DS 4400

Machine Learning and Data Mining I
Spring 2021

Alina Oprea

Associate Professor

Khoury College of Computer Science

Northeastern University

February 11 2021

Outline

- Classification
  - K Nearest Neighbors (kNN)
- Cross validation
  - K-fold cross validation
  - Leave-one-out cross validation
- Linear classifiers
- Logistic regression

Gradient Descent vs Closed Form

Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update
for j = 0 … d

Closed form

$$\theta = (X^{\mathsf{T}} X)^{-1} X^{\mathsf{T}} y$$

• Gradient Descent	• Closed Form
+ Linear increase in d and N	+ No parameter tuning
+ Generally applicable	+ Gives the global optimum
- Need to choose α and stopping conditions	- Not generally applicable
- Might get stuck in local optima	- Slow computation

Ridge vs Lasso

- Both methods can be applied to any loss function (regression or classification)
- In both methods, value of regularization parameter λ needs to be adjusted
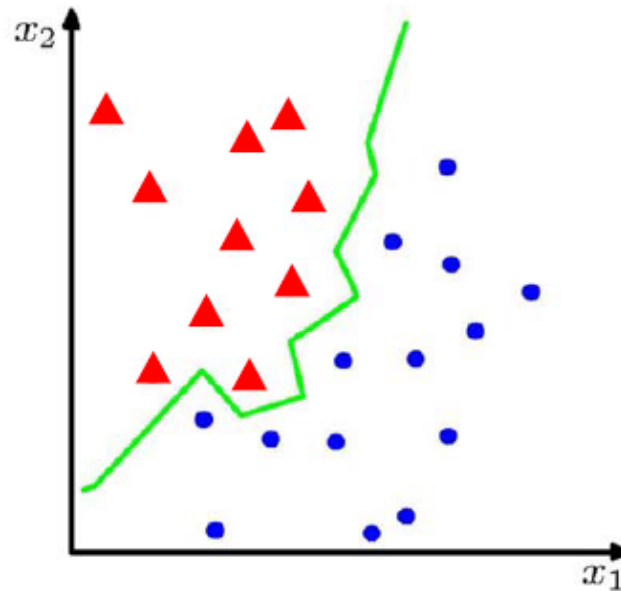- Both reduce model complexity

• Ridge	• Lasso
+ Differentiable objective	- Gradient descent needs to be adapted
+ Gradient descent converges to global optimum	+ Results in sparse model
- Shrinks all coefficients	+ Can be used for feature selection in large dimensions

Classification

Binary or discrete

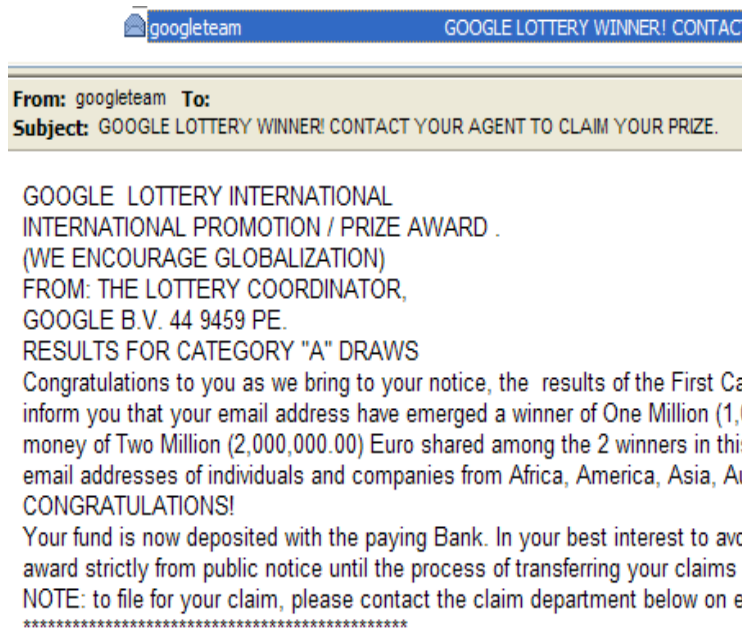- Suppose we are given a training set of N observations

$$\{x_1, \ldots, x_N\} \text{ and } \{y_1, \ldots, y_N\}, x_i \in R^d, y_i \in \{0, 1\}$$

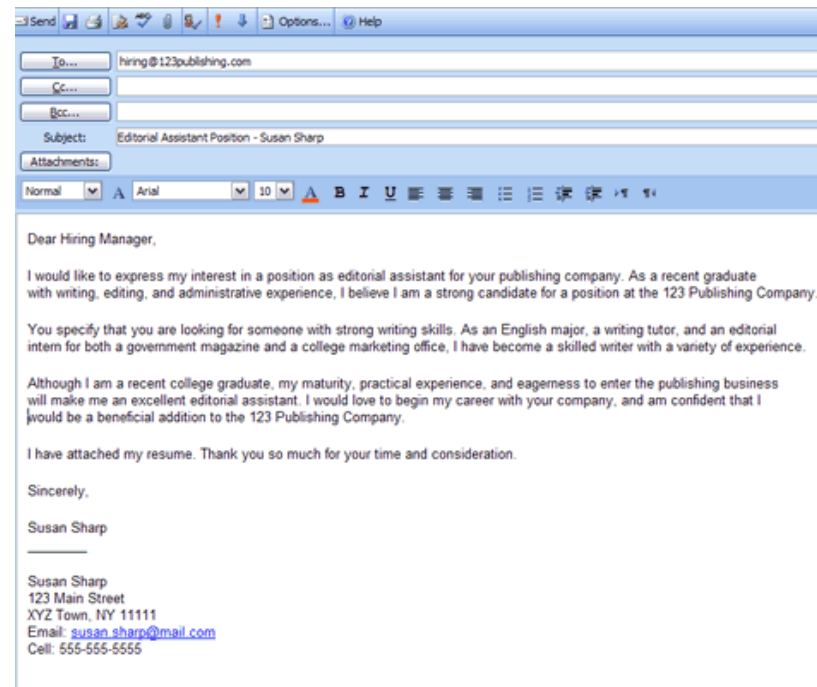- Classification problem is to estimate f(x) from this data such that

$$f(x_i) = y_i$$

Example 1: Binary classification

Classifying spam email

Content-related features
- Use of certain words
- Word frequencies
- Language
- Sentence

