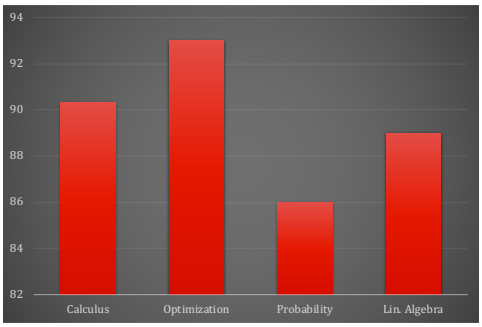
Machine Learning for Intelligent Systems

Lecture 3: Supervised Learning and Decision Trees

Reading: UML 18-18.2

Instructors: Nika Haghtalab (this time) and Thorsten Joachims

Placement Exam



Example: Apple Harvest Festival

Learn to classify **tasty apples** based on experience.
→ Training set: Apples you have tasted in the past, their features and whether they were tasty.
→ You see an apple, would you buy it?

	Features				Label
Apple	**Farm**	**Color**	**Size**	**Firmness**	**Tasty?**
#1	A	Red	Medium	Soft	No
#2	A	Green	Small	Crunchy	No
#3	A	Red	Medium	Soft	No
#4	B	Red	Large	Crunchy	Yes
#5	B	Green	Large	Crunchy	Yes
#6	B	Green	Small	Crunchy	No
#7	A	Red	Small	Soft	No
#8	B	Red	Small	Crunchy	Yes
#9	B	Green	Medium	Soft	No
#10	A	Red	Small	Crunchy	Yes
#11	A	Red	Medium	Crunchy	?

Supervised Learning

Instance Space:
Instance space X including feature representation.
E.g., $X = \{A, B\} \times \{red, green\} \times \{large, medium, small\} \times \{crunchy, soft\}$.

Target Attributes (Labels):
A set Y of labels. E.g., $Y = \{Tasty, Not\ Tasty\}$ or $Y = \{Yes, No\}$.

Hidden target function:
An unknown function, $f: X \to Y$, e.g., how an apples features correlate with its tastiness in real life.

Training Data:
A set of labeled pairs $(x, f(x)) \in X \times Y$ that we have seen before, e.g., we have tasted a $(A, red, large, crunchy)$ apple before that was Tasty.

Informal: Our main goal

Given a large enough number of training examples, learn a hypothesis $h: X \to Y$ that approximates $f(\cdot)$

Hypothesis Space

Hypothesis space:
The set H of functions we consider when looking for $h: X \to Y$.

Example: All hypotheses that are AND of feature-values:
$(Farm = whatever) \wedge (color = red) \wedge (size = whatever) \wedge (firmness = Crunchy)$

Apple	**Farm**	**Color**	**Size**	**Firmness**	**Tasty?**
#1	A	Red	Medium	Soft	No
#2	A	Green	Small	Crunchy	No
#3	A	Red	Medium	Soft	No
#4	B	Red	Large	Crunchy	Yes
#5	B	Green	Large	Crunchy	Yes
#6	B	Green	Small	Crunchy	No
#7	A	Red	Small	Soft	No
#8	B	Red	Small	Crunchy	Yes
#9	B	Green	Medium	Soft	No
#10	A	Red	Small	Crunchy	Yes

Consistency

Consistency

A function $h: X \to Y$ is **consistent with** a set of labeled training examples S if and only if $h(x) = y$ for all $(x, y) \in S$.

Example:
Is $(Farm = whatever) \wedge (color = red) \wedge (size = whatever) \wedge (firmness = Crunchy)$ consistent with the table of apples?

Is it consistent with the apples that came from farm A?

Apple	**Farm**	**Color**	**Size**	**Firmness**	**Tasty?**
#1	A	Red	Medium	Soft	No
#2	A	Green	Small	Crunchy	No
#3	A	Red	Medium	Soft	No
#7	A	Red	Small	Soft	No
#10	A	Red	Small	Crunchy	Yes

Inductive Learning

More Formal: Our main goal

Given a large enough number of training examples,
and a hypothesis space H,
learn a hypothesis $h \in H$ that approximates $f(\cdot)$

Our hope: There is a $h \in H$ that approximates f.

