Supervised Learning

MSBD 5001, Fall 2018, Lecture 2

Kai Chen, CSE, HKUST

Outline

- Introduction to supervised learning
- A few supervised learning methods
  - Linear Regression/classification
  - Nearest Neighbor Methods
  - Decision Tree
- Potential problems

What is Machine Learning?

- "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T as measured by P, improves with experience E."

  ------- Machine Learning, Tom Mitchell, 1997

Supervised earning

- A part of machine learning
- Given training set {(x,y)}
- Find a good approximation to f: X→Y
- Examples: what are X, Y in these cases?
  - Spam detection
    - X: email, Y: {Spam, not spam}
  - Digit recognition
    - X: Input pixels, Y: {0~9}
  - House price prediction
    - X: House information, Y: Set of price (or the real numbers)

Terminologies

- Given a data point (x, y), x is called feature vector. y is called label
- The dataset given for learning is training data (or training set);
- The dataset to be tested (predicted) is called testing data (or testing set)
  - Cannot use testing data for training

Machine Learning: 3 Steps

- 1. Collect data, extract features
- 2. Determine a model
- 3. Train the model with the data, i.e., “learning”

- Question: how to decide if we trained a good model?

Loss: Measure of Error

- We measure the error using a loss function $L(y, \hat{y})$
- For regression, squared error is often used

$$L(y_i, f(x_i)) = (y_i - f(x_i))^2$$

- This method is called “least squares”
- The *empirical loss* of function f applied to training data is then

$$\frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i)) = \frac{1}{N}\sum_{i=1}^{N} (y_i - f(x_i))^2$$

Loss on Testing Set

- *Empirical loss* is measuring the loss on the training set, i.e., data to train the model.
- We hope to use the model for predicting y values on another dataset, testing set.
- We assume both training set and testing set are i.i.d. (independently identically distributed) from the same distribution D
  - Minimizing loss on training set will make loss on testing set small

Review: Linear Regression

- Recall least square fit for a linear model last lecture:
  - Fit a line $Y = a + bX$ that minimizes the error S
  - $S(a, b) = \sum_{i=1}^{n}(a + bx_i - y_i)^2$. Solve for $\frac{\partial S}{\partial a} = 0, \frac{\partial S}{\partial b} = 0$
  - $b = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^{n}(x_i - \bar{X})^2}$, $a = \bar{Y} - b\bar{X}$, where $\bar{X}$, $\bar{Y}$ denotes the mean of X, Y.
- Squared loss for linear regression has analytical solution: an exact solution you can explicitly derive by analyzing the formula.

Minimizing Loss Functions

- Unlike linear regression, minimizer of loss function may not have analytical solutions, e.g.,
  - Logistic Regression (discuss today)
  - Deep Neural Networks (will discuss later)

- Use optimization methods, e.g., gradient descent to approximate the minimal value of function.

Important Concepts

- Derivatives: The ”speed of change” of a function value at given point
- Gradients: A vector, points to the direction where the function value is changing the fastest.
  - Gradient can be derived by taking partial derivative of every variable. Partial derivative of one variable regard other variables as constants

Example

- What's the partial derivative $\frac{\partial G}{\partial w}$, where $G(x, w) = w^2 + 2xw$?
- Answer: Treat x as a constant, we get

$$\frac{\partial G}{\partial w} = 2w + 2x$$

Gradient Descent Method

- An optimization method to search for minimizer of function (minimizer is where function gets minimal value)

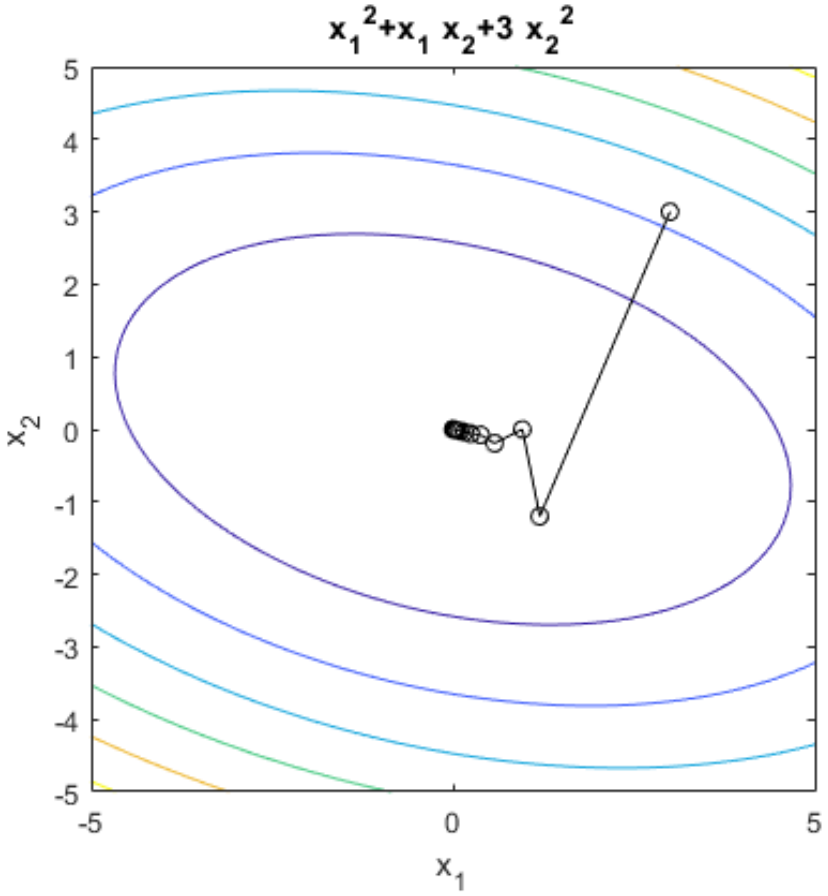
1. For function G, randomly guess an initial value x_0
2. Repeat: $x_{i+1} = x_i - r \times \nabla G(x)$, where ∇ denotes the gradients, reads "nabla". Here r denotes learning rate
3. Until convergence*

* Convergence is a pre-defined rule, e.g., function value is not changing too fast, fixed number of iterations, etc.

Example

- Please use gradient descent method to find the minimal value of $f(x_1, x_2) = x_1^2 + 3x_2^2 + x_1 x_2$. Your initial guess of $(x_1, x_2) = (3,3)$. What's the value of (x_1, x_2) after 3 iterations? Please use a learning rate of $r = 0.2$
- Answer: First we find the gradient $\nabla f(x_1, x_2) = (2x_1 + x_2, x_1 + 6x_2)$, then we do gradient descent step by step:
  - 1st iteration: $(x_1, x_2) = (3,3) - 0.2(2 * 3 + 3, 3 + 6 * 3) = (1.2, -1.2)$
  - 2nd iteration: $(x_1, x_2) = (0.96,0)$
  - 3rd iteration: $(x_1, x_2) = (0.576, -0.192)$

Plotted Gradient Descent



Plotted contour line + gradient descent in matlab: After 3rd iteration, the estimated value is quite closed to the final minimizer.

Example (2)

- $f(x, y, z) = \log(x) + 3yz + 2z^2$ Initial guess (2,2,2), r=0.1, run for 2 steps. Log is "natural log" (base is e).
- Answer:
  - $\nabla f(x, y, z) = (\frac{1}{x}, 3z, 4z + 3y)$
  - (2,2,2) → (1.75, -1, -5) → (1.46, 6.5, 6.5)

