Decision Trees

Sven Koenig, USC

Russell and Norvig, 3rd Edition, Section 18.3

These slides are new and can contain mistakes and typos.
Please report them to Sven (skoenig@usc.edu).

1

Rule Learning

- So far, we assumed that rules need to be specified by experts.
- Sometimes, this works well and, sometimes, it does not.
- For example, people have trouble specifying how to ride a bicycle without falling even if they are experts at it.
- We now find out how a system can learn rules from examples.
- Thus, we study how to acquire knowledge with machine learning.

2

Inductive Learning for Classification

- Labeled examples

How old are they?	What is their current salary per year?	Do they have a savings account?	Have they ever declared bankruptcy?	…	Would you issue a credit card to them?
52	$150,000	yes	no	…	yes
40	$50,000	no	yes	…	no
20	$60,000	yes	no	…	yes
31	$20,000	yes	no	…	yes

- Unlabeled examples

How old are they?	What is their current salary per year?	Do they have a savings account?	Have they ever declared bankruptcy?	…	Would you issue a credit card to them?
26	$40,000	no	no	…	?

3

Inductive Learning for Classification

- Labeled examples

Feature_1	Feature_2	Class
true	true	true
true	false	false
false	true	false

- Unlabeled examples

Feature_1	Feature_2	Class
false	false	?

4

Inductive Learning for Classification

- Labeled examples

Feature_1	Feature_2	Class
true	true	true
true	false	false
false	true	false

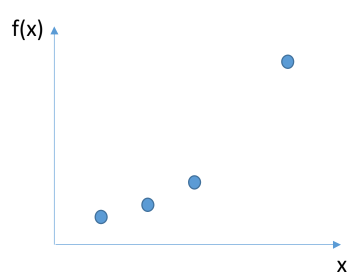
Learn f(Feature_1, Feature_2) = Class from
f(true, true) = true
f(true, false) = false
f(false, true) = false
The function needs to be consistent with all labeled examples
and should make the fewest mistakes on the unlabeled examples.

- Unlabeled examples

Feature_1	Feature_2	Class
false	false	?

5

Inductive Learning for Classification



- Labeled examples

Feature_1 = x	Class = f(x)
1.0	0.5
2.0	0.7
3.0	1.0
5.0	3.0

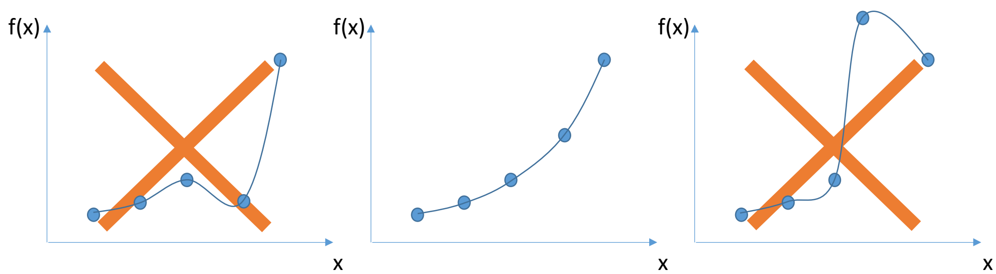
- Unlabeled examples

Feature_1 = x	Class = f(x)
4.0	?

6

Inductive Learning for Classification

- Function learning needs bias, i.e. to prefer some functions over others.



- Many students choose the function in the center.
- They prefer “simple” functions.

7

Example: Decision Tree (and Rule) Learning

Frog

8

Example: Decision Tree (and Rule) Learning

Is it grey?

yes → Elephant

no → Frog

Example: Decision Tree (and Rule) Learning

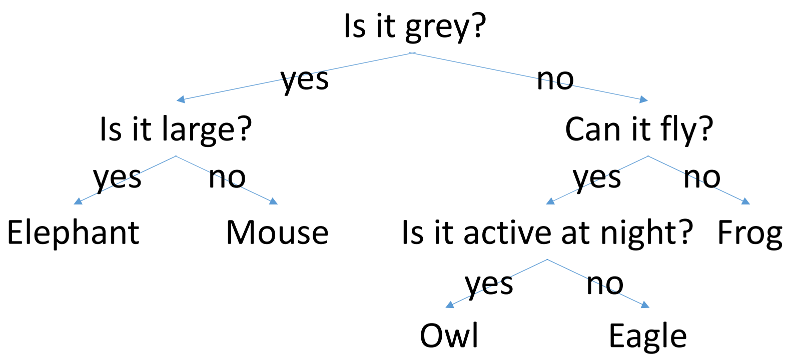
Is it grey?

yes → Elephant

no → Can it fly?