Our strategy: Our training examples are representative of the reality
→ We have seen **many** examples.
→ They were selected without any bias.
→Any hypothesis that is consistent with training data would be accurate on unobserved instances.

Find $h \in H$ such that is consistent with S. ⇒ $h(x) = f(x)$ for almost all $x \in X$

Using Consistency in Learning

Idea: On a training set S, return any $h \in H$ that is consistent with it.

Version Space:

The **version space** $VS(H, S)$ is a subset of functions from H that is **consistent with** S.

Example:
For the following set of training examples S what and H that is all hypotheses that are AND of feature-value settings. What is $VS(H, S)$?

Apple	Farm	Color	Size	Firmness	Tasty?
#1	A	Red	Medium	Soft	No
#2	A	Green	Small	Crunchy	No
#3	A	Red	Medium	Soft	No
#7	A	Red	Small	Soft	No
#10	A	Red	Small	Crunchy	Yes

Computing the Version Space

At a high level:
List all hypothesis in H,
Remove them if they are not consistent with some example in S.

List-then-Eliminate Algorithm

- Initialize $VS = H$
- For each training example $(x, y) \in S$,
  - remove any $h \in H$ such that $h(x) \neq y$.
- Output VS.

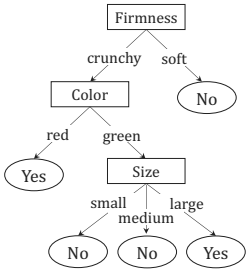
Takeaway: The algorithm works, but
→ Keeping track of the version space is time and space consuming.
→ Only need one consistent
→ Build a consistent hypothesis directly.

Decision Trees

Internal nodes:
Test one feature, e.g., color.

Branches from an internal node:
Possible values of the feature that's being tested, e.g., {red, green}.

Leaf nodes:
The label of instances whose features correspond to the root-to-leaf path, e.g., Yes or No.

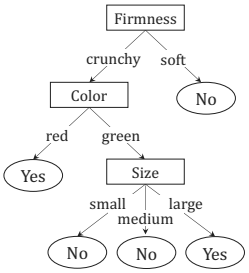


Using a Decision Tree

Let function f be the following decision tree.
1. What is $f((B, Green, Large, Crunchy))$?
2. What is $f((B, Red, Large, Soft))$?
3. What is $f((A, Green, Small, Crunchy))$?

Is f consistent with the training set in the table.

Apple	Farm	Color	Size	Firmness	Tasty?
#1	A	Red	Medium	Soft	No ✓
#2	A	Green	Small	Crunchy	No ✓
#3	A	Red	Medium	Soft	No ✓
#4	B	Red	Large	Crunchy	Yes ✓
#5	B	Green	Large	Crunchy	Yes ✓
#6	B	Green	Small	Crunchy	No ✓
#7	A	Red	Small	Soft	No ✓
#8	B	Red	Small	Crunchy	Yes ✓
#9	B	Green	Medium	Soft	No ✓
#10	A	Red	Small	Crunchy	Yes ✓



Making a Tree

Example: What is the decision tree corresponding to functions
1. $(firmness = Crunchy)$?
2. $(firmness = Crunchy) \wedge (color = Red)$?
3. $(color = Red) \vee (size = Large)$?
4. $(firmness = Crunchy) \wedge ((color = Red) \vee (size = Large))$?

Top-Down Induction of Decision Trees

Idea:

- Don't use List-then-Eliminate too inefficient.
- Instead grow a good decision tree to start with.
  - → We grow from the root to the leaves
  - → Repeatedly take an exiting leaf that does not include a definitive label and replace it with an internal node.

TD-IDT (S, y)

- If all examples in S have the same label y
  - →Make a leaf with label y.
- Else
  - → Pick feature A
  - → For each value a_i of A, make a child node such that $S_i = \{ (x, y) \in S$: feature A of x has value $a_i\}$
  - →Make a tree with A as a root and TD-IDT(S_i, y) as subtrees.

Grow a consistent DT

TD-IDT (S, y)

- If all examples in S have the same label y
  - →Make a leaf with label y.
- Else
  - → Pick feature A
  - → For each value a_i of A, make a child node such that $S_i = \{ (x, y) \in S$: feature A of x has value $a_i\}$
  - →Make a tree with A as a root and TD-IDT(S_i, y) as subtrees.