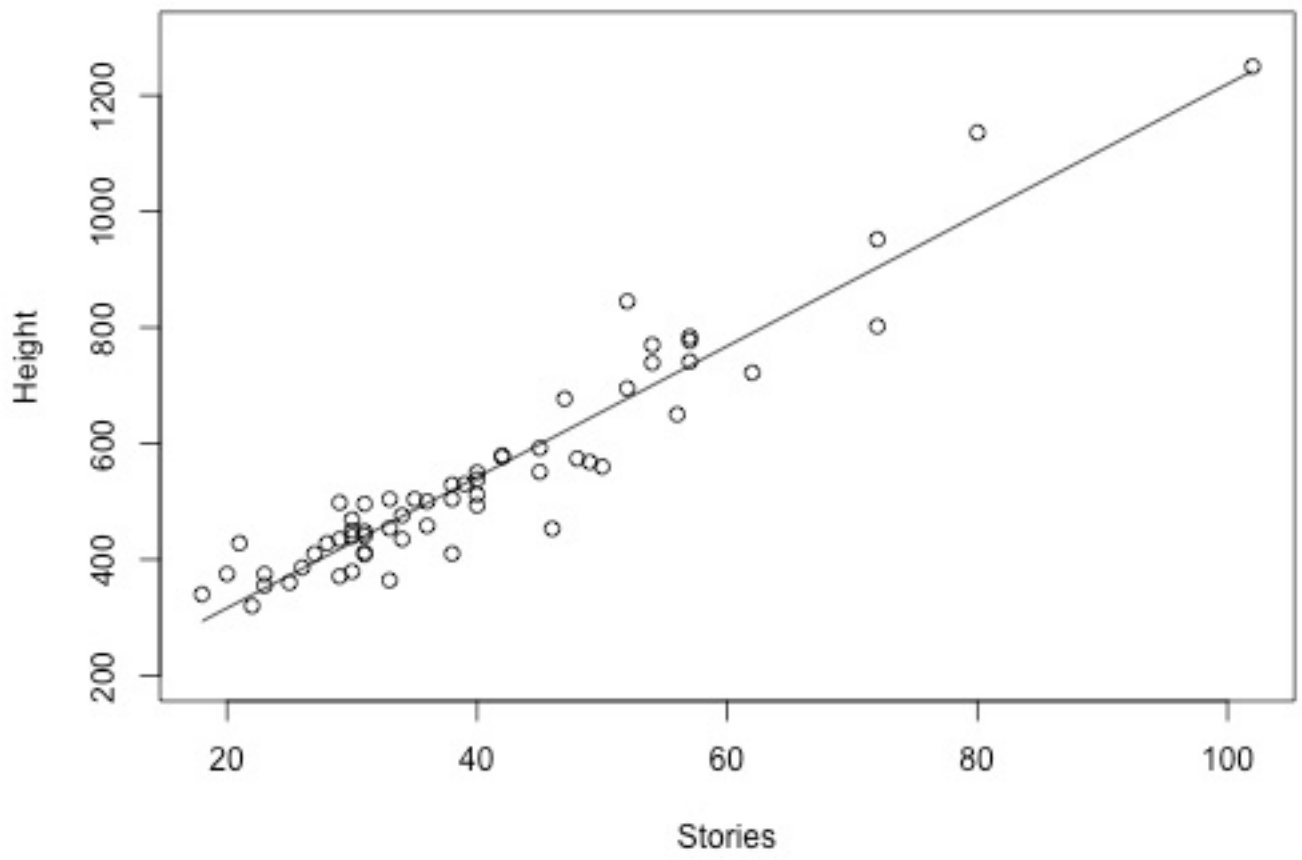
Outline

- Introduction to supervised learning
- A few supervised learning methods
  - Linear Regression/classification
  - Nearest Neighbor Methods
  - Decision Tree
- Potential problems

Linear Regression

- Use a “line” to fit the data points, discussed last lecture.
- Is there a relationship between no. of stories and height of building?
- Example dataset https://onlinecourses.science.psu.edu/stat501/sites/onlinecourses.science.psu.edu.stat501/files/data/bldgstories/index.txt

Example: Stories vs Height



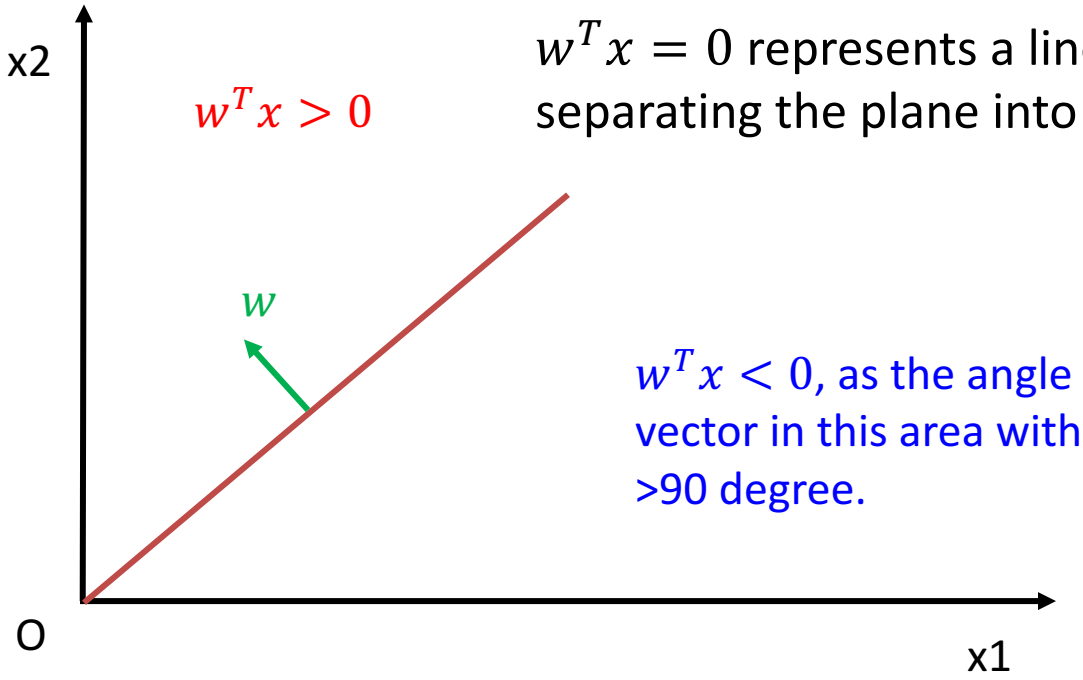
Source: https://onlinecourses.science.psu.edu/stat501/node/257/

Outline

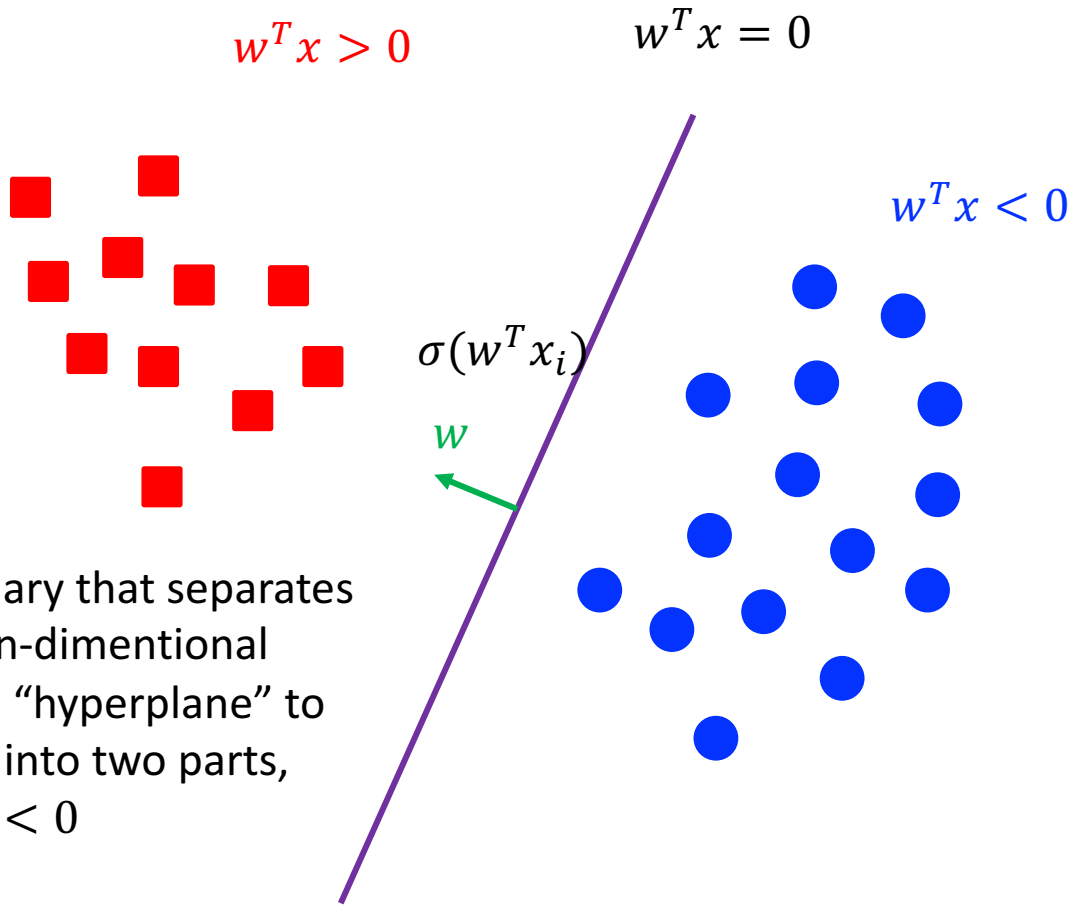
- Introduction to supervised learning
- A few supervised learning methods
  - Linear Regression/classification
  - Nearest Neighbor Methods
  - Decision Tree
- Potential problems

"Lines"

- Let $x = (x_1, x_2)$, w $= (w_1, w_2)$. I.e., x, w are vectors in 2D space.