Structural features
- Sender IP address
- IP blacklist
- DNS information
- Email server
- URL links (non-matching)

Binary classification: SPAM or HAM

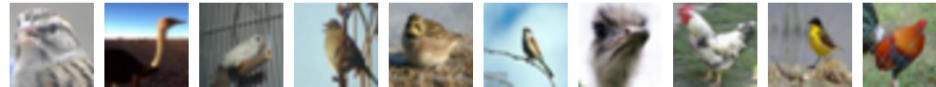# Example 2: Multi-class classification
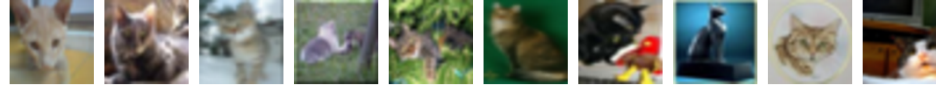
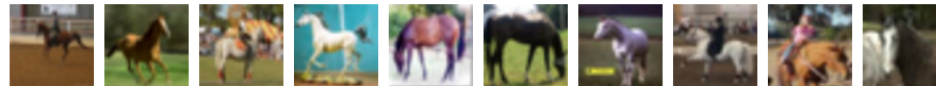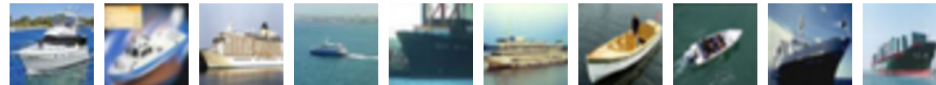Image classification

airplane
automobile
bird
cat
deer
dog
frog
horse
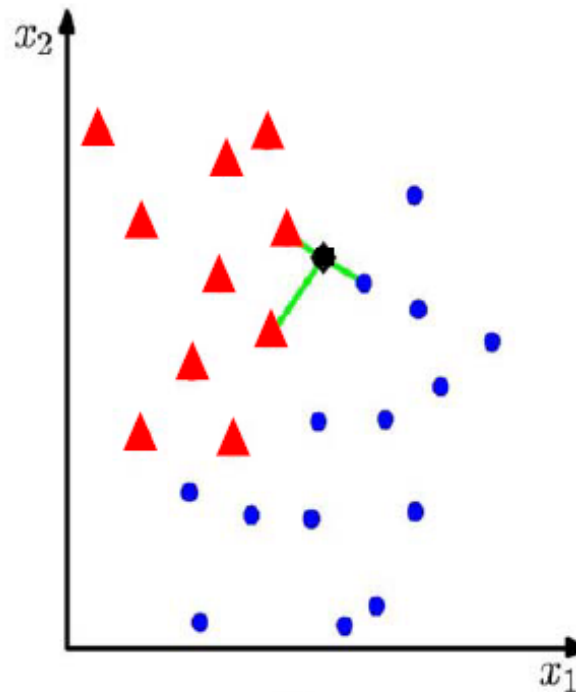ship
truck

Multi-class classification

K Nearest Neighbour (K-NN) Classifier

Algorithm

- For each test point, x, to be classified, find the K nearest samples in the training data

- Classify the point, x, according to the majority vote of their class labels

e.g. K = 3

• applicable to multi-class case

Distance Metrics

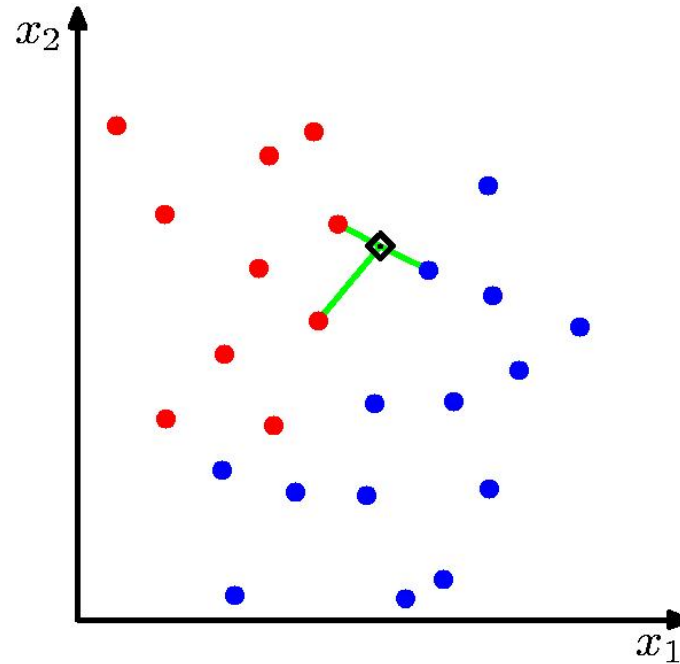- Euclidean Distance $\sqrt{\left(\sum_{i=1}^{k}(x_i - y_i)^2\right)}$

