CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2020

HAVERFORD
COLLEGE

Admin

- Lab 2 due **Tuesday**
  - Post on Piazza!
- TA hours **Sunday 8:30-10pm** (Fiona)
- Next office hours **Mon 9:45-11am**
- We are working on getting a **peer tutor**

Video on if possible!!

Outline for September 18

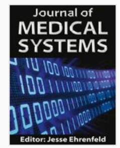
- Recap high level Decision Tree algorithm
- Entropy and information gain
- Continuous features
- Lab 2 implementation suggestions

Outline for September 18

- Recap high level Decision Tree algorithm
- Entropy and information gain
- Continuous features
- Lab 2 implementation suggestions

Real-World Examples

- Medical diagnostics



Journal of Medical Systems
October 2002, Volume 26, Issue 5, pp 445–463 | Cite as

Decision Trees: An Overview and Their Use in Medicine

Authors | Authors and affiliations

Vili Podgorelec, Peter Kokol, Bruno Stiglic, Ivan Rozman

- Credit risk analysis

Computational Economics
April 2000, Volume 15, Issue 1–2, pp 107–143 | Cite as

Credit Risk Assessment Using Statistical and Machine Learning: Basic Methodology and Risk Modeling Applications

Authors | Authors and affiliations

J. Galindo, P. Tamayo

- Modeling calendar scheduling preferences

Decision Trees in Chemistry reactions

- Example of decision trees in practice
- Use decision trees to interpret another ML algorithm (SVMs)

Machine-learning-assisted materials discovery using failed experiments

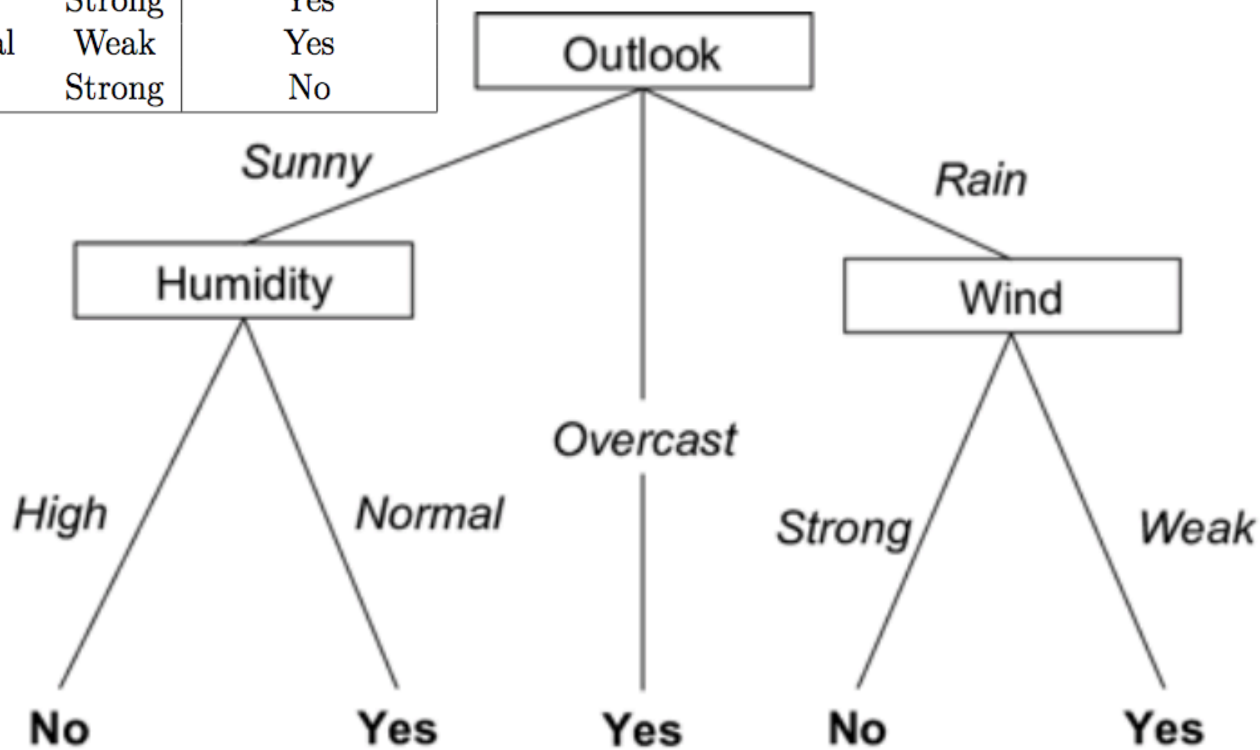
Paul Raccuglia, Katherine C. Elbert, Philip D. F. Adler, Casey Falk, Malia B. Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A. Friedler, Joshua Schrier & Alexander J. Norquist

Nature **533**, 73–76 (05 May 2016) | Download Citation

Optional Reading!