Linear Classification



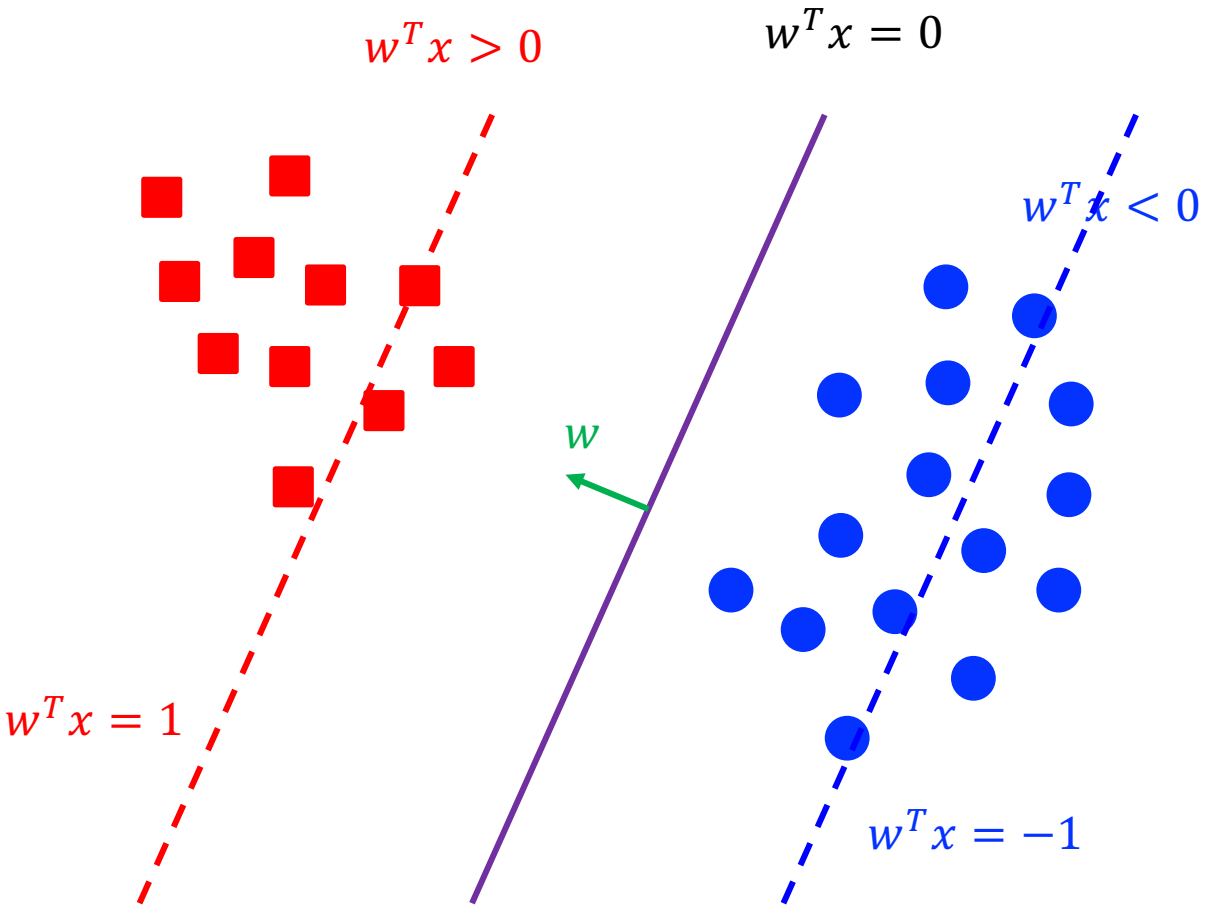
Find a linear boundary that separates the two classes. In n-dimentional space $w^T x = 0$ is a "hyperplane" to separate the space into two parts, $w^T x > 0$ and $w^T x < 0$

Linear Classification

- Consider 2-class classification problems.
- Given training data (x_i, y_i) , where x are input features, and y are labels for the data.
- Can we use methods similar to linear regression?
  - Label y as either 1 (red) or -1 (blue).
  - Find $f_w(x) = w^T x$ that minimizes the loss function:

$$L\big(f_w(x)\big) = \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2$$

Linear Classification



Find a line $w^T x = 0$ such that minimizes the sum of distance

- From red dots to $w^T x = 1$, and
- From blue dots to $w^T x = -1$

How Does It Work?

- Find a line $w^T x = 0$ such that minimizes the sum of distance
  - From red dots to $w^T x = 1$, and
  - From blue dots to $w^T x = -1$
- Doesn't make sense!
  - y are categorical, discrete values, but $w^T x$ are unbounded continuous, numerical values
  - Data points far from the classification line ("outliers") may affect the results

Outliers?