- Manhattan Distance $\sum_{i=1}^{k}|x_i - y_i|$

- Minkowski Distance $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}}$
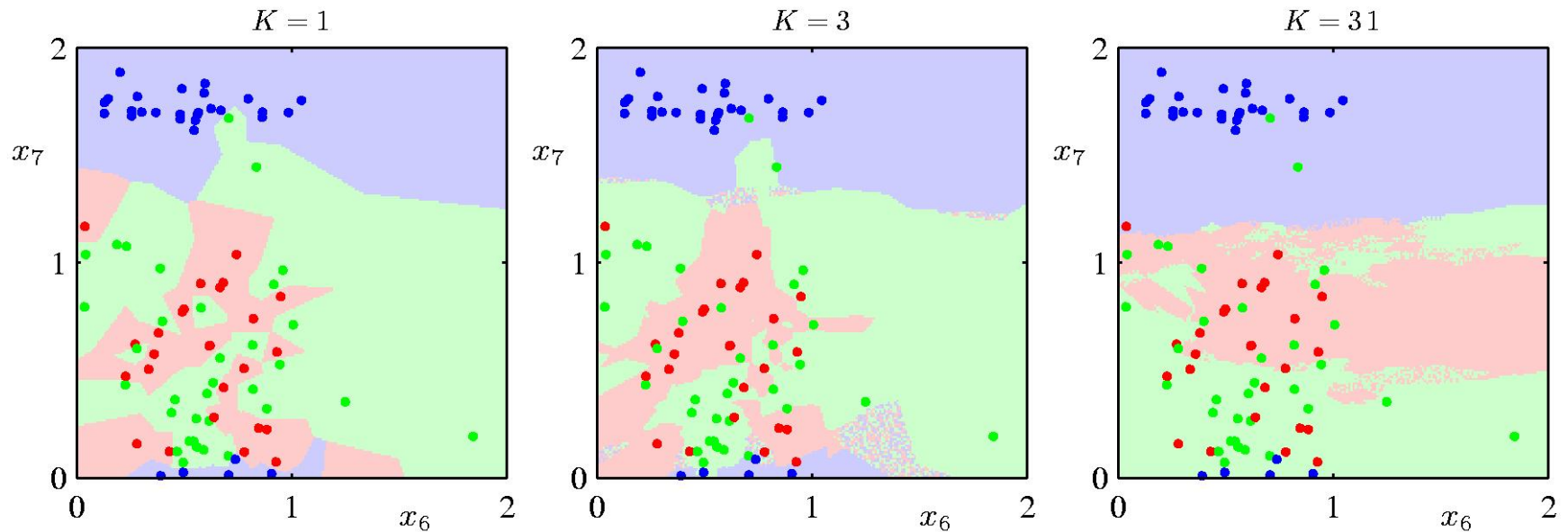
kNN

- Algorithm (to classify point x)
  - Find k nearest points to x (according to distance metric)
  - Perform majority voting to predict class of x
- Properties
  - Does not learn any model in training!
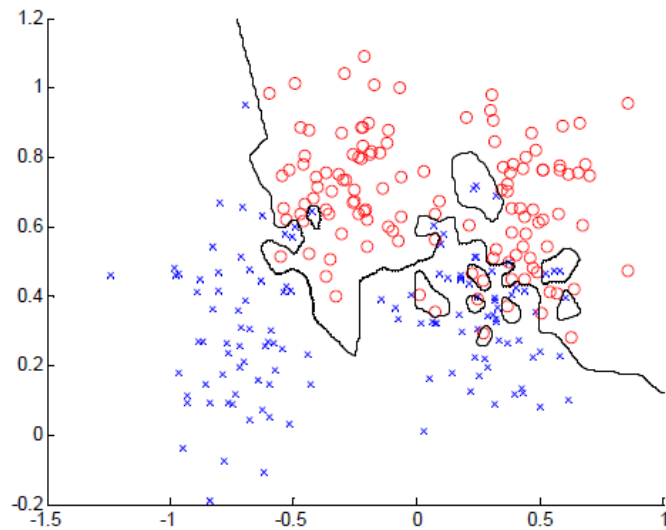 - Instance learner (needs all data at testing time)

K-Nearest-Neighbours for Multi-class Classification

Vote among multiple classes

K = 1

Training data

Testing data
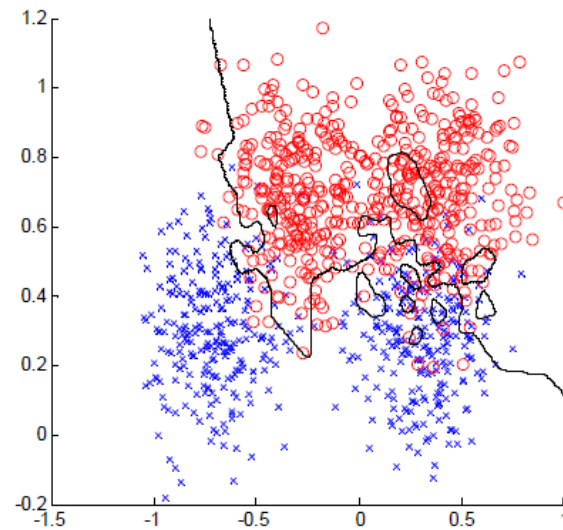
error = 0.0

error = 0.15

How to choose k (hyper-parameter)?

K = 3
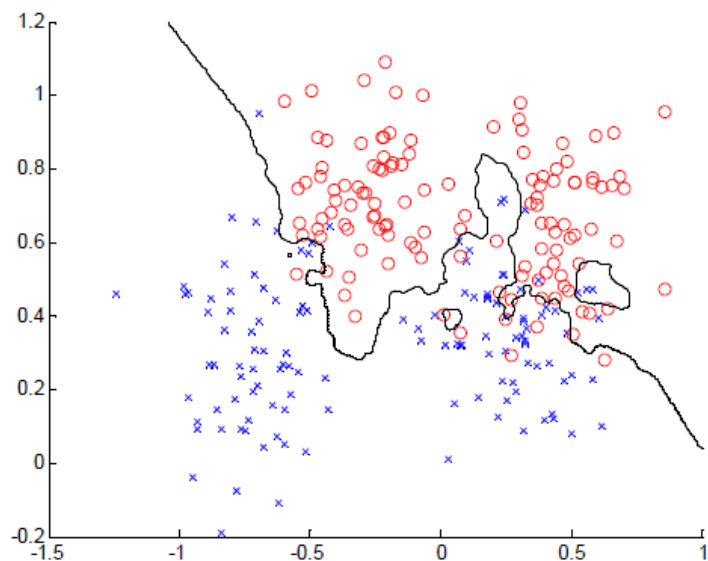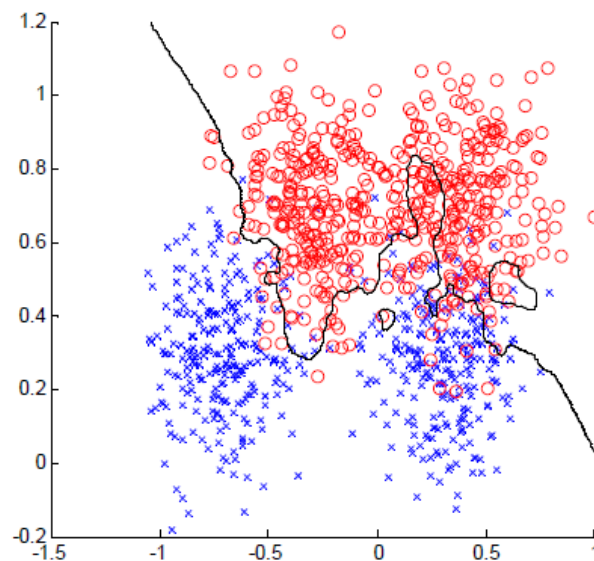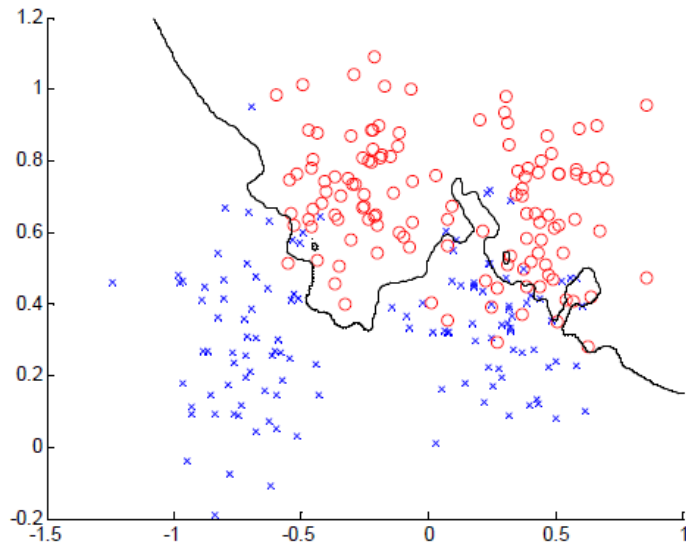
Training data