- yes → Eagle
- no → Frog

Example: Decision Tree (and Rule) Learning



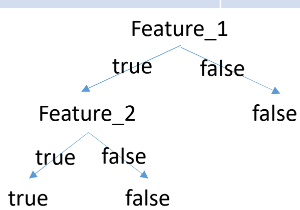
- Objective: Learn a decision tree
- Read off rules, such as: “If it is grey and not large then it is a mouse.”
- From now on: binary (feature and class) values only.

11

Example: Decision Tree (and Rule) Learning

- Labeled examples

Feature_1	Feature_2	Class
true	true	true
true	false	false
false	true	false



Feature_1 AND Feature_2 → Class
Feature_1 AND NOT Feature_2 → NOT Class
NOT Feature_1 → NOT Class

- Unlabeled examples (note: classification is very fast)

Feature_1	Feature_2	Class
false	false	? (guess: false)

12

Example: Decision Tree (and Rule) Learning

- Can decision trees represent all Boolean functions?
  f(Feature_1, …, Feature_n) ≡ some propositional sentence

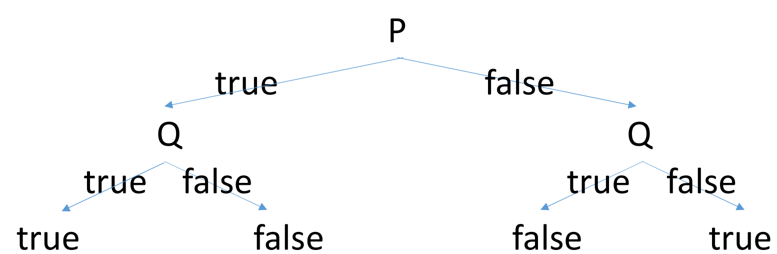
- This question is important because we need to find a decision tree that classifies all labeled examples correctly. This is always possible if decision trees can represent all Boolean functions.

13

Example: Decision Tree (and Rule) Learning

- Can decision trees represent all Boolean functions? – Yes.
  f(Feature_1, …, Feature_n) ≡ some propositional sentence

- Convert the propositional sentence into disjunctive normal form:
  Example: (P AND Q) OR (NOT P AND NOT Q)



14

Example: Decision Tree (and Rule) Learning

- There might be many decision trees that are consistent with all labeled examples. And they might differ in which classes they assign to the unlabeled examples. Which one to choose? (Especially since one does not know which one makes the fewest mistakes on the unlabeled examples.)

15

Example: Decision Tree (and Rule) Learning

- Function learning needs bias, i.e. to prefer some functions over others.
- Occam's razor: "Small is beautiful."
- Here: Prefer small decision trees over large ones (e.g. with respect to their depth, their number of nodes, or (used here) their average number of feature tests to determine the class).
- Reason: The functions encountered in the real world are often simple.
- That makes sense since simple explanations of natural phenomena are often the best ones, such as Kepler's three laws of planetary motion.

16

Example: Decision Tree (and Rule) Learning

- Function learning needs bias, i.e. to prefer some functions over others.
- Occam's razor: "Small is beautiful."
- Here: Prefer small decision trees over large ones (e.g. with respect to their depth, their number of nodes, or (used here) their average number of feature tests to determine the class).
- Reason: ~~The functions encountered in the real world are often simple.~~
- Real reason: There are fewer small decision trees than large ones. Thus, there is only a small chance that ANY small decision tree that does not represent the correct function is consistent with all labeled examples.
- Problem: Finding the smallest decision tree that is consistent with all labeled examples is NP-hard. So, we just try to find a small decision tree.