Blue outliers make the model sub-optimal

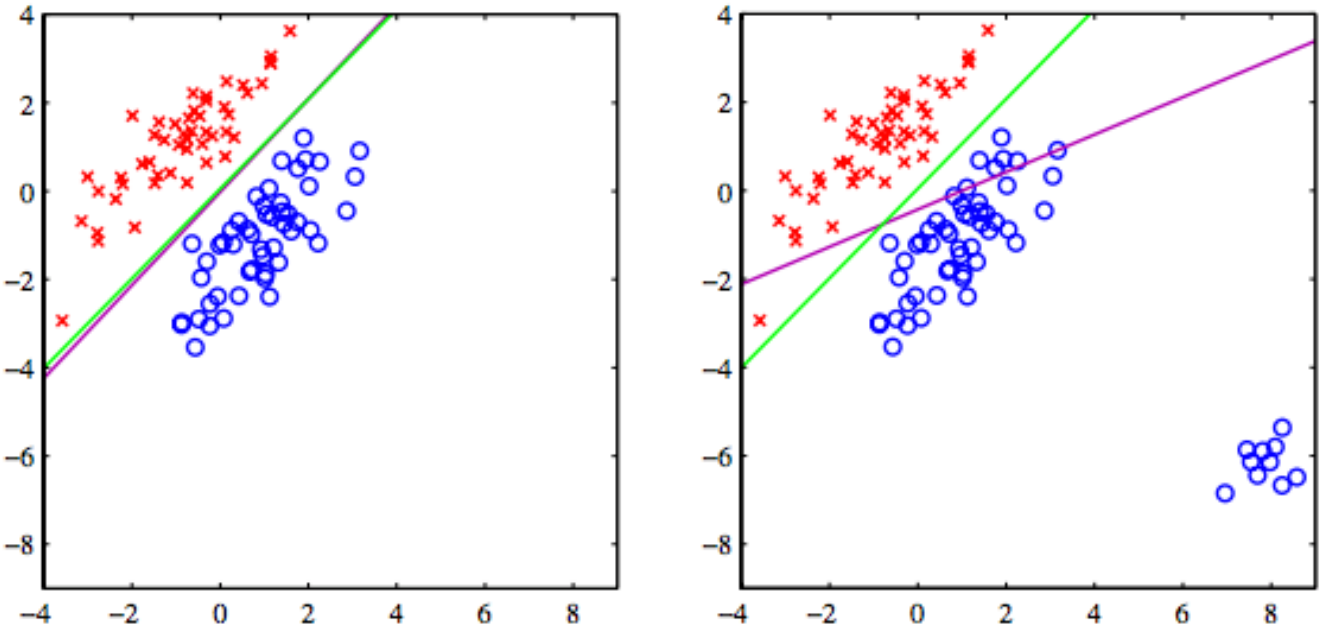


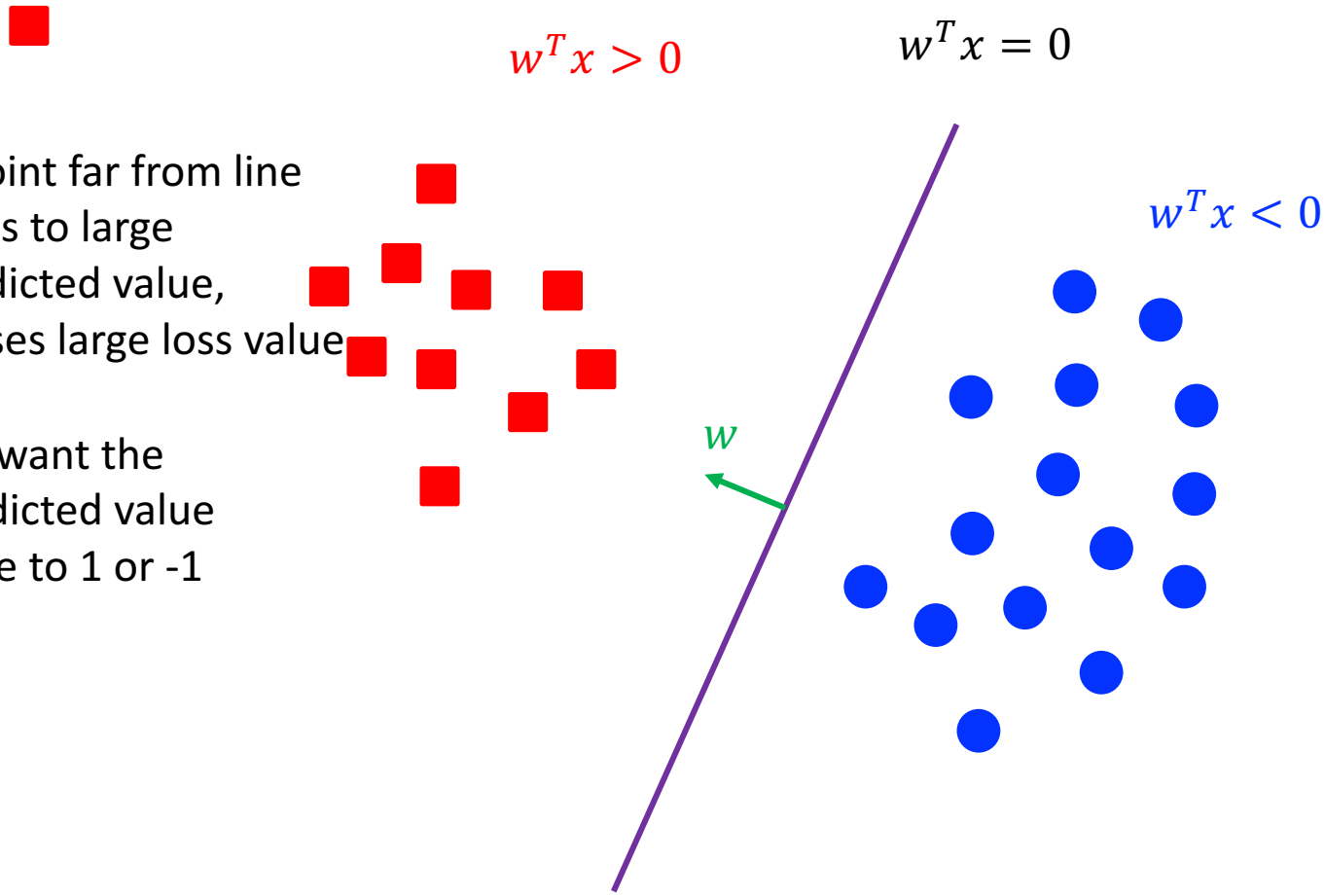
Figure 4.4 The left plot shows data from two classes, denoted by red crosses and blue circles, together with the decision boundary found by least squares (magenta curve) and also by the logistic regression model (green curve), which is discussed later in Section 4.3.2. The right-hand plot shows the corresponding results obtained when extra data points are added at the bottom left of the diagram, showing that least squares is highly sensitive to outliers, unlike logistic regression.

Source: Bishop et al, ”Pattern Recognition and Machine Learning”, Springer 2006

Outlier Explained

- $L\left(f_w(x)\right) = \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2$ has problems
  - y_i are discrete values, 1 or -1.
  - $w^T x_i$ can take very large value, making loss value $(w^T x_i - y_i)^2$ large
  - $w^T x_i = 10000$, $y_i = 1$ → loss value is large even when the prediction is correct (Predicted value $w^T x_i$ is positive)
  - We want the predicted value close to 1 or -1

Outlier Explained



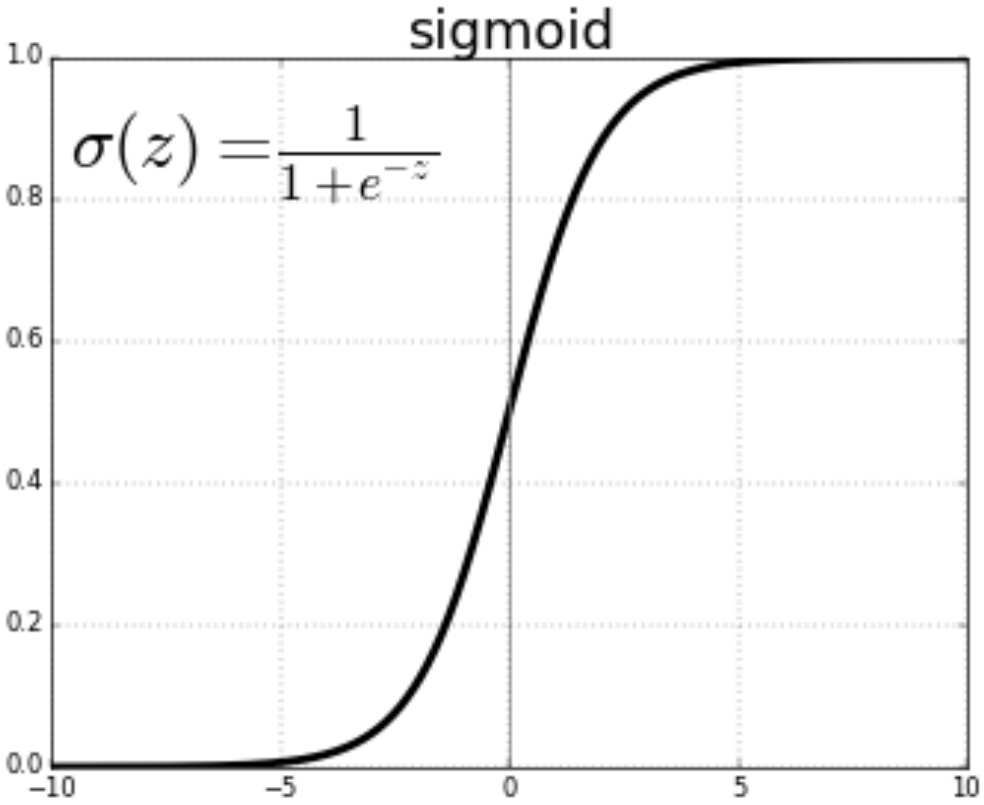
A point far from line leads to large predicted value, causes large loss value

We want the predicted value close to 1 or -1

Preferred Predicted Value

- In later discussions, we relabel -1 (blue) as 0 to simplify the presentation
- We want values close to 1 or 0
  - Directly using $w^T x$ infeasible: values can be large
  - Step functions? ($f(x) = 1\ if\ x >= 0; f(x) = 0$ otherwise)
    - Good, but step functions are not continuous: Computation difficulties.
- In between: Sigmoid function
  - $\sigma(a) = \frac{1}{1+\exp(-a)}$

Sigmoid function



Good properties from visualization:

1. Similar to step functions → Low impact on outliers

2. Continuous → Easy to compute

Some Properties of Sigmoid Function

- Bounded

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \in (0,1)$$

- Symmetric

$$1 - \sigma(a) = \frac{\exp(-a)}{1 + \exp(-a)} = \frac{1}{\exp(a) + 1} = \sigma(-a)$$

- Gradient

$$\sigma'(a) = \frac{\exp(-a)}{(1 + \exp(-a))^2} = \sigma(a)(1 - \sigma(a))$$

Logistic Regression

- Prediction: $\sigma(w^T x_i) > 0.5$, predict as 1, 0 otherwise

- Can use $L\big(f_w(x)\big) = \frac{1}{n}\sum_{i=1}^{n}(\sigma(w^T x_i) - y_i)^2$
  - $(\sigma(w^T x_i) - y_i)^2$ is bounded in [0,1], we want to give large loss value to misclassified data.

