Supervised Learning Framework and Decision Tree Representation

Fall 2022

Instructor: Shandian Zhe
Outline

1. The general settings of supervised learning

2. Warm-up: Decision tree representation
What is supervised learning?
Instances and Labels

Running example: Automatically tag news articles
Instances and Labels

Running example: Automatically tag news articles

An instance of a news article that needs to be classified
Instances and Labels

Running example: Automatically tag news articles

An instance of a news article that needs to be classified

Sports

A label
Instances and Labels

Running example: Automatically tag news articles

Instance Space: All possible news articles

Label Space: All possible labels

Mapped by the classifier to:
- Sports
- Business
- Politics
- Entertainment
Instances and Labels

$X$: Instance Space

The set of examples that need to be classified

Eg: The set of all possible names, documents, sentences, images, emails, etc.
Instances and Labels

$X$: Instance Space

The set of examples that need to be classified

Eg: The set of all possible names, documents, sentences, images, emails, etc.

$Y$: Label Space

The set of all possible labels

Eg: \{\textit{Spam}, \textit{Not-Spam}\}, \{+,-\}, etc.
Instances and Labels

$X$: Instance Space

The set of examples that need to be classified

Eg: The set of all possible names, documents, sentences, images, emails, etc

$Y$: Label Space

The set of all possible labels

Eg: \{Spam, Not-Spam\}, \{+, -\}, etc.

Target function $y = f(x)$
Instances and Labels

$X$: Instance Space
The set of examples that need to be classified

$Y$: Label Space
The set of all possible labels

$\text{Target function } y = f(x)$

The goal of learning: Find this target function

Eg: The set of all possible names, documents, sentences, images, emails, etc

Eg: {Spam, Not-Spam}, {+, -}, etc.

Learning is search over functions
Supervised learning

$X$: Instance Space

The set of examples

$Y$: Label Space

The set of all possible labels

Target function $y = f(x)$

Learning algorithm only sees examples of the function $f$ in action
Supervised learning: Training

\[ X: \text{Instance Space} \]
\[ \text{The set of examples} \]

\[ Y: \text{Label Space} \]
\[ \text{The set of all possible labels} \]

Target function\[ y = f(x) \]

Learning algorithm only sees examples of the function \( f \) in action

Labeled training data

\[ (x_1, f(x_1)) \]
\[ (x_2, f(x_2)) \]
\[ (x_3, f(x_3)) \]
\[ \vdots \]
\[ (x_N, f(x_N)) \]
Supervised learning: Training

\[ X: \text{Instance Space} \]
The set of examples

\[ Y: \text{Label Space} \]
The set of all possible labels

Target function
\[ y = f(x) \]

Learning algorithm only sees examples of the function \( f \) in action

Labeled training data

\[(x_1, f(x_1))\]
\[(x_2, f(x_2))\]
\[(x_3, f(x_3))\]
\[\vdots\]
\[(x_N, f(x_N))\]

Learning algorithm
Supervised learning: Training

$X$: Instance Space
The set of examples

$Y$: Label Space
The set of all possible labels

Target function $y = f(x)$

Learning algorithm only sees examples of the function $f$ in action

Labeled training data

Learning algorithm

A learned function $g: X \to Y$
Supervised learning: Training

$X$: Instance Space
The set of examples

$Y$: Label Space
The set of all possible labels

Target function $y = f(x)$

Learning algorithm only sees examples of the function $f$ in action

Labeled training data

Learning algorithm

A learned function $g: X \rightarrow Y$

Can you think of other protocols?
Unsupervised learning

\( X: \text{Instance Space} \)

The set of examples

\( Y: \text{Label Space} \)

The set of all possible labels

Target function

\[ y = f(x) \]

Learning algorithm only sees examples of the function \( f \) in action

Learning algorithm

A learned function \( g: X \rightarrow Y \)

Un-Labeled training data
Semi-Supervised learning

$X$: Instance Space

The set of examples

$Y$: Label Space

The set of all possible labels

Target function $y = f(x)$

Learning algorithm only sees examples of the function $f$ in action

Learning algorithm

A learned function $g: X \rightarrow Y$

Partially Labeled training data
Supervised learning: Training

- $X$: Instance Space
  - The set of examples

- $Y$: Label Space
  - The set of all possible labels

Target function: $y = f(x)$

Learning algorithm only sees examples of the function $f$ in action

Labeled training data:

- $(x_1, f(x_1))$
- $(x_2, f(x_2))$
- $(x_3, f(x_3))$
- $\ldots$
- $(x_N, f(x_N))$

Learning algorithm

A learned function $g$: $X \rightarrow Y$
Supervised learning: Evaluation

\[ X: \text{Instance Space} \]

The set of examples

\[ Y: \text{Label Space} \]

The set of all possible labels

Target function

\[ y = f(x) \]

Learned function

\[ y = g(x) \]
Supervised learning: Evaluation

- **$X$: Instance Space**
  - The set of examples

- **$Y$: Label Space**
  - The set of all possible labels

Target function: $y = f(x)$

Learned function: $y = g(x)$

Draw test example $x \in X$

Are they different?