Testing data

error = 0.0760

error = 0.1340

How to choose k (hyper-parameter)?

K = 7

Training data

error = 0.1320

Testing data
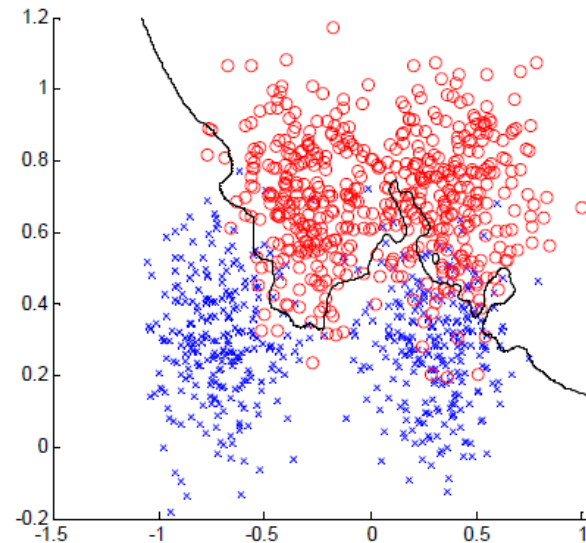
error = 0.1110

How to choose k (hyper-parameter)?

Bias-Variance Tradeoff for kNN

K decreases
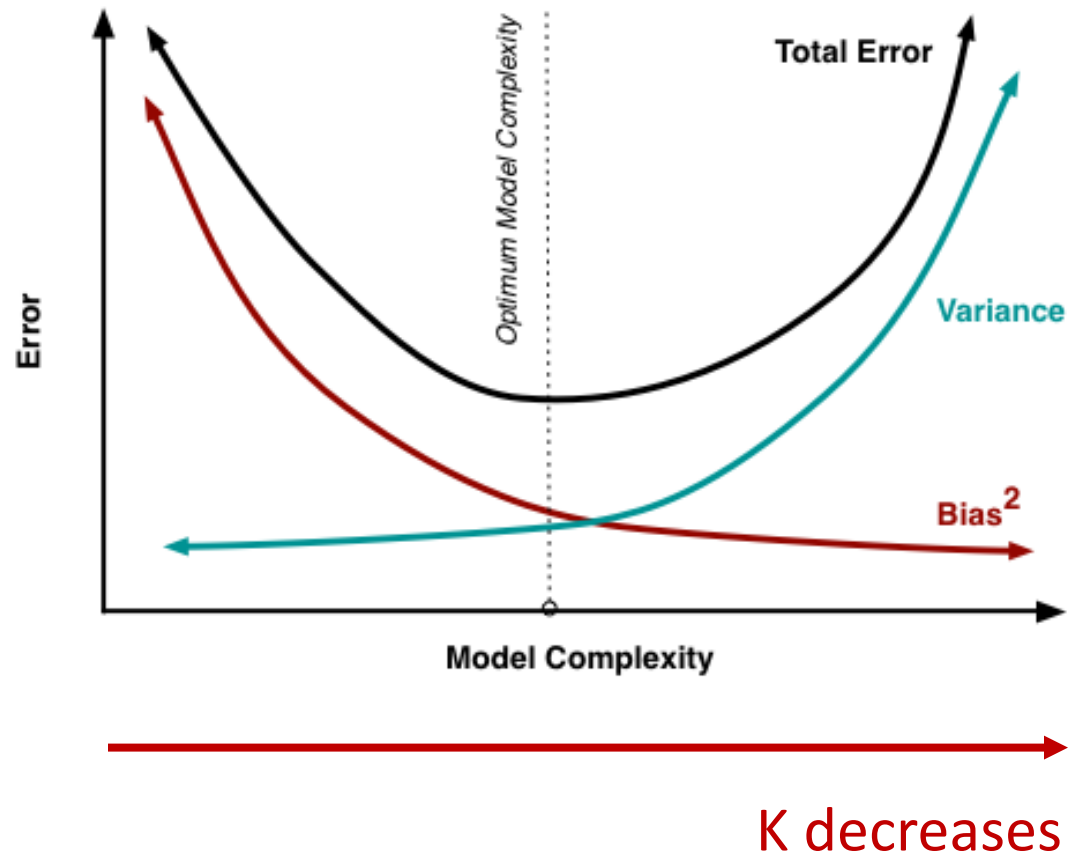
How Overfitting Affects Prediction
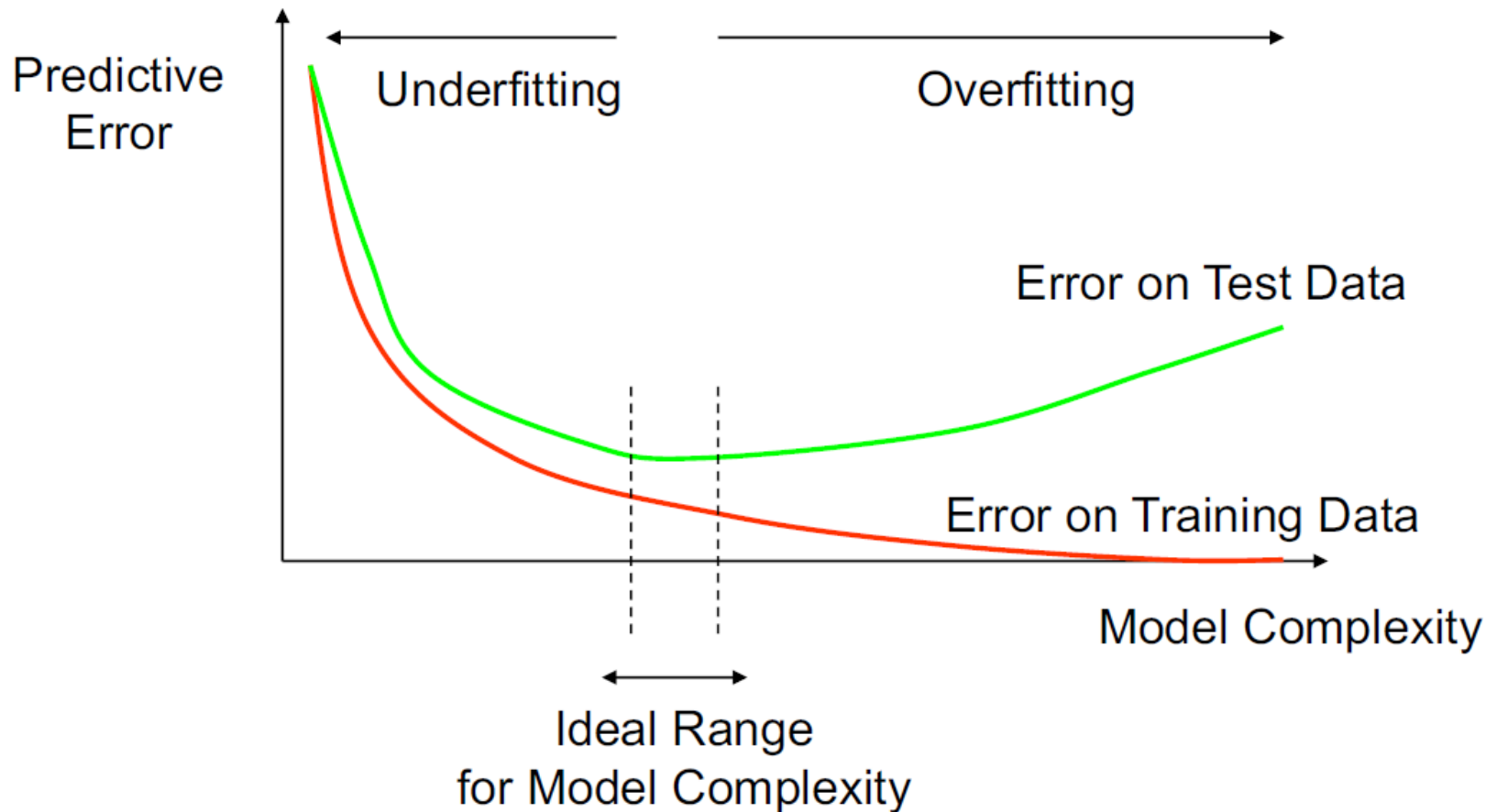
How can we avoid over-fitting without having
access to testing data?

Cross Validation

As K increases:
- Classification boundary becomes smoother
- Training error can increase

Choose (learn) K by cross-validation
- Split training data into training and validation
- Hold out validation data and measure error on this

K=1

| Train set 1 | Validation set 1 | Error |

...

| Train set 10 | Validation set 10 | Error |

Avg Validation Error

K=7

| Train set 1 | Validation set 1 | Error |

...

| Train set 10 | Validation set 10 | Error |

Avg Validation Error
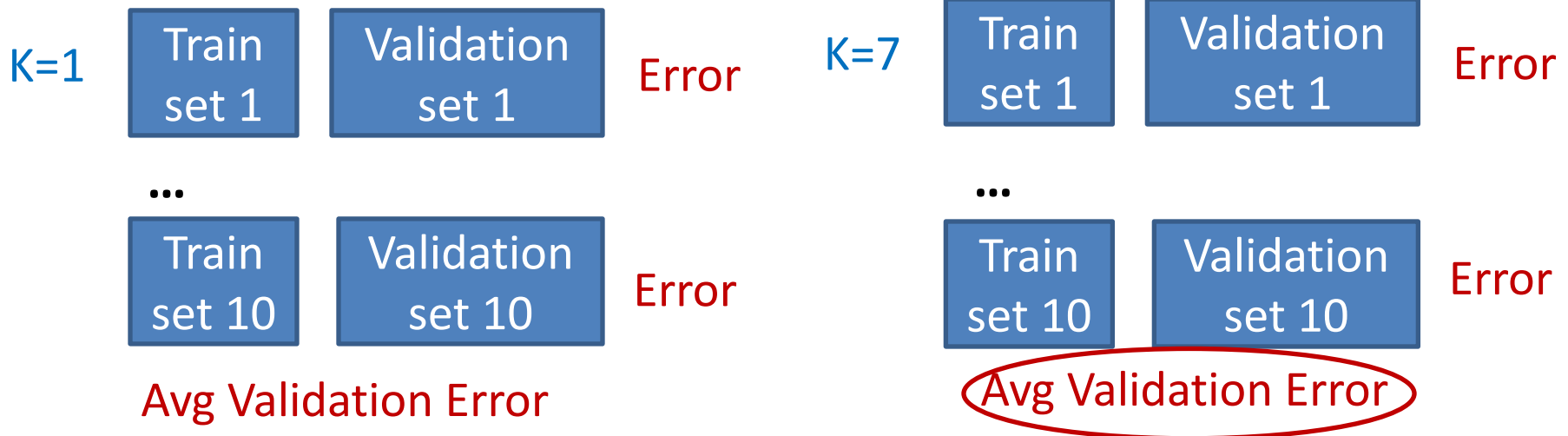
Minimum error!