Logistic Regression

- Better approach: find w that minimizes

$$-\frac{1}{n}\sum_{y_i=1}\log(\sigma(w^T x_i)) - \frac{1}{n}\sum_{y_i=0}\log\left(1-\sigma(w^T x_i)\right)$$

- If misclassification happens on i-th data with label 1, $\log(\sigma(w^T x_i))$ is very large
  - E.g., $\sigma(w^T x_i) = 0.001$ when $y_i = 1$, misclassification
  - Squared loss: 0.999*0.999=0.998
  - Log loss: $-\log(0.001) = 3$

- No analytical solution, needs gradient descent

Logistic (Green) vs Linear (Purple)

Blue outliers make the model sub-optimal

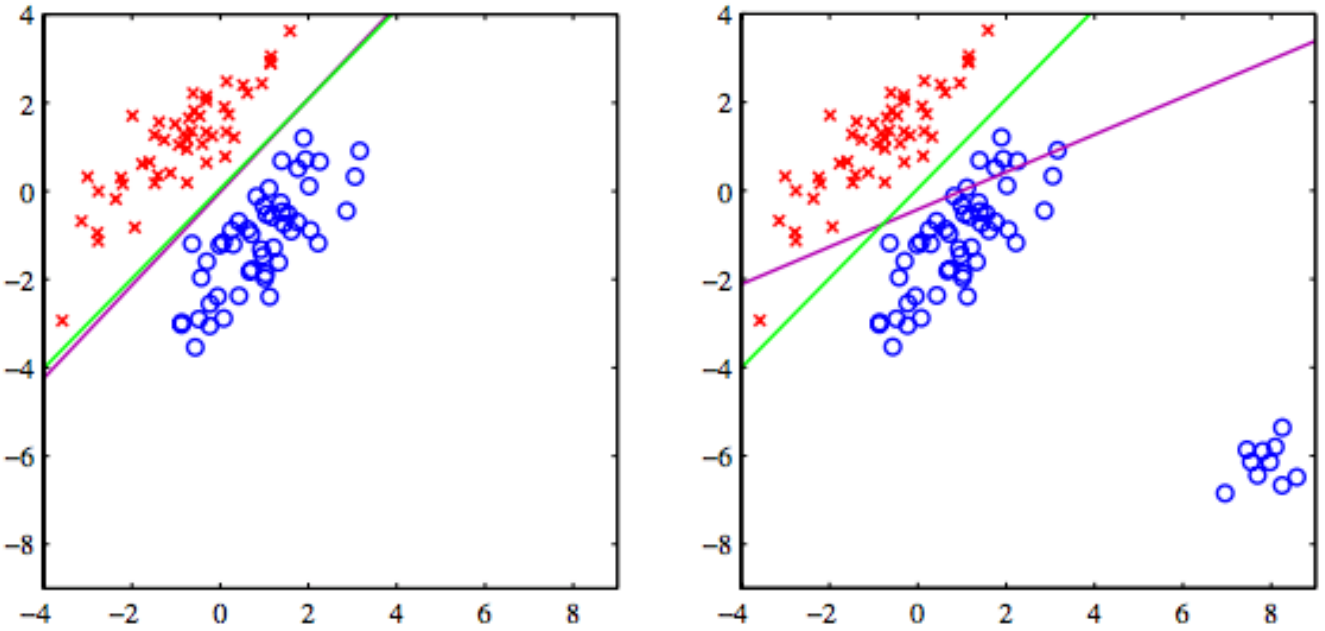
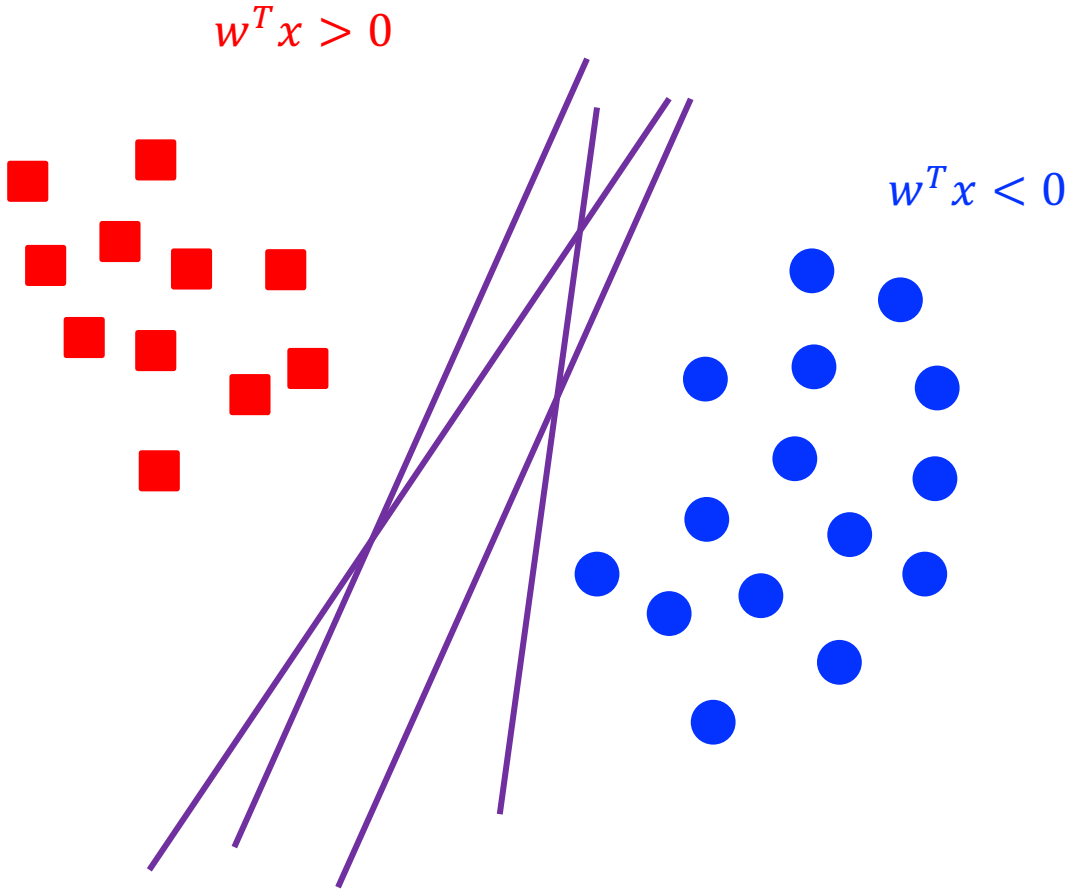


Figure 4.4 The left plot shows data from two classes, denoted by red crosses and blue circles, together with the decision boundary found by least squares (magenta curve) and also by the logistic regression model (green curve), which is discussed later in Section 4.3.2. The right-hand plot shows the corresponding results obtained when extra data points are added at the bottom left of the diagram, showing that least squares is highly sensitive to outliers, unlike logistic regression.

Source: Bishop et al, ”Pattern Recognition and Machine Learning”, Springer 2006

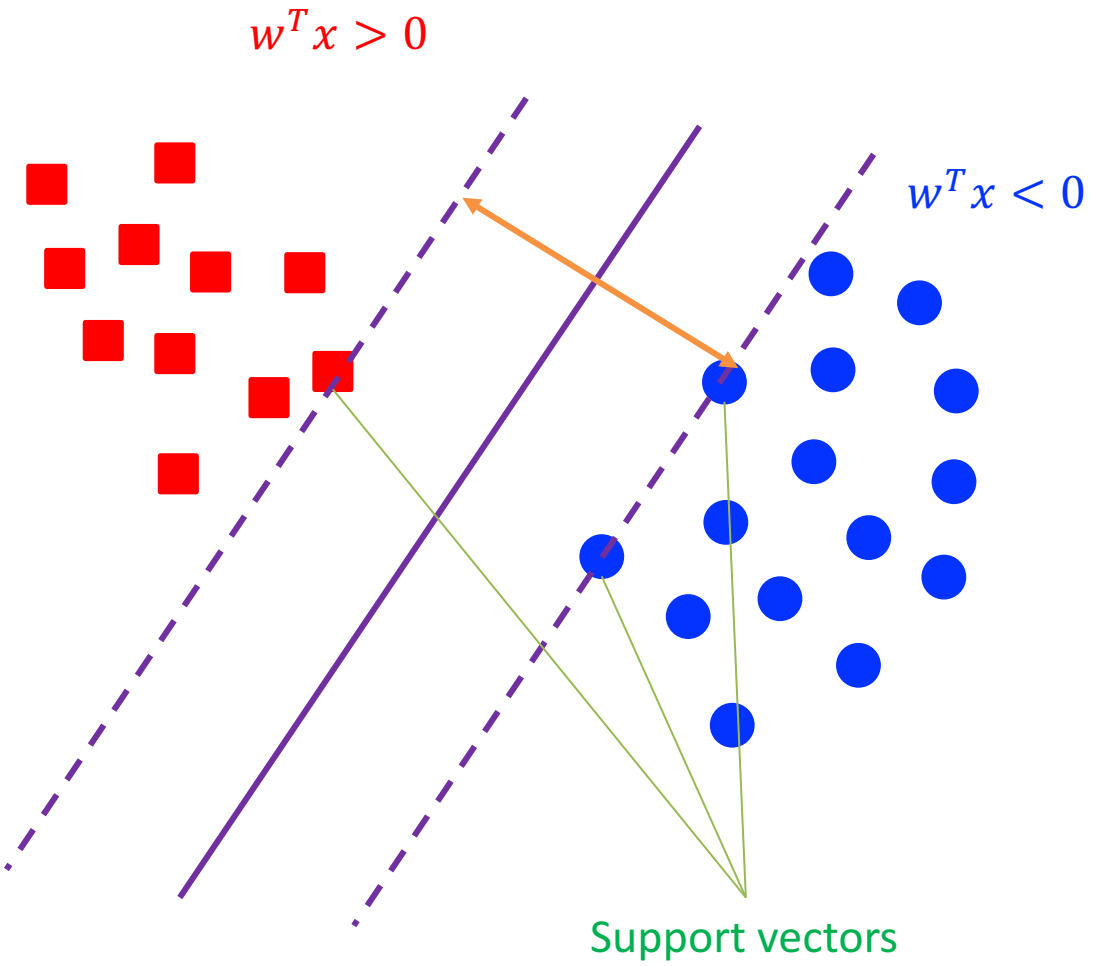
Multiple Linear Separators



Multiple lines separate the training set, which one is the best? The one that minimizes the loss?

