Lecture 12: Decision Trees

Iain Styles

22 November 2018

Learning Outcomes

By the end of this lecture you should be able to:

- Understand the concept of a decision tree
- Appreciate that decision trees can be constructed in different ways
- Understand and be able to apply the concept of information entropy to construct a decision tree
- Appreciate some of the limitations of decision trees.

Introduction

- Decision trees mimic the way in which humans make decisions.
- We do not (consciously) map every problem into a vector notation
- In a decision tree, we learn an explicit set of binary decisions on the features.
- Followed in sequence these form a classification rule.

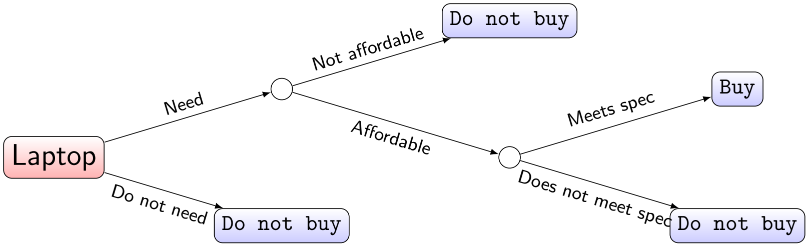
Introduction

- Decision trees mimic the way in which humans make decisions.
- We do not (consciously) map every problem into a vector notation
- In a decision tree, we learn an explicit set of binary decisions on the features.
- Followed in sequence these form a classification rule.
- A (trivial) example: buying a new laptop.
  - Do I need a new laptop?
  - Can I afford it?
  - Does it meet my specication?

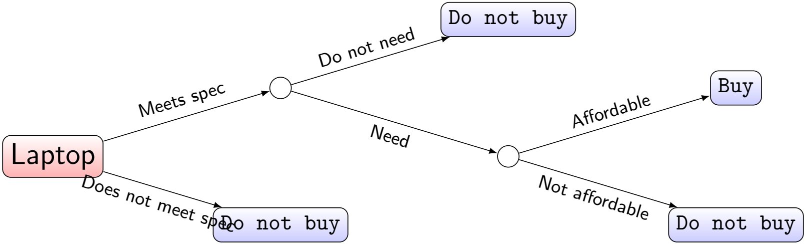
Introduction

- Decision trees mimic the way in which humans make decisions.
- We do not (consciously) map every problem into a vector notation
- In a decision tree, we learn an explicit set of binary decisions on the features.
- Followed in sequence these form a classification rule.
- A (trivial) example: buying a new laptop.
  - Do I need a new laptop?
  - Can I afford it?
  - Does it meet my specication?
- In a decision tree we apply each of these questions in turn to arrive at a final decision.

A Simple Decision Tree



Why not a different tree?



Learning a Tree

- How can we construct this algorithmically?

Learning a Tree

- How can we construct this algorithmically?
  1. Start with a dataset $\mathcal{D} = \{\mathbf{x}_i, t_i\}_{i=1}^{N}$ with binary target variables $t_i = \{-1, 1\}$. Each data point $\mathbf{x}_i$ is a vector of P features.

Learning a Tree

- How can we construct this algorithmically?
  1. Start with a dataset $\mathcal{D} = \{\mathbf{x}_i, t_i\}_{i=1}^{N}$ with binary target variables $t_i = \{-1, 1\}$. Each data point $\mathbf{x}_i$ is a vector of P features.
  2. Determine which feature is best able to split the dataset according to its target values and determine the value on which to split.

Learning a Tree

- How can we construct this algorithmically?
  1. Start with a dataset $\mathcal{D} = \{\mathbf{x}_i, t_i\}_{i=1}^{N}$ with binary target variables $t_i = \{-1, 1\}$. Each data point $\mathbf{x}_i$ is a vector of P features.
  2. Determine which feature is best able to split the dataset according to its target values and determine the value on which to split.
  3. Split the dataset into two according to the decision learned in the previous step.

Learning a Tree

- How can we construct this algorithmically?
  1. Start with a dataset $\mathcal{D} = \{\mathbf{x}_i, t_i\}_{i=1}^{N}$ with binary target variables $t_i = \{-1, 1\}$. Each data point $\mathbf{x}_i$ is a vector of P features.
  2. Determine which feature is best able to split the dataset according to its target values and determine the value on which to split.
  3. Split the dataset into two according to the decision learned in the previous step.
  4. Recurse on the two partitioned subsets.