Handout 2

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	Sunny	Mild	Normal	Strong	Yes
$\boldsymbol{x}_{12}$	Overcast	Mild	High	Strong	Yes
$\boldsymbol{x}_{13}$	Overcast	Hot	Normal	Weak	Yes
$\boldsymbol{x}_{14}$	Rain	Mild	High	Strong	No



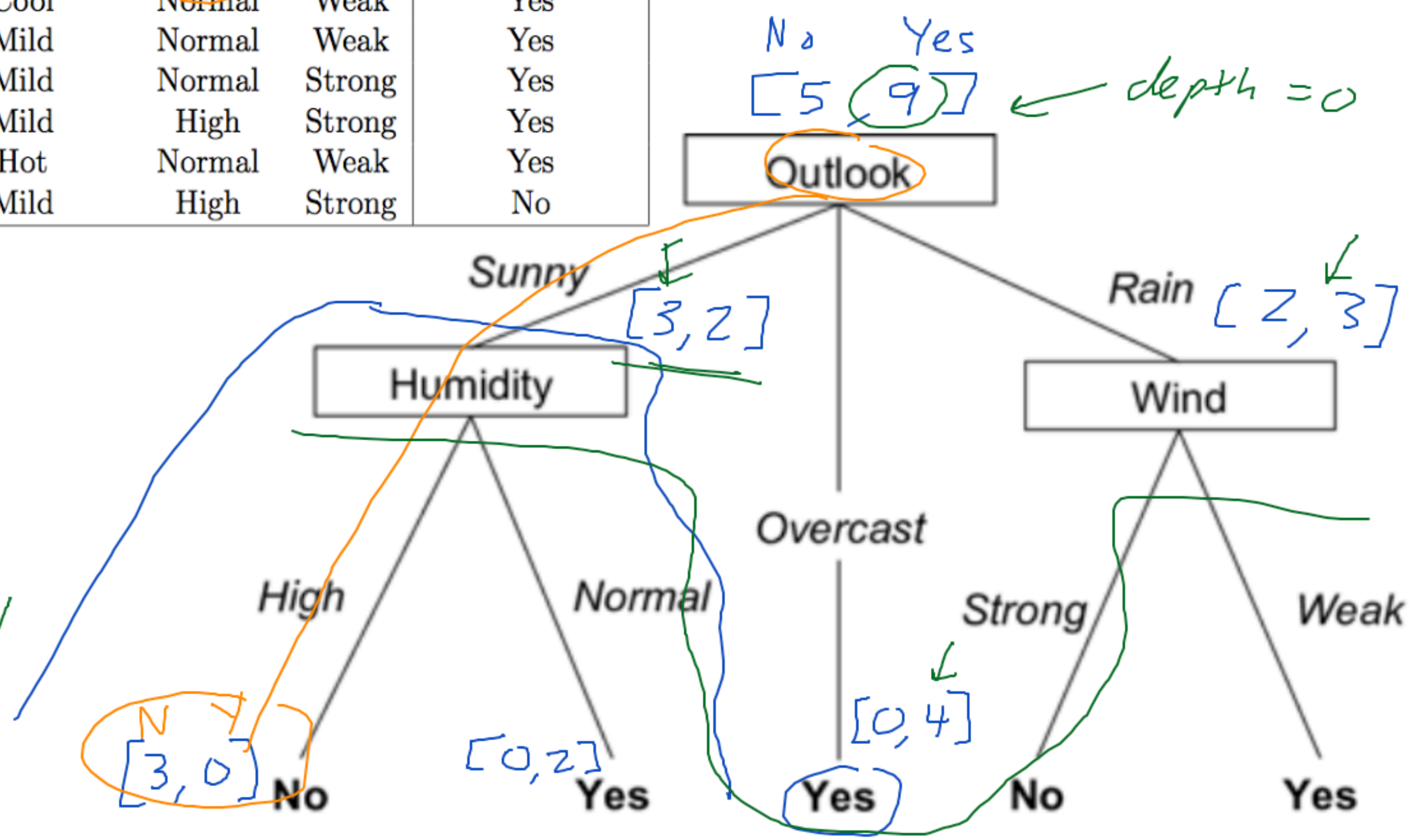
Handout 2

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	Sunny	Mild	Normal	Strong	Yes
$\boldsymbol{x}_{12}$	Overcast	Mild	High	Strong	Yes
$\boldsymbol{x}_{13}$	Overcast	Hot	Normal	Weak	Yes
$\boldsymbol{x}_{14}$	Rain	Mild	High	Strong	No

$d=2$ train acc $= \underline{100\%}$

$d=1$ train acc $= \frac{10}{14}$

depth = 1



Recursive algorithm: Partition data structure

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	Sunny	Mild	Normal	Strong	Yes
$\boldsymbol{x}_{12}$	Overcast	Mild	High	Strong	Yes
$\boldsymbol{x}_{13}$	Overcast	Hot	Normal	Weak	Yes
$\boldsymbol{x}_{14}$	Rain	Mild	High	Strong	No

Recursive algorithm: Partition data structure

$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	Sunny	Mild	Normal	Strong	Yes
$\boldsymbol{x}_{12}$	Overcast	Mild	High	Strong	Yes
$\boldsymbol{x}_{13}$	Overcast	Hot	Normal	Weak	Yes
$\boldsymbol{x}_{14}$	Rain	Mild	High	Strong	No

Recursive algorithm: Partition data structure

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	Sunny	Mild	Normal	Strong	Yes
$\boldsymbol{x}_{12}$	Overcast	Mild	High	Strong	Yes
$\boldsymbol{x}_{13}$	Overcast	Hot	Normal	Weak	Yes
$\boldsymbol{x}_{14}$	Rain	Mild	High	Strong	No

$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	Sunny	Mild	Normal	Strong	Yes

$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_{12}$	Overcast	Mild	High	Strong	Yes