Apple	Farm	Color	Size	Firmness	Tasty?
#1	A	Red	Medium	Soft	No
#2	A	Green	Small	Crunchy	No
#3	A	Red	Medium	Soft	No
#4	B	Red	Large	Crunchy	Yes
#5	B	Green	Large	Crunchy	Yes
#6	B	Green	Small	Crunchy	No
#7	A	Red	Small	Soft	No
#8	B	Red	Small	Crunchy	Yes
#9	B	Green	Medium	Soft	No
#10	A	Red	Small	Crunchy	Yes

Which Split?

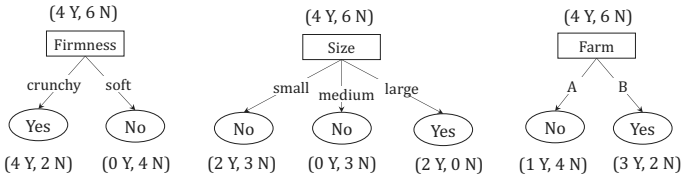
How should we pick feature A for splitting.

→ How does the prediction improve after the split

→ At each node, take the number of positive and negative instances. E.g. (4 Yes, 2 No).

→ If we were to stop, best to predict the majority label. E.g., Yes.

→ How do we achieve a more definite prediction?

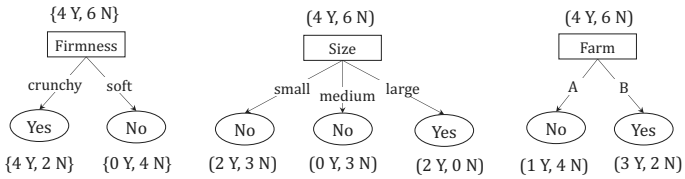


Which Split?

Different ways to measure this:

- Attribute that minimizes the number of error:
  - Before split: $\mathrm{Err}(S)$ = size of the minority label.
  - After split: $\mathrm{Err}(S|A) = \sum_{a_i} \mathrm{Err}(S_i)$

- Attribute that minimizes entropy (maximize Information Gain).
  - Before split: $H(S) = -\sum p(x) \log(p(x))$
  - After split: $\mathrm{I}(S|A) = \sum_{a_i} p(S_i)\, H(S_i)$



Example: Text Classification

- Task: Learn rule that classifies Reuters Business News
  - Class +: "Corporate Acquisitions"
  - Class -: Other articles
  - 2000 training instances
- Representation:
  - Boolean attributes, indicating presence of a keyword in article
  - 9947 such keywords (more accurately, word "stems")

LAROCHE STARTS BID FOR NECO SHARES +

Investor David F. La Roche of North Kingstown, R.I., said he is offering to purchase 170,000 common shares of NECO Enterprises Inc at 26 dlrs each. He said the successful completion of the offer, plus shares he already owns, would give him 50.5 pct of NECO's 962,016 common shares. La Roche said he may buy more, and possible all NECO shares. He said the offer and withdrawal rights will expire at 1630 EST/2130 gmt, March 30, 1987.

SALANT CORP 1ST QTR FEB 28 NET −

Oper shr profit seven cts vs loss 12 cts. Oper net profit 216,000 vs loss 401,000. Sales 21.4 mln vs 24.9 mln. NOTE: Current year net excludes 142,000 dlr tax credit. Company operating in Chapter 11 bankruptcy.

Decision Tree for "Corporate Acq."

```
• vs = 1: -
• vs = 0:
• |  export = 1:
...
• |  export = 0:
• |  |  rate = 1:
• |  |  |  stake = 1: +
• |  |  |  stake = 0:
• |  |  |  |  debenture = 1: +
• |  |  |  |  debenture = 0:
• |  |  |  |  |  takeover = 1: +
• |  |  |  |  |  takeover = 0:
• |  |  |  |  |  |  file = 0: -
• |  |  |  |  |  |  file = 1:
• |  |  |  |  |  |  |  share = 1: +
• |  |  |  |  |  |  |  share = 0: -
... and many more
```

Learned tree:

- has 437 nodes
- is consistent

Accuracy of learned tree:

- 11% error rate

Note: word stems expanded for improved readability.