Learning a Tree

- How can we construct this algorithmically?
  1. Start with a dataset $\mathcal{D} = \{\mathbf{x}_i, t_i\}_{i=1}^N$ with binary target variables $t_i = \{-1, 1\}$. Each data point $\mathbf{x}_i$ is a vector of P features.
  2. Determine which feature is best able to split the dataset according to its target values and determine the value on which to split.
  3. Split the dataset into two according to the decision learned in the previous step.
  4. Recurse on the two partitioned subsets.
  5. Stop recursing if a subset contains samples from only one of the target classes

Learning a Tree

- How can we construct this algorithmically?
  1. Start with a dataset $\mathcal{D} = \{\mathbf{x}_i, t_i\}_{i=1}^{N}$ with binary target variables $t_i = \{-1, 1\}$. Each data point $\mathbf{x}_i$ is a vector of P features.
  2. Determine which feature is best able to split the dataset according to its target values and determine the value on which to split.
  3. Split the dataset into two according to the decision learned in the previous step.
  4. Recurse on the two partitioned subsets.
  5. Stop recursing if a subset contains samples from only one of the target classes
- How to determine the feature order?

When and how to split

- Goal: split data such that the target can be predicted

When and how to split

- Goal: split data such that the target can be predicted
- Must maximise intra-group homogeneity (target values grouped together)

When and how to split

- Goal: split data such that the target can be predicted
- Must maximise intra-group homogeneity (target values grouped together)
- Choose feature order accordingly – how?

When and how to split

- Goal: split data such that the target can be predicted
- Must maximise intra-group homogeneity (target values grouped together)
- Choose feature order accordingly – how?
- We quantify the *information gained* by splitting the data

When and how to split

- Goal: split data such that the target can be predicted
- Must maximise intra-group homogeneity (target values grouped together)
- Choose feature order accordingly – how?
- We quantify the *information gained* by splitting the data
- Key quantity: Entropy

Entropy

- Concept is from thermodynamics
- For a system with a given set of macroscopic properties. . .

Entropy

- Concept is from thermodynamics
- For a system with a given set of macroscopic properties...
- ...how many microscopic states are there with those properties?

Entropy

- Concept is from thermodynamics
- For a system with a given set of macroscopic properties...
- ...how many microscopic states are there with those properties?

$$S = k \ln w \tag{1}$$

- Example: gas in a box

Entropy

- Concept is from thermodynamics
- For a system with a given set of macroscopic properties...
- ...how many microscopic states are there with those properties?

$$S = k \ln w \tag{1}$$

- Example: gas in a box
- Organised states possible, but highly unlikely

Entropy

- Concept is from thermodynamics
- For a system with a given set of macroscopic properties...
- ...how many microscopic states are there with those properties?

$$S = k \ln w \tag{1}$$

- Example: gas in a box
- Organised states possible, but highly unlikely
- Information Entropy has a similar interpretation

Information Entropy

- A measure of how much information an event contains

Information Entropy

- A measure of how much information an event contains
- The core idea of that frequent events are uninformative

Information Entropy

- A measure of how much information an event contains
- The core idea of that frequent events are uninformative
- Infrequent events give new information

Information Entropy

- A measure of how much information an event contains
- The core idea of that frequent events are uninformative
- Infrequent events give new information

$$S = -\sum_i p(i) \ln p(i) \tag{2}$$

Information Entropy

- A measure of how much information an event contains
- The core idea of that frequent events are uninformative
- Infrequent events give new information

$$S = -\sum_i p(i) \ln p(i) \tag{2}$$

- Homogeneous sequences have low entropy
- Random Sequences have high entropy
- We use this to select the feature that gives the biggest gain in information

Information Gain

- Given $S = -\sum_i p(i) \ln p(i)$ we calculate

$$G(P,C) = S(P) - S(C) \tag{3}$$

$$- \sum_{i \in P} p(i) \ln p(i) - \sum_{c \in C} p(c) \sum_{i \in c} -p(i|c) \ln p(i|c) \tag{4}$$

- Let's do an example...

Selecting features by maximising IG

- Outcomes for buying a laptop. . .

Selecting features by maximising IG

- Outcomes for buying a laptop. . .

N	Need	Afford	Spec	Buy
1	T	F	T	F
2	F	T	F	F
3	T	F	T	T
4	T	F	T	T
5	F	T	F	F
6	T	T	T	T
7	F	F	F	F
8	T	T	T	T
9	F	T	T	T
10	T	F	F	F

- What variable should we split on first?

Selecting features by maximising IG

- Parent entropy
- Buy: 4T, 6F

$$S(P) = -\sum_i p(i) \ln p(i)$$
$$= -0.4 \ln 0.4 - 0.6 \ln 0.6 = 0.673$$

N	Need	Afford	Spec	Buy
1	T	F	T	F
2	F	T	F	F
3	T	F	T	T
4	T	F	T	T
5	F	T	F	F
6	T	T	T	T
7	F	F	F	F
8	T	T	T	T
9	F	T	T	T
10	T	F	F	F

Selecting features by maximising IG

- “Need”