$\boldsymbol{x}_{13}$	Overcast	Hot	Normal	Weak	Yes

Recursive algorithm: Partition data structure

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
x_1	Sunny	Hot	High	Weak	No
x_2	Sunny	Hot	High	Strong	No
x_3	Overcast	Hot	High	Weak	Yes
x_4	Rain	Mild	High	Weak	Yes
x_5	Rain	Cool	Normal	Weak	Yes
x_6	Rain	Cool	Normal	Strong	No
x_7	Overcast	Cool	Normal	Strong	Yes
x_8	Sunny	Mild	High	Weak	No
x_9	Sunny	Cool	Normal	Weak	Yes
x_{10}	Rain	Mild	Normal	Weak	Yes
x_{11}	Sunny	Mild	Normal	Strong	Yes
x_{12}	Overcast	Mild	High	Strong	Yes
x_{13}	Overcast	Hot	Normal	Weak	Yes
x_{14}	Rain	Mild	High	Strong	No

x_1	Sunny	Hot	High	Weak	No
x_2	Sunny	Hot	High	Strong	No
x_8	Sunny	Mild	High	Weak	No
x_9	Sunny	Cool	Normal	Weak	Yes
x_{11}	Sunny	Mild	Normal	Strong	Yes

x_3	Overcast	Hot	High	Weak	Yes
x_7	Overcast	Cool	Normal	Strong	Yes
x_{12}	Overcast	Mild	High	Strong	Yes
x_{13}	Overcast	Hot	Normal	Weak	Yes

x_4	Rain	Mild	High	Weak	Yes
x_5	Rain	Cool	Normal	Weak	Yes
x_6	Rain	Cool	Normal	Strong	No
x_{10}	Rain	Mild	Normal	Weak	Yes
x_{14}	Rain	Mild	High	Strong	No

Recursive algorithm: Partition data structure

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	Sunny	Mild	Normal	Strong	Yes
$\boldsymbol{x}_{12}$	Overcast	Mild	High	Strong	Yes
$\boldsymbol{x}_{13}$	Overcast	Hot	Normal	Weak	Yes
$\boldsymbol{x}_{14}$	Rain	Mild	High	Strong	No

temp

$\boldsymbol{x}_1$	~~Sunny~~	Hot	High	Weak	No
$\boldsymbol{x}_2$	~~Sunny~~	Hot	High	Strong	No
$\boldsymbol{x}_8$	~~Sunny~~	Mild	High	Weak	No
$\boldsymbol{x}_9$	~~Sunny~~	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	~~Sunny~~	Mild	Normal	Strong	Yes