17

Example: Decision Tree (and Rule) Learning

- Real reason: There are fewer small decision trees than large ones. Thus, there is only a small chance that ANY small decision tree that does not represent the correct function is consistent with all labeled examples.
- In a country with 10 cities, if the majority of the population of a city voted for the winning president in the past 10 elections, perhaps they represent the "average citizen" of the country well.
- In a country with 10,000 cities, if the majority of the population of a city voted for the winning president in the past 10 elections, it could just be by chance. For example, if every citizen voted randomly for one of two candidates in the past 10 elections, there is still a good chance that there exists a city where the majority of the population voted for the winning president in the past 10 elections, just because there are so many cities.

18

ID3 Algorithm

	Feature_1	Feature_2	Feature_3	Feature_4	Class
E(xample) 1	true	true	false	true	true
E(xample) 2	true	false	false	false	true
E(xample) 3	true	true	true	true	false
E(xample) 4	true	true	true	false	false

19

ID3 Algorithm

- The trivial decision trees (“always true” or “always false”) do not work here.

	Feature_1	Feature_2	Feature_3	Feature_4	Class
E(xample) 1	true	true	false	true	true
E(xample) 2	true	false	false	false	true
E(xample) 3	true	true	true	true	false
E(xample) 4	true	true	true	false	false

true

This decision tree does not work here since the examples do not all have class true.

false

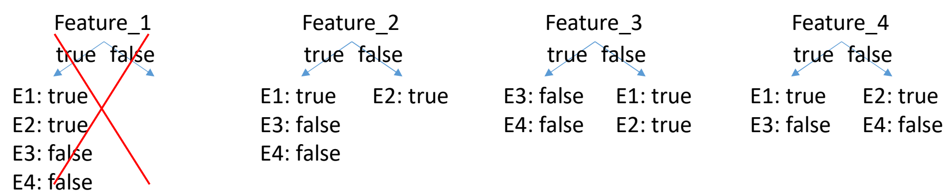
This decision tree does not work here since the examples do not all have class false.

20

ID3 Algorithm

- Put the most discriminating feature at the root.

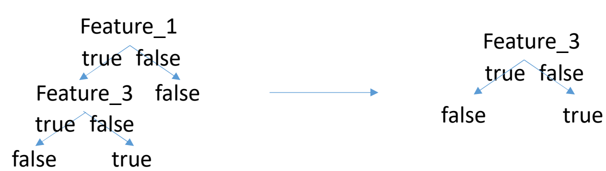
	Feature_1	Feature_2	Feature_3	Feature_4	Class
E(xample) 1	true	true	false	true	true
E(xample) 2	true	false	false	false	true
E(xample) 3	true	true	true	true	false
E(xample) 4	true	true	true	false	false



21

ID3 Algorithm

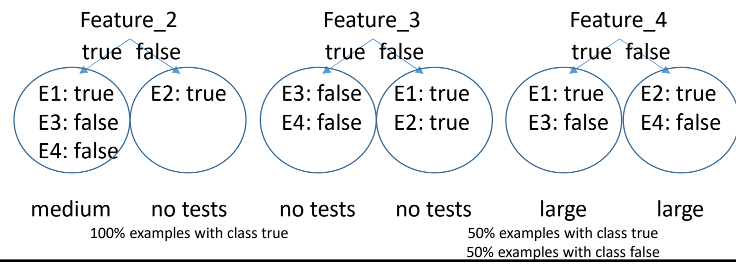
- Putting Feature_1 at the root is not helpful at all since all labeled examples have the same value for Feature_1.
- If we eventually find a decision tree that is consistent with all labeled examples, then we can decrease the average number of feature tests to determine the class by deleting the root.



22

ID3 Algorithm

- What's the average number of feature tests to determine the class in the following cases:
  - 1,00 (= 100%) examples with class true – no feature tests!
  - 999 examples with class true, 1 example with class false – likely: a very small number
  - 500 (= 50%) examples with class true, 500 (= 50%) examples with class false – likely: a large number
  - 1 example with class true, 999 examples with class false – likely: a very small number
  - 1,000 examples with class false – no feature tests!