Selecting features by maximising IG

- “Need”
- 6T, 4F

Selecting features by maximising IG

- "Need"
- 6T, 4F
- 6T$\mapsto$ 4T, 2F

Selecting features by maximising IG

- "Need"
- 6T, 4F
- 6T$\mapsto$ 4T, 2F
- 4F$\mapsto$ 0T, 4F

Selecting features by maximising IG

- "Need"
- 6T, 4F
- 6T$\mapsto$ 4T, 2F
- 4F$\mapsto$ 0T, 4F

$$
\begin{aligned}
S(C) &= \sum_{c \in C} p(c) \sum_{i \in c} -p(i|c) \ln p(i|c) \\
&= \left[p(\mathrm{Need}) \times \sum_{i \in \mathrm{Need}} -p_i \ln p_i \right] + \left[p(\neg\mathrm{Need}) \times \sum_{i \in \neg\mathrm{Need}} -p_i \ln \right. \\
&= 0.6 \times \left(-\frac{4}{6} \ln \frac{4}{6} - \frac{2}{6} \ln \frac{2}{6} \right) + 0.4 \times (-1 \ln 1 - 0 \ln 0) \\
&= 0.382
\end{aligned}
$$

Selecting features by maximising IG

- “Afford”

Selecting features by maximising IG

- “Afford”
- 5T, 5F

Selecting features by maximising IG

- “Afford”
- 5T, 5F
- 5T$\mapsto$ 2T, 3F

Selecting features by maximising IG

- "Afford"
- 5T, 5F
- 5T$\mapsto$ 2T, 3F
- 5F$\mapsto$ 2T, 3F

Selecting features by maximising IG

- "Afford"
- 5T, 5F
- 5T$\mapsto$ 2T, 3F
- 5F$\mapsto$ 2T, 3F

$$S(C) = 0.5 \times \left(-\frac{2}{5}\ln\frac{2}{5} - \frac{3}{5}\ln\frac{3}{5}\right) + 0.5 \times \left(-\frac{2}{5}\ln\frac{2}{5} - \frac{3}{5}\ln\frac{3}{5}\right) \tag{5}$$

$$= 0.5 \times 0.673 + 0.5 \times 0.673 = 0.673 \tag{6}$$

$x \cdot \ln x$

Selecting features by maximising IG

- “Spec”

Selecting features by maximising IG

- "Spec"
- 6T, 4F

Selecting features by maximising IG

- “Spec”
- 6T, 4F
- 6T$\mapsto$ 5T, 1F

Selecting features by maximising IG

- "Spec"
- 6T, 4F
- 6T$\mapsto$ 5T, 1F
- 4F$\mapsto$ 0T, 4F

Selecting features by maximising IG

- "Spec"
- 6T, 4F
- 6T$\mapsto$ 5T, 1F
- 4F$\mapsto$ 0T, 4F

$$S(C) = 0.6 \times \left(-\frac{5}{6}\ln\frac{5}{6} - \frac{1}{6}\ln\frac{1}{6}\right) + 0.4 \times (-1\ln 1 - 0\ln 0) \tag{7}$$

$$= 0.451 + 0 = 0.270 \tag{8}$$

Selecting features by maximising IG

- So, the information gained during each split is:

Selecting features by maximising IG

- So, the information gained during each split is:
- Need: $0.673 - 0.382 = 0.291$

Selecting features by maximising IG

- So, the information gained during each split is:
- Need: $0.673 - 0.382 = 0.291$
- Afford: $0.673 - 0.673$

Selecting features by maximising IG

- So, the information gained during each split is:
- Need: $0.673 - 0.382 = 0.291$
- Afford: $0.673 - 0.673$
  No information gained (both groups have same outcome distribution as parent)

Selecting features by maximising IG

- So, the information gained during each split is:
- Need: $0.673 - 0.382 = 0.291$
- Afford: $0.673 - 0.673$
  No information gained (both groups have same outcome distribution as parent)
- Spec: $0.673 - 0.270 = 0.403$

Selecting features by maximising IG

- So, the information gained during each split is:
- Need: $0.673 - 0.382 = 0.291$
- Afford: $0.673 - 0.673$
  No information gained (both groups have same outcome distribution as parent)
- Spec: $0.673 - 0.270 = 0.403$
- The best initial predictor of purchasing a new laptop is its specification.

Selecting features by maximising IG

- So, the information gained during each split is:
- Need: $0.673 - 0.382 = 0.291$
- Afford: $0.673 - 0.673$
  No information gained (both groups have same outcome distribution as parent)
- Spec: $0.673 - 0.270 = 0.403$
- The best initial predictor of purchasing a new laptop is its specification.
- Apply these ideas recursively to each partition to build the tree.

Tips and Tricks for Decision Trees

- Decision trees can always fit their training data exactly: low bias
- But leads to unstable model: high variance
- Homogeneous leaf nodes can lead to serious overfitting, especially on small data
- Can be good to limit tree depth: more robust to model noise
- Model is easy to interpret

Summary

- Decision Trees are an intuitive way to build interpretable classifiers
- But they are unstable
- Uncommon to use a single tree
- Much more common to use them as part of an *ensemble*
- Next lecture: how to use ensembles of weak learners to build a strong learner