Cross Validation

Training Data

1st Partition
Validation Set
Training Data

2nd Partition
Training Data
Validation Set
Training Data

kth Partition
Training Data
Validation Set

Test Data

Compute error metrics in each fold
Average error across folds
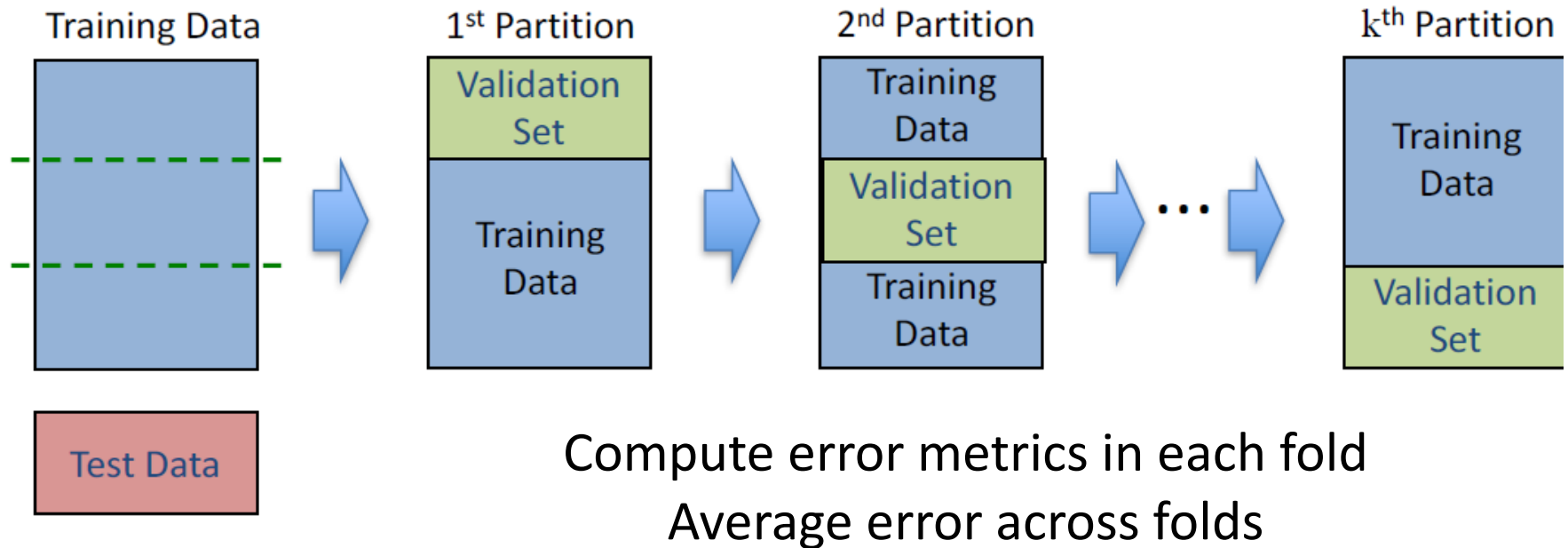
1. k-fold CV

- Split training data into k partitions (folds) of equal size
- Pick the optimal value of hyper-parameter according to error metric averaged over all folds

Cross Validation
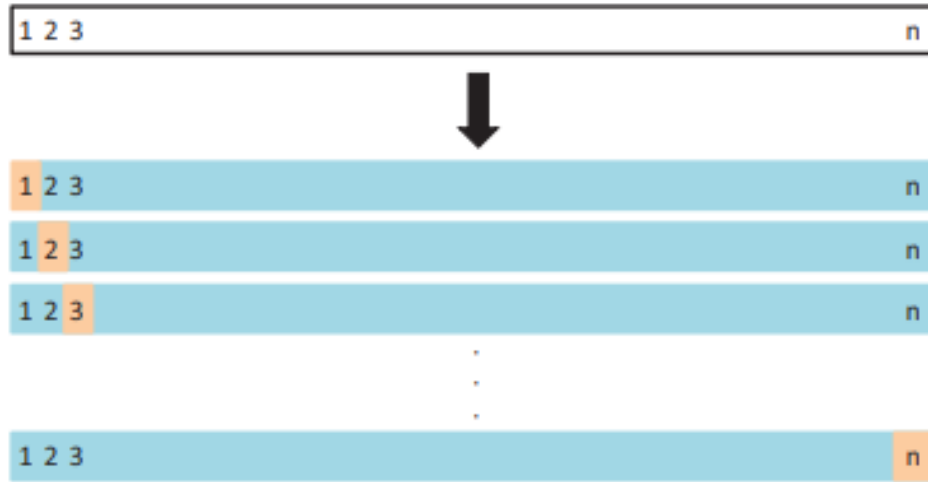
2. Leave-one-out CV (LOOCV)

– k=n (validation set only one point)

- Pros: Less bias
- Cons: More expensive to implement, higher variance
- Recommendation: perform k-fold CV with k=5 or k=10

Cross-Validation Takeaways

- General method to estimate performance of ML model at testing and select hyper-parameters
  - Improves model generalization
  - Avoids overfitting to training data
- Techniques for CV: k-fold CV and LOOCV
- Compare to regularization
  - Regularization works when training with GD
  - Cross-validation can be used for hyper-parameter selection
  - The two methods can be combined

Linear classifiers

- A **hyperplane** partitions space into 2 half-spaces
  - Defined by the normal vector $\boldsymbol{\theta} \in \mathbb{R}^{d+1}$
    - $\boldsymbol{\theta}$ is orthogonal to any vector lying on the hyperplane
  - Assumed to pass through the origin
    - This is because we incorporated bias term θ_0 into it by $x_0 = 1$