$\boldsymbol{x}_3$	~~Overcast~~	Hot	High	Weak	Yes
$\boldsymbol{x}_7$	~~Overcast~~	Cool	Normal	Strong	Yes
$\boldsymbol{x}_{12}$	~~Overcast~~	Mild	High	Strong	Yes
$\boldsymbol{x}_{13}$	~~Overcast~~	Hot	Normal	Weak	Yes

temp

$\boldsymbol{x}_4$	~~Rain~~	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	~~Rain~~	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	~~Rain~~	Cool	Normal	Strong	No
$\boldsymbol{x}_{10}$	~~Rain~~	Mild	Normal	Weak	Yes
$\boldsymbol{x}_{14}$	~~Rain~~	Mild	High	Strong	No

Partition class

```
class Example:

    def __init__(self, features, label):
        """Helper class (like a struct) that stores info about each example."""
        # dictionary. key=feature name: value=feature value for this example
        self.features = features
        self.label = label # in {-1, 1}

class Partition:

    def __init__(self, data, F):
        """Store information about a dataset"""
        self.data = data # list of examples
        # dictionary. key=feature name: value=set of possible values
        self.F = F
        self.n = len(self.data)
```

Partition class

DTree

class

self.children = {}

self.children["sun"] = DTree(....

outlook

```python
class Example:

    def __init__(self, features, label):
        """Helper class (like a struct) that stores info about each example."""
        # dictionary. key=feature name: value=feature value for this example
        self.features = features
        self.label = label # in {-1, 1}

class Partition:

    def __init__(self, data, F):
        """Store information about a dataset"""
        self.data = data # list of examples
        # dictionary. key=feature name: value=set of possible values
        self.F = F
        self.n = len(self.data)
```

F = { outlook : (sun, rain, overcast), }

key

value

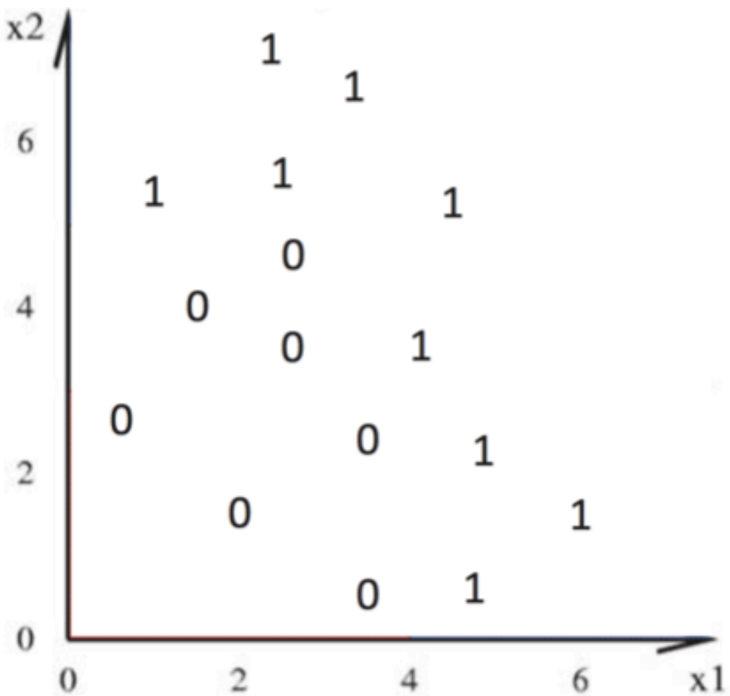
Tree

self.left = Tree(...)

self.right = Tree(...)