One suggestion: The line that is far from the points closest to it from both classes.

Support Vector Machine (SVM)



A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between the two classes.

Outline

- Introduction to supervised learning
- A few supervised learning methods
  - Linear Regression/classification
  - Nearest Neighbor Methods
  - Decision Tree

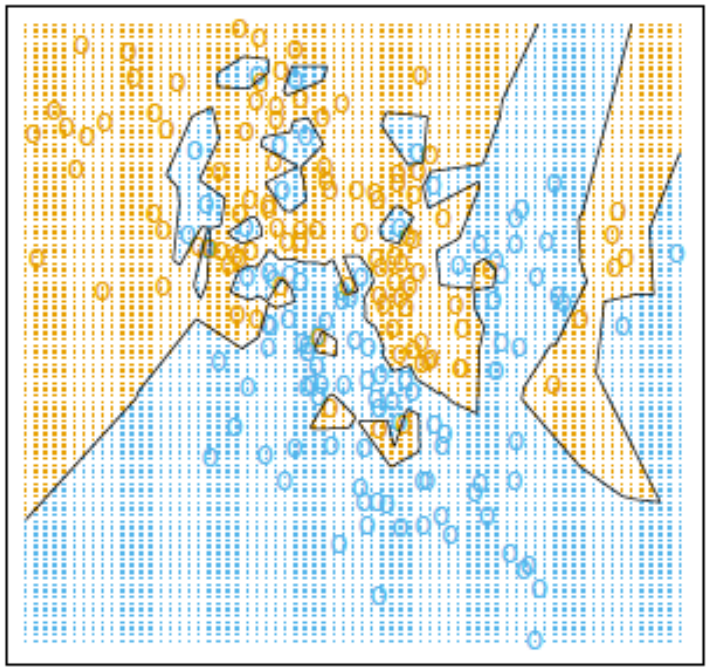
- Potential problems

K-Nearest Neighbor Methods (kNN)

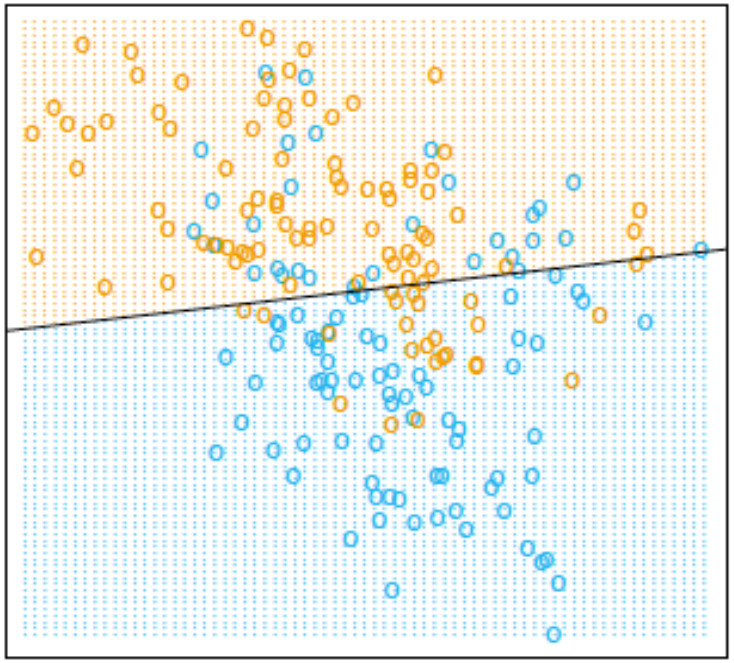
- Learning Algorithm: Store training examples
- Prediction Algorithm:
  - Regression: Take the average value of k nearest neighbors
  - Classification: Assign to the most frequent class of k nearest neighbors.
- Easy to train, but high-computation cost at prediction.

Example

Nearest neighbor, k=1



Linear classification



Linear Regression vs kNN*

	Advantages	Disadvantages
Linear regression	Easy to fit, easy to interpret	Strong assumptions on linear relationship
kNN	Flexible, no assumption on the explicit form of y=f(x)	Hard to understand and interpret

*Demo in the tutorial to compare the two methods.

Outline

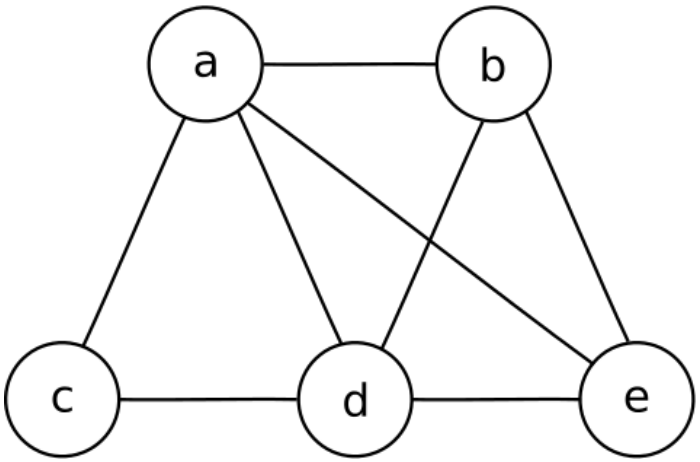
- Introduction to supervised learning
- A few supervised learning methods
  - Linear Regression/classification
  - Nearest Neighbor Methods
  - Decision Tree
- Potential problems

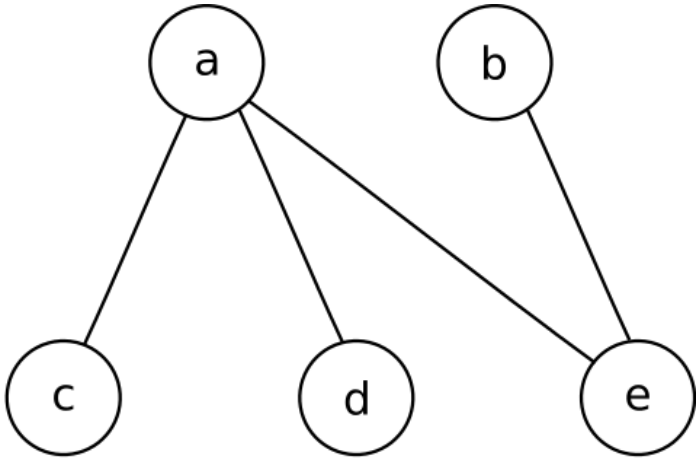
A Short Note on Graph Theory

- Graph is a set G=(V,E) of (V)ertices and (E)=edges.
- Each edge connect two vertices
- A few definitions:
  - Path: sequence of edges which connect a sequence of vertices. These vertices are distinct with each other
  - Cycle: a subset of edges on the graph that forms a path, such that the first node of the path corresponds to the last node.

”Tree” in Graph Theory

- Tree: Connected graph without cycles
  - Any two vertices are connected by only 1 path.
  - A leaf of an unrooted tree is a node of vertex degree 1
- Which one is a tree? Which are its leaves?