- Consider classification with +1, -1 labels …
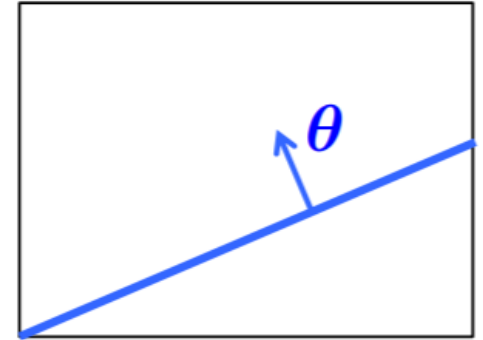
Linear Classifiers

- **Linear classifiers**: represent decision boundary by hyperplane

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \quad \boldsymbol{x}^{\mathsf{T}} = \begin{bmatrix} 1 & x_1 & \dots & x_d \end{bmatrix}$$
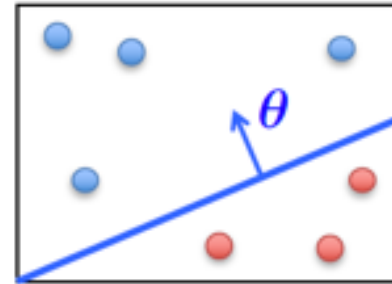
$h_\theta(x) = f(\theta^T x)$ linear function
- If $\theta^T x > 0$ classify "Class 1"
- If $\theta^T x < 0$ classify "Class 0"

All the points x on the hyperplane satisfy: $\theta^T x = 0$

Linear vs Non-Linear Classifiers
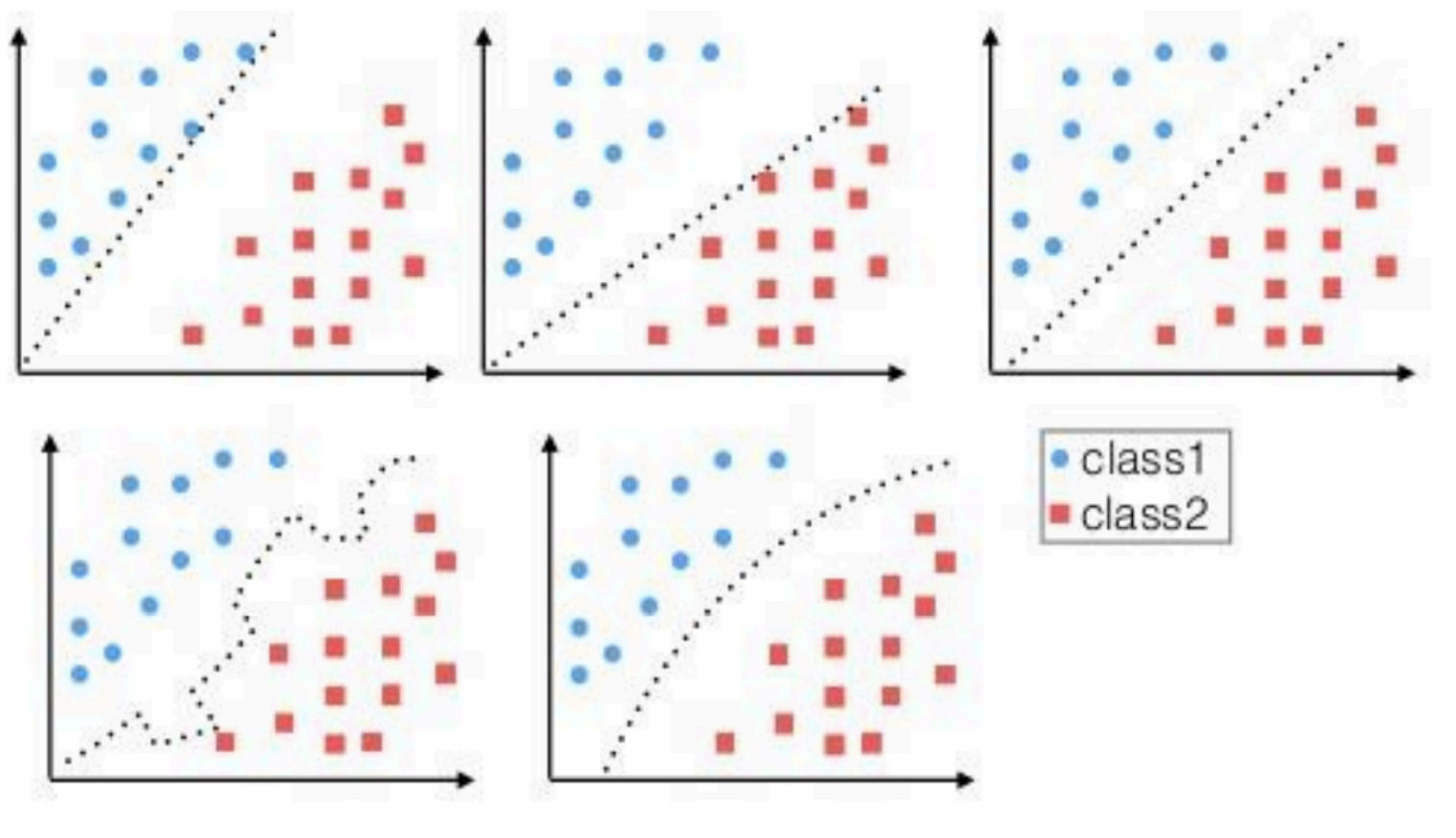
Classification Based on Probability

- Instead of just predicting the class, give the *probability of the instance being in that class*
  - Learn $P(Y|X)$
- Consider binary classifier with classes 0 and 1
  - $P(Y = 1|X) + P(Y = 0|X) = 1$
  - Sufficient to learn $P(Y = 1|X)$
- Advantages: interpretability and confidence of output

Logistic Regression

- ## Setup
  - Training data: $\{x_i, y_i\}$, for $i = 1, \dots, N$
  - Labels: $y_i \in \{0,1\}$
- ## Goals
  - Learn $P(Y = 1 | X = x)$
- ## Highlights
  - Probabilistic output
  - At the basis of more complex models (e.g., neural networks)
  - Supports regularization (Ridge, Lasso)
  - Can be trained with Gradient Descent

Interpretation of Model Output

$h_{\boldsymbol{\theta}}(\boldsymbol{x})$ = estimated $P(Y = 1 | X; \theta)$

Example: Cancer diagnosis from tumor size

$$\boldsymbol{x} = \left[\begin{array}{c} x_0 \\ x_1 \end{array} \right] = \left[\begin{array}{c} 1 \\ \text{tumorSize} \end{array} \right]$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0.7$$