23

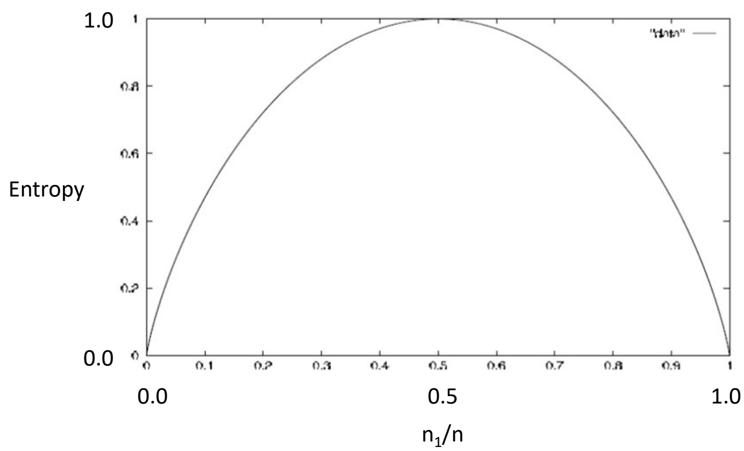
ID3 Algorithm

- We use the entropy as a measure that we assume to be proportional to the average number of feature tests to determine the class, which we are trying to minimize (without guarantees).
- Assume that there are n examples and that n_i examples have class i.
- Entropy = $- \Sigma_i (n_i/n) \log_2 (n_i/n)$
- The entropy is zero if no feature tests are necessary.
- Remember that $\log_2 x = \ln x / \ln 2 = \log_{10} x / \log_{10} 2$.
- Remember that $\lim_{x \to 0} (x \log_2 x) = 0$. Thus, "$0 \log_2 0 = 0$."

24

ID3 Algorithm

- In our case, there are 2 classes.



25

ID3 Algorithm

- Put the feature at the root that results in the smallest average entropy after splitting the examples.

Feature_2

true false

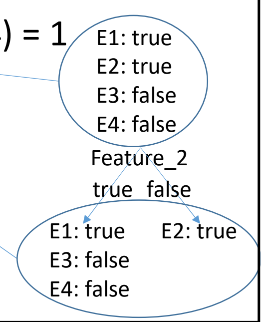
E1: true E2: true
E3: false
E4: false

- Left branch:
  - 3 out of 4 examples go down the left branch.
  - The entropy of the 3 examples is $-(1/3 \log_2 (1/3) + 2/3 \log_2 (2/3)) = 0.9182$.
- Right branch:
  - 1 out of 4 examples go down the right branch.
  - The entropy of the 1 example is –(1/1 log2 (1/1) + 0/1 log2 (0/1)) = 0.
- The average entropy after splitting the examples is ¾ 0.9182 + ¼ 0 = 0.6887.

26

ID3 Algorithm

- The textbook does not pick the feature that results in the smallest average entropy. Instead, it (equivalently) picks the feature that results in the largest information gain, which is the entropy of the examples (= before splitting them) minus the average entropy after splitting them.
- The entropy of the examples is $-(2/4 \log_2 2/4 + 2/4 \log_2 2/4) = 1$
- The average entropy after splitting them is 0.6887.
- The information gain is 1 – 0.6887 = 0.3113

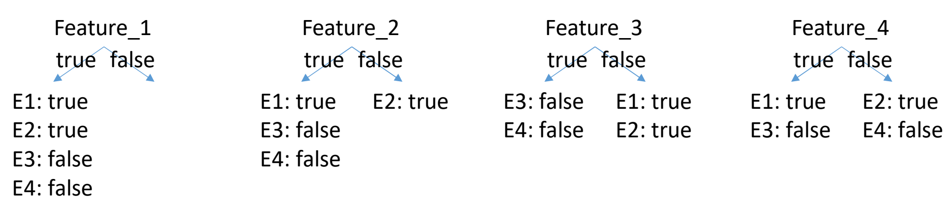


27

ID3 Algorithm

- Put the feature at the root that results in the smallest average entropy.

	Feature_1	Feature_2	Feature_3	Feature_4	Class
E(xample) 1	true	true	false	true	true
E(xample) 2	true	false	false	false	true
E(xample) 3	true	true	true	true	false
E(xample) 4	true	true	true	false	false



average entropy: 1.00 0.69 0.00 1.00

28

ID3 Algorithm

- Put the feature at the root that results in the smallest average entropy.