$f(x)$

$g(x)$
Supervised learning: Evaluation

\[ X: \text{Instance Space} \]
- The set of examples

\[ Y: \text{Label Space} \]
- The set of all possible labels

Target function \( y = f(x) \)

Learned function \( y = g(x) \)

Draw test example \( x \in X \)

\[ f(x) \quad \rightarrow \quad g(x) \]

Are they different?

Apply the model to many test examples and compare to the target’s prediction
Supervised learning: Evaluation

**Instance Space**  
The set of examples

**Label Space**  
The set of all possible labels

**Target function**  
\( y = f(x) \)

**Learned function**  
\( y = g(x) \)

Draw test example \( x \in X \)

\( f(x) \)  
\( g(x) \)

Are they different?

Apply the model to many test examples and compare to the target’s prediction

*Can you use test examples during training?*
Supervised learning: General setting

• Given: Training examples of the form \langle x, f(x) \rangle
  – The function \( f \) is an unknown function

• Typically the input \( x \) is represented in a **feature space**
  – Example: \( x \in \{0,1\}^n \) or \( x \in \mathbb{R}^n \)
  – A deterministic mapping from instances in your problem (e.g., news articles) to feature vectors

• Goal: Find a good approximation for \( f \)

• The label determines the kind of problem we have
  – **Binary classification**: \( f(x) \in \{-1,1\} \)
  – **Multiclass classification**: \( f(x) \in \{1, 2, 3, \cdots, K\} \)
  – **Regression**: \( f(x) \in \mathbb{R} \)

Questions?
On using supervised learning

We need to decide:

1. **What is our instance space?**
   What are the inputs to the problem? What are the features? How to extract features?

2. **What is our label space?**
   What is the prediction task?

3. **What is our hypothesis space?**
   What functions should the learning algorithm search over?

4. **What is our learning algorithm?**
   How do we learn from the labeled data?

5. **What is our loss function or evaluation metric?**
   What is success?
Taxonomy

- Supervised Learning
  - Decision trees
  - Boosting and ensemble learning
  - Linear models
    - Linear regression
    - Logistic regression
    - Naïve Bayes
    - Perceptron
    - Support vector machines
    - ...
  - Nonlinear models
    - Nearest neighbor
    - Polynomial regression
    - Kernel methods: kernel SVM, kernel perceptron, kernel regression ...
    - Artificial neural networks: CNN, MLP, auto-encoder, ...
    - ...
- Unsupervised Learning
  - Clustering: kmeans, k-medoid, GMM, ...
  - Dimension reduction: PCA, SVD, Embedding learning, Isomap, ...
  - ...
- Semi-Supervised Learning
  - Mostly combine loss functions from supervised and unsupervised learning
Taxonomy

• Supervised Learning
  – **Decision trees**
  – Boosting and ensemble learning
  – Linear models
    • Linear regression
    • Logistic regression
    • Naïve Bayes
    • Perceptron
    • Support vector machines
    • ...
  – Nonlinear models
    • Nearest neighbor
    • Polynomial regression
    • Kernel methods: kernel SVM, kernel perceptron, kernel regression ....
    • Artificial neural networks: CNN, MLP, auto-encoder, ....
    • ....

• Unsupervised Learning
  – Clustering: kmeans, k-medoid, GMM, ...
  – Dimension reduction: PCA, SVD, Embedding learning, Isomap, ....
  – ...

• Semi-Supervised Learning
  – Mostly combine loss functions from supervised and unsupervised learning
We Will Start with Decision Trees

1. Representation: What are decision trees?

2. Algorithm: Learning/create decision trees
   – The ID3 algorithm: A greedy heuristic

3. Variants/Extensions: post pruning, other node splitting principles, ...
Why Decision Trees?

1. Simple and intuitive: a set of rules

2. Interpretability: again, rules!

3. Can be promoted to be very strong: boosting, random forest, gradient boosting trees, ...

Decision trees are very widely used!
Warm-up

1. **Representation**: What are decision trees?

2. **Algorithm**: Learning decision trees
   - The ID3 algorithm: A greedy heuristic

3. Some extensions
What are decision trees?

• A **hierarchical structure** that represents data with a divide-and-conquer strategy

• **Each node** represents a data subset, and meanwhile is associated with a test feature

• From the node, one **branch** is created for every value that the feature can take (hence further partition the data accordingly)

• **Leaves** of the tree specify the class labels (the point to stop partition)
Decision Tree Learning

• **General idea**: Given a collection of labeled examples, construct a decision tree that represents it
Let’s build a decision tree for classifying shapes
Let’s build a decision tree for classifying shapes

Before building a decision tree:

What is the label for a red triangle? And why?
Let’s build a decision tree for classifying shapes

What are some attributes of the examples?
Let’s build a decision tree for classifying shapes

What are some attributes of the examples?