→ Tell patient that 70% chance of tumor being malignant

Note that: $P(Y = 0 | X; \theta) + P(Y = 1 | X; \theta) = 1$

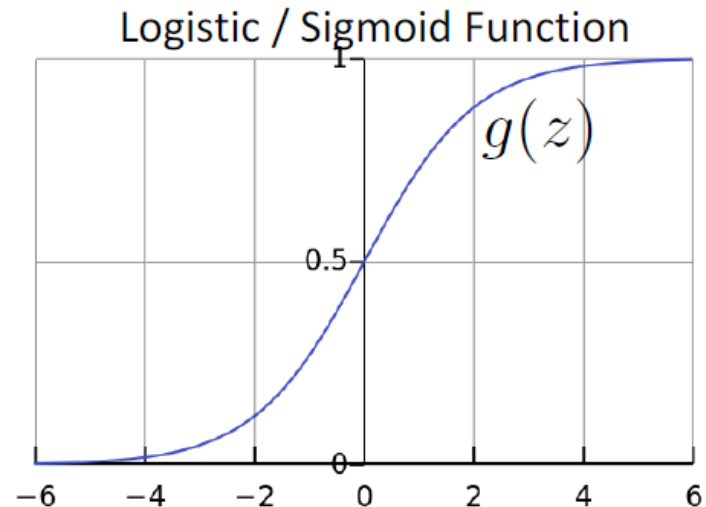Therefore, $P(Y = 0 | X; \theta) = 1 - P(Y = 1 | X; \theta)$

Logistic Regression

- Takes a probabilistic approach to learning discriminative functions (i.e., a classifier)

- $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ should give $P(Y = 1 | X; \theta)$
  - Want $0 \leq h_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$

- Logistic regression model:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^{\mathsf{T}} \boldsymbol{x}\right)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^{\mathsf{T}} \boldsymbol{x}}}$$

Logistic / Sigmoid Function

$g(z)$

LR is a Linear Classifier!

- Predict $Y = 1$ if:
  - $\mathrm{P}[Y = 1 | X = x; \theta] > \mathrm{P}[Y = 0 | X = x; \theta]$
  - $\mathrm{P}[Y = 1 | X = x; \theta] > \frac{1}{2}$

$$\frac{1}{1 + e^{-\theta^T x}} > \frac{1}{2}$$

- Equivalent to:

  - $e^{\theta_0 + \sum_{j=1}^{d} \theta_j x_j} > 1$

  - $\theta_0 + \sum_{j=1}^{d} \theta_j x_j > 0$

Logistic Regression is a linear classifier!

Logistic Regression

$$h_\theta(x) = g(\theta^\mathsf{T} x)$$
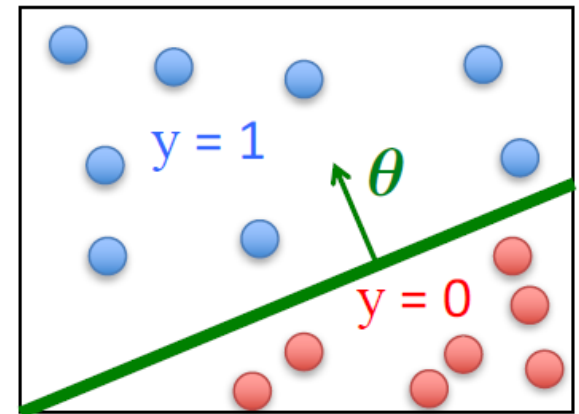
$$g(z) = \frac{1}{1 + e^{-z}}$$

$g(z)$
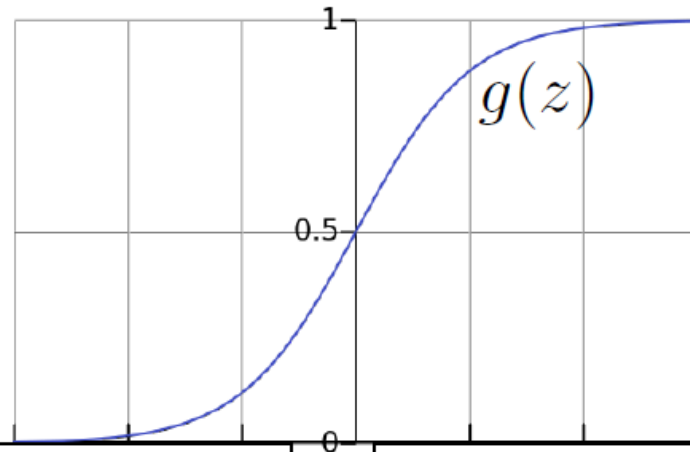
$\theta^\mathsf{T} x$ should be large <u>negative</u> values for negative instances

$\theta^\mathsf{T} x$ should be large <u>positive</u> values for positive instances

- Assume a threshold and...

  - Predict $Y = 1$ if $h_\theta(x) \geq 0.5$
  - Predict $Y = 0$ if $h_\theta(x) < 0.5$

y = 1

θ

y = 0

Logistic Regression is a linear classifier!

Maximum Likelihood Estimation (MLE)

Given training data $X = \{x_1, \dots, x_N\}$ with labels $Y = \{y_1, \dots, y_N\}$

What is the likelihood of training data for parameter θ?

Define likelihood function

$$Max_\theta \, L(\theta) = P[Y|X; \theta]$$

Assumption: training points are independent

$$L(\theta) = \prod_{i=1}^{N} P[Y = y_i | X = x_i; \theta]$$

General probabilistic method for classifier training

Log Likelihood

- Max likelihood is equivalent to maximizing log of likelihood

$$L(\theta) = \prod_{i=1}^{N} P[Y = y_i | X = x_i; \theta]$$

$$\log L(\theta) = \sum_{i=1}^{N} \log P[Y = y_i | X = x_i; \theta]$$

- They both have the same maximum θ_{MLE}

Review

- K nearest neighbors is the first example of classifier
  - Instance learner
- Cross-validation should be performed to
  - Improve generalization and avoid over-fitting
  - Choose hyper parameters (k in kNN)
- Logistic regression is a linear classifier that predicts class probability