Feature_3
true false
E3: false E1: true
E4: false E2: true

Feature_3
true false
false true

29

ID3 Algorithm

	Feature_1	Feature_2	Feature_3	Feature_4	Class
E(xample) 1	true	true	false	true	true
E(xample) 2	true	false	false	false	true
E(xample) 3	true	true	true	true	false
E(xample) 4	true	true	true	false	false

Feature_3
true false
false true

Feature_1	Feature_2	Feature_3	Feature_4	Class
false	false	false	false	? (guess: true)
true	true	true	true	? (guess: false)

30

ID3 Algorithm: Complete Example

	Feature_1	Feature_2	Class
E(xample) 1	true	true	true
E(xample) 2	true	false	false
E(xample) 3	false	true	false

Feature_1
true → E1: true, E2: false
false → E3: false

Feature_2
true → E1: true, E3: false
false → E2: false

average entropy: 2/3 1 + 1/3 0 = 2/3   2/3 1 + 1/3 0 = 2/3

We have a tie that we can break arbitrarily.

31

ID3 Algorithm: Complete Example

	Feature_1	Feature_2	Class
E(xample) 1	true	true	true
E(xample) 2	true	false	false
E(xample) 3	false	true	false

Feature_1
true → E1: true, E2: false
false → E3: false

Now, we solve this classification problem recursively.

Feature_1
true → E1: true, E2: false
false → false

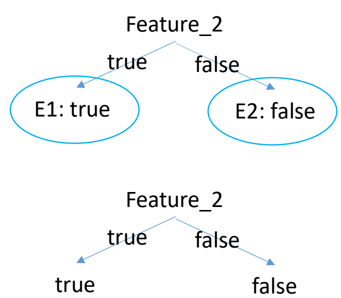
Now, we solve this classification problem recursively.

32

ID3 Algorithm: Complete Example

	~~Feature_1~~	Feature_2	Class
E(xample) 1	~~true~~	true	true
E(xample) 2	~~true~~	false	false

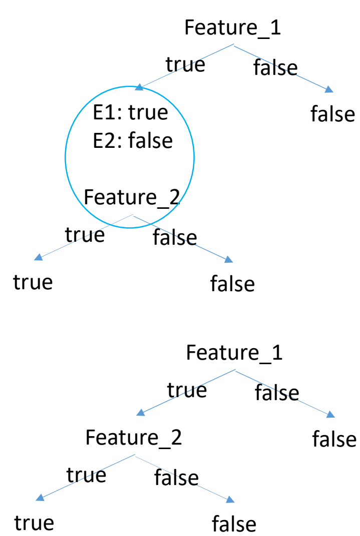
Now, we solve this classification problem recursively.



Now, we solve this classification problem recursively.

33

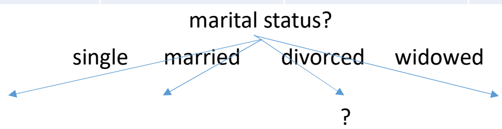
ID3 Algorithm: Complete Example



34

Issues with Multi-Valued Features, Issues with Missing Feature Values

How old are they?	What is their current salary per year?	Do they have a savings account?	What is their marital status?	…	Would you issue a credit card to them?
52	$150,000	yes	widowed	…	yes
40	$50,000	no	married	…	no
20	$60,000	yes	married	…	yes
31	$20,000	yes	single	…	yes

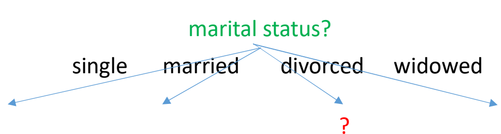


How old are they?	What is their current salary per year?	Do they have a savings account?	What is their marital status?	…	Would you issue a credit card to them?
26	$40,000	no	divorced	…	?

35

Issues with Multi-Valued Features, Issues with Missing Feature Values

- There is no labeled example that gets here during the construction of the decision tree. Thus, it is unclear how that subtree of the decision tree should look like. What to do?
- Find a common-sense rule that makes the best guess. For example, predict the majority class of all labeled examples that get here during the construction of the decision tree.



36

Example: Decision Tree (and Rule) Learning

- Want to play around with decision tree learning?
- Go here: http://aispace.org/dTree/

37