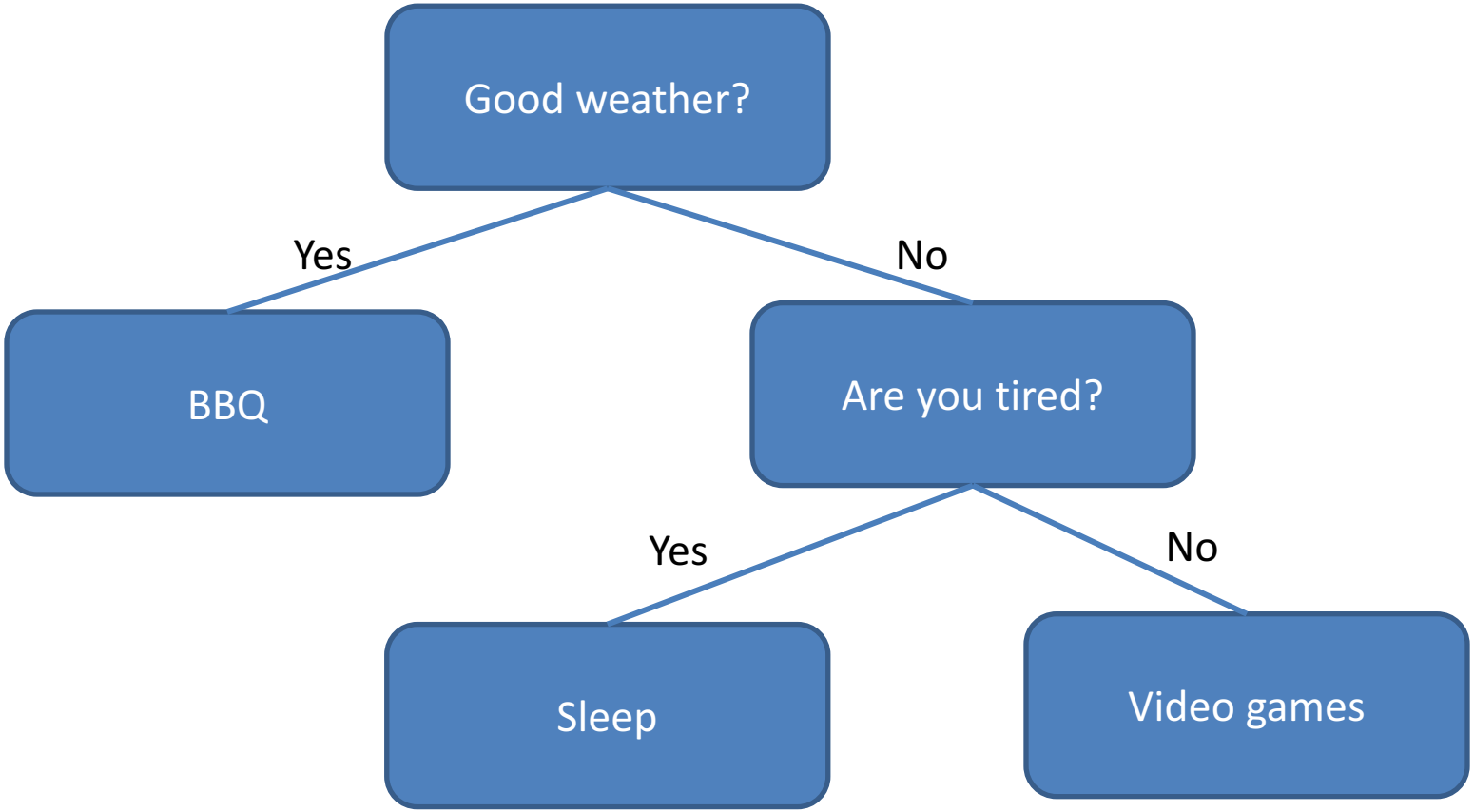
Decision Tree

- Represents a data on a tree such that:
  - Each internal node: test one attribute Xi
  - Each branch from a node: selects one value for Xi
    - If the value is continuous, select value against a threshold
    - E.g., Over 90% score: A range; Below 90%: B or below
  - Each leaf node: predict Y

Example

- Imagine your holiday plan:
  - If good weather I will go to BBQ
  - If bad weather, depends on whether I feel tired. If not I play video games, otherwise I sleep.
- First decide “is the weather good?” If bad weather, decide if I’m tired.
  - Only one attribute at a time.

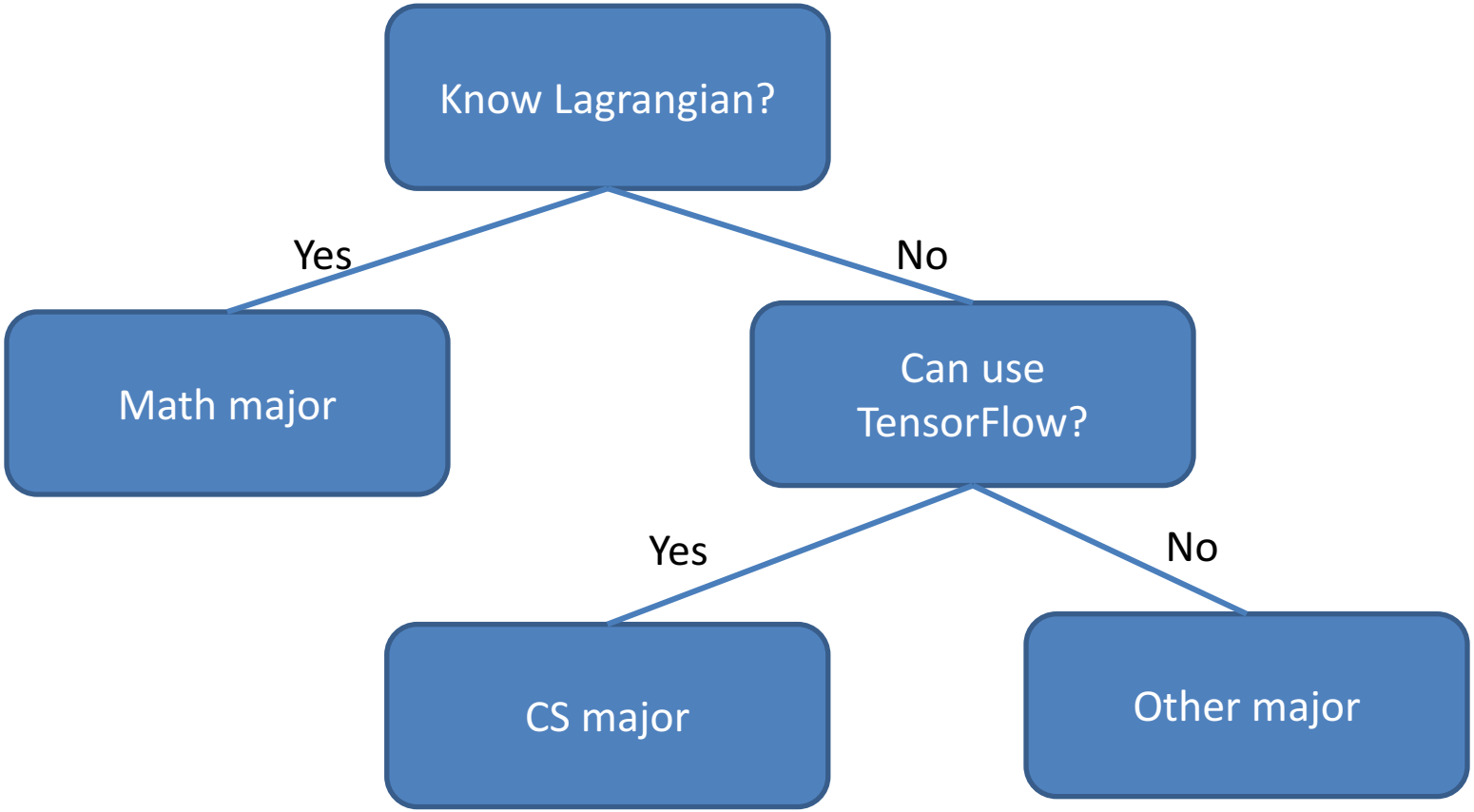
Example


Good weather?
Yes
No
BBQ
Are you tired?
Yes
No
Sleep
Video games

Example (2)

- Consider students in MSBD 5001 they are from math, CS, or other majors.
- Can you predict students' major based on following Yes/no survey questions?
  - Do you know how to do deep learning with TensorFlow?
  - Do you know what Lagrangian Multiplier is?
  - …
- Can you draw a decision tree on the board?

Example (2)


Know Lagrangian?
Yes
No
Math major
Can use TensorFlow?
Yes
No
CS major
Other major

Building a Decision Tree

- Why do we put Lagrangian on the top instead of TensorFlow?
  - Or why do you put an attribute on the top?
- Intuition: Attributes on “top” of the tree should be more “useful”, i.e., provide us most information gain.
  - The more information gain an attribute provides, the more important it is.