Color, Shape
Let's build a decision tree for classifying shapes

What are some attributes of the examples?

Color, Shape

Color?
Let's build a decision tree for classifying shapes

What are some attributes of the examples?

Color, Shape

Color?

Blue

Red

Green

Label=C

Label=B

Label=A
Let’s build a decision tree for classifying shapes

What are some attributes of the examples?

Color, Shape

Color?

Blue

Red

Green

B

Label=A

Label=B

Label=C
Let’s build a decision tree for classifying shapes.

What are some attributes of the examples?

Color, Shape

- Color: Blue, Red, Green
- Shape: Square, Triangle, Circle

Label=A
Label=B
Label=C
Let’s build a decision tree for classifying shapes

What are some attributes of the examples?

Color, Shape
Let’s build a decision tree for classifying shapes

What are some attributes of the examples?

- Color
- Shape

1. How to use a decision tree for *prediction*?
   - What is the label for a red triangle?
     - Just follow a path from the root to a leaf
   - What about a green triangle?
Essence of Decision trees

What decision functions do decision trees represent?

Every path from the root to a leaf is a rule

The full tree is equivalent to the conjunction of all the rules
Essence of Decision trees

What decision functions do decision trees represent?

Every path from the root to a leaf is a rule

The full tree is equivalent to the conjunction of all the rules

\[(\text{Color} = \text{blue} \ \text{AND} \ \text{Shape} = \text{triangle} \ \text{Label} = \text{B}) \ \text{AND} \ \]
\[(\text{Color} = \text{blue} \ \text{AND} \ \text{Shape} = \text{square} \ \text{Label} = \text{A}) \ \text{AND} \ \]
\[(\text{Color} = \text{blue} \ \text{AND} \ \text{Shape} = \text{circle} \ \text{Label} = \text{C}) \ \text{AND} \ldots \]
Decision Trees

• Outputs are discrete categories

• But real valued outputs are also possible (regression trees)
  How to?

• Methods for handling noisy data (noise in the label or in the features) and for handling missing attributes
  – Pruning trees helps with noise
  – More on this later...
Decision Trees

- Outputs are discrete categories

- But real valued outputs are also possible (regression trees)
  How to? Take average!

- Methods for handling noisy data (noise in the label or in the features) and for handling missing attributes
  - Pruning trees helps with noise
  - More on this later...
Numeric attributes and decision boundaries

• We have seen instances represented as attribute-value pairs (color=blue, second letter=e, etc.)
  – Values have been categorical

• How do we deal with numeric feature values? (eg length = ?)
Numeric attributes and decision boundaries

• We have seen instances represented as attribute-value pairs (color=blue, second letter=e, etc.)
  – Values have been categorical

• How do we deal with **numeric feature values**? (eg length = ?)
  – Discretize them or use thresholds on the numeric values
Numeric attributes and decision boundaries

• We have seen instances represented as attribute-value pairs (color=blue, second letter=e, etc.)
  – Values have been categorical

• How do we deal with numeric feature values? (eg length = ?)
  – Discretize them or use thresholds on the numeric values
  – This example divides the feature space into axis parallel rectangles
Numeric attributes and decision boundaries

• We have seen instances represented as attribute-value pairs (color=blue, second letter=e, etc.)
  – Values have been categorical

• How do we deal with **numeric feature values**? (eg length = ?)
  – Discretize them or use thresholds on the numeric values
  – This example divides the feature space into axis parallel rectangles
Numeric attributes and decision boundaries

- We have seen instances represented as attribute-value pairs (color=blue, second letter=e, etc.)
  - Values have been categorical
- How do we deal with numeric feature values? (eg length = ?)
  - Discretize them or use thresholds on the numeric values
  - This example divides the feature space into axis parallel rectangles

```
1 3  X
Y
7  +  +  +  +
5  +  +  -  -
  -  +  -  -
  1  3  X
```

```
X<3
  no  yes
  Y>7
    no  yes
    X<1
      no  yes
      Y<5
        no  yes
          +  +
```
Numeric attributes and decision boundaries

- We have seen instances represented as attribute-value pairs (color=blue, second letter=e, etc.)
  - Values have been categorical

- How do we deal with numeric feature values? (e.g., length = ?)
  - Discretize them or use thresholds on the numeric values
  - This example divides the feature space into axis parallel rectangles

*Decision boundaries* can be non-linear
Summary: Decision trees

- Compact data representation
- A natural representation (think 20 questions)
- **Predicting** with a decision tree is easy, how?

- Clearly, given a dataset, there are many decision trees that can represent it. Why?
Exercise

Write down the decision tree for the shapes data if the root node was *Shape* instead of *Color*.

Will the two trees make the same predictions for unseen shapes/color combinations?