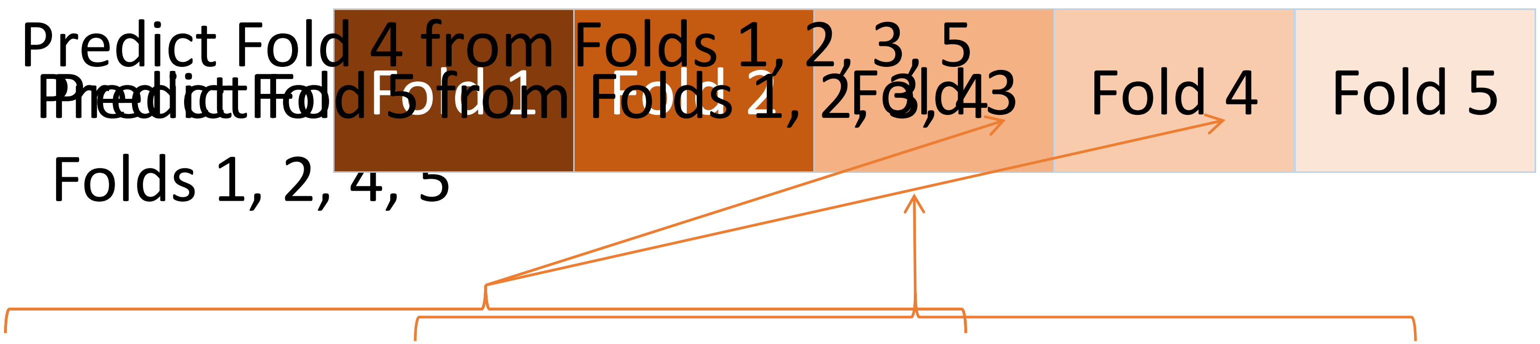
- Basic assumption: there exist the same patterns across training, test and new data

k-fold Cross Validation

- k-fold Cross Validation
  - Given the **training** set, split into k pieces ("folds")
  - Use $(k-1)$ folds to estimate a model, and test model on remaining one fold (which acts as a validation set) for each candidate parameter value
  - Repeat for each of the k folds
  - For each candidate parameter value, average accuracy over the k folds, or validation sets

k-fold Cross-Validation Graphically

- Assume five folds (k = 5)



- Continue to predict Fold 2 and Fold 1…

Lecture 2 wrap-up

- Basics of supervised learning
  - ✓Learning process
  - ✓Discriminative models and generative models
  - ✓Machine learning three elements
    - ✓Model
    - ✓Strategy
    - ✓Algorithm
  - ✓Model evaluation
  - ✓Model selection & Regularization
  - ✓Cross validation

Assignment 2

- **Read home-reading-2a**
  - Install R and R studio
  - Run the codes
  - Get familiar with R asap
  - If you have any trouble, send your problems to TA
- Bonus question (**not required**)
  - Prove Generalization error bound
  - Send your answer to TA (any form, e.g., word, pdf, photo …)
- Due: Apr. 26, 11pm
- TA: Mr. Xiong, xiongym3@mail2.sysu.edu.cn

Next lecture

- Supervised learning
  - Linear regression
  - Logistic regression
  - SVM and kernel
  - Tree models
- Unsupervised learning
  - Clustering
  - PCA
  - EM
- Deep learning
  - Neural networks
  - Convolutional NN
  - Recurrent NN
- Reinforcement learning
  - MDP
  - ADP
  - Deep Q-Network

Questions?

Shan Wang (王杉)
https://wangshan731.github.io/