Information Gain: Entropy

- Entropy on a single attribute (pi: probability of ith category):

$$E(T) = -\sum_{i=1}^{n} P(T=i)\log_2 P(T=i)$$

- Entropy after adding new attribute:

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

- Information gain is E(T)-E(T,X). Pick the attribute with largest information gain

Example

- 30 students: 14 math major, 16 CS major
- After question Lagrangian:
  - Know Lagrangian: 13 math major, 4 CS major
  - Don’t know: 1 math major, 12 CS major

- What is the information gain?

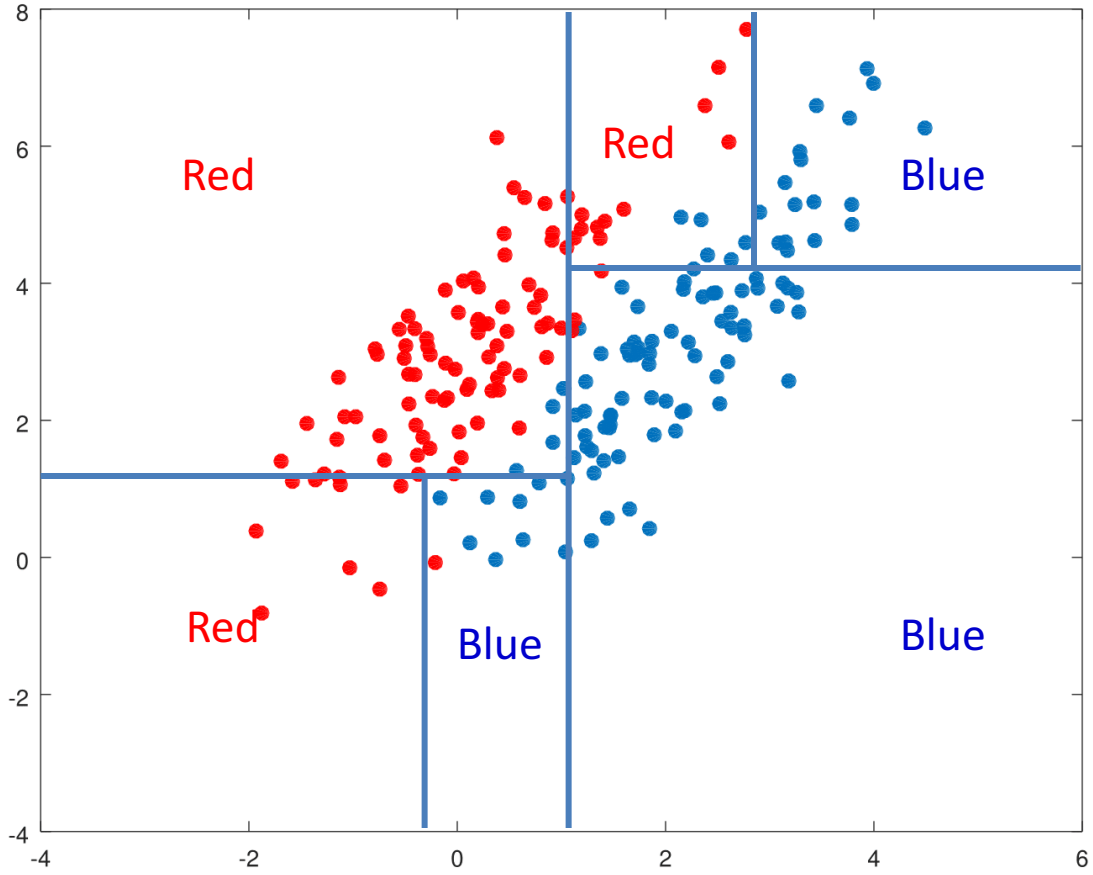
Example

- Entropy before Lagrangian E(X) = -(14/30 $\log_2$ 14/30) – (16/30 $\log_2$ 16/30)=0.996
- After Lagrangian:
  - Know: E(X,L1) = -(13/17 $\log_2$ 13/17) – (4/17 $\log_2$ 4/17) = 0.787
  - Don't know: E(X,L2) = -(1/13 $\log_2$ 1/13) – (12/13 $\log_2$ 12/13) = 0.391
- Take weighted average:
  17/30*0.787+13/30*0.391=0.615
- Information gain: 0.996-0.615=0.381

Basic Algorithm for Building Decision Tree

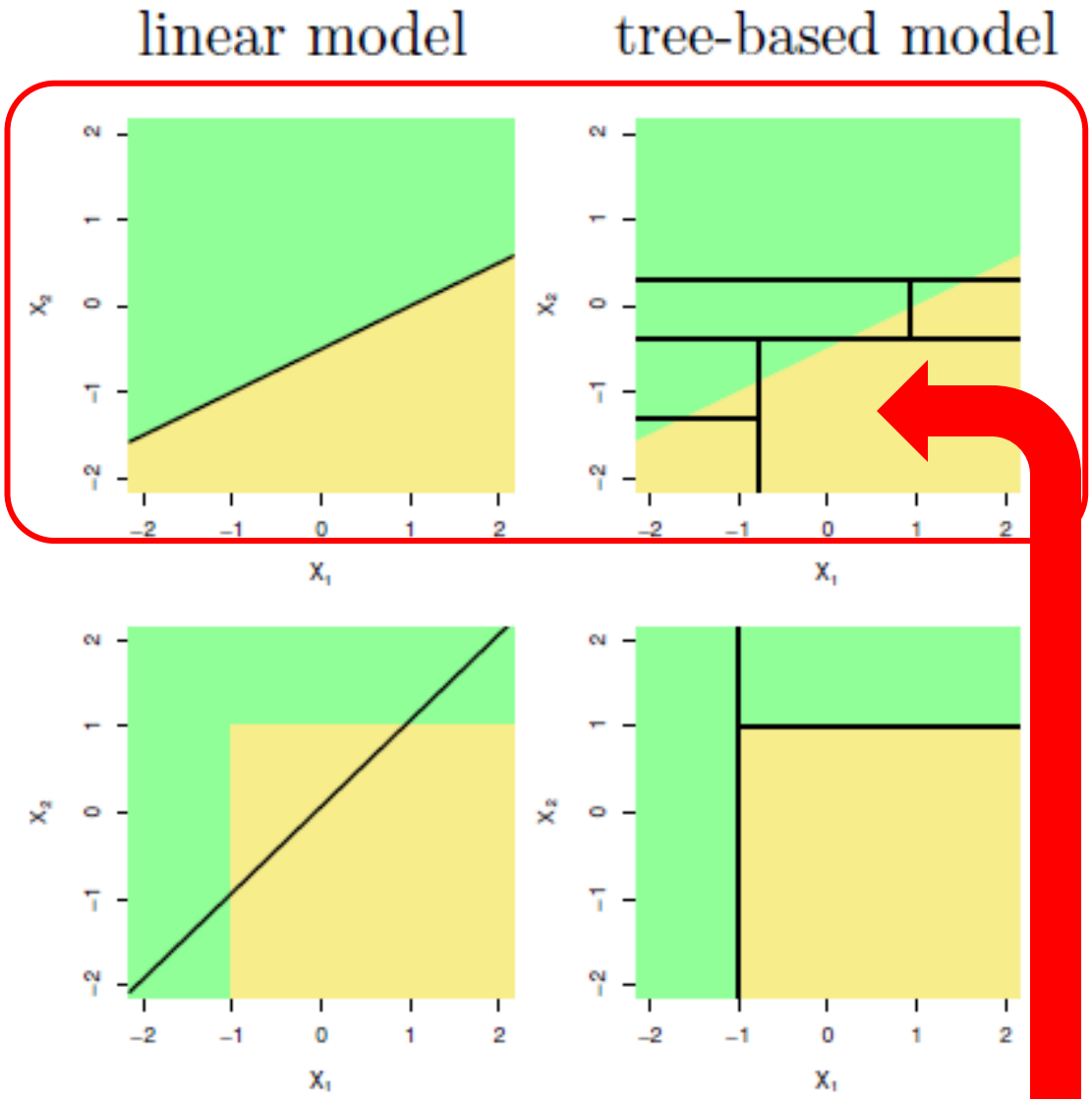
- 1. Select an attribute A with most information gain as root.
- 2. For each value of A, create a subtree
- 3. Repeat 1,2 on each subtree.

Visualizing Decision Trees



Every time: find one feature with highest information gain, split the feature into two

Linear Models vs. Decision Trees



Understand your data → select right model!

Wrap-up: Machine Learning in 3 Steps

- 1. Collect data, extract features
- 2. Determine a model
  - Selecting good model requires good understanding of data!
- 3. Train the model with the data, i.e., “learning”

Topics Next Week

- Our methods are based on the assumption that training and testing data are drawn i.i.d. from the same distribution
  - Often hard to achieve.

- Model may perform well on training data, but poor on testing data
- We will see in next lecture!