Handout 2: continuous features

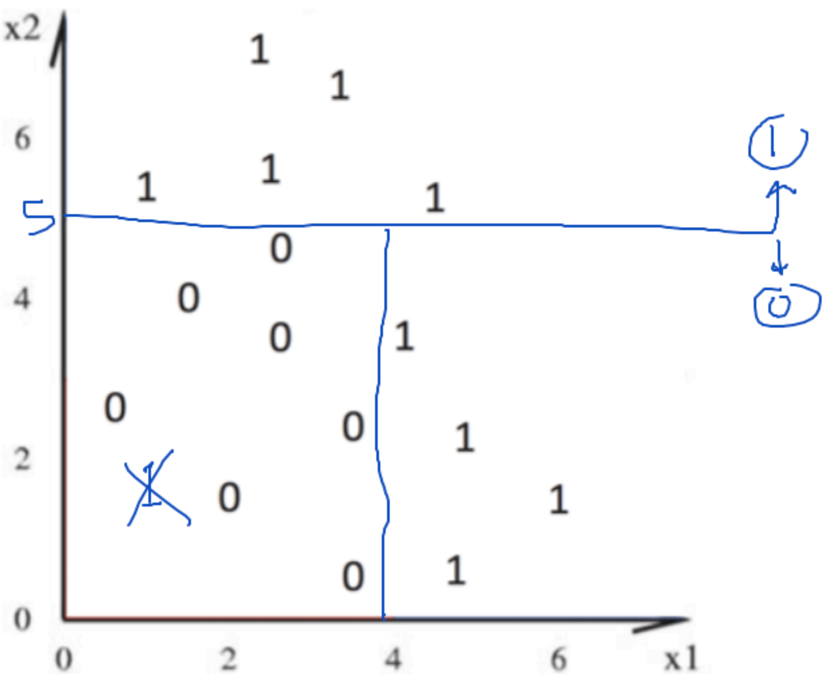
4. For the dataset below, the label $y \in \{0, 1\}$. What is n? What is p? Devise a decision tree for this data that perfectly classifies the given examples. Internal node labels should be of the form "$x_j \leq a$", where a is some constant.

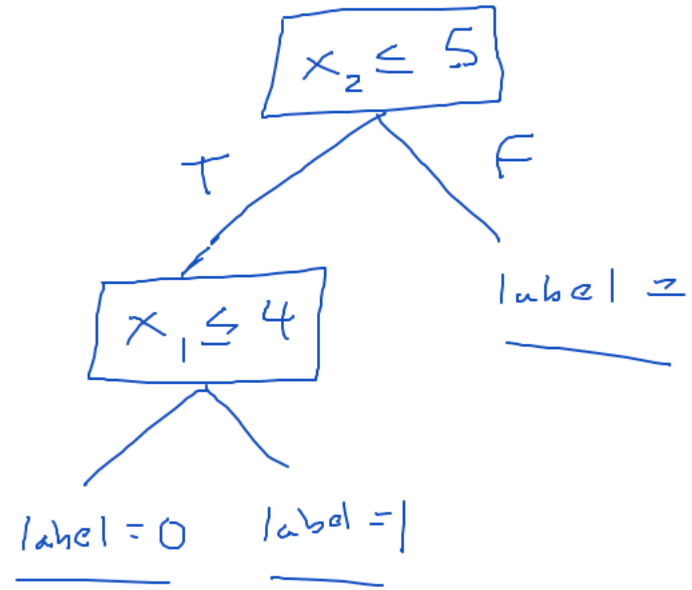


5. Repeat Question (2) for this decision tree (i.e. label each node with the "0" and "1" counts.)

Handout 2: continuous features

4. For the dataset below, the label $y \in \{0, 1\}$. What is n? What is p? Devise a decision tree for this data that perfectly classifies the given examples. Internal node labels should be of the form "$x_j \leq a$", where a is some constant.





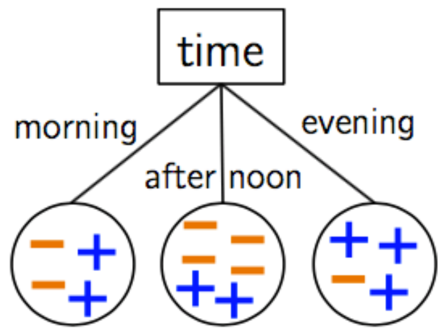
5. Repeat Question (2) for this decision tree (i.e. label each node with the "0" and "1" counts.)

Reading check-in: work individually for a few minutes

1. Match the decision tree component on the left with its corresponding data component on the right.

• internal nodes	class labels
• branches	feature names
• leaves	feature values

2. Say I am trying to predict if a student will like a course (+) or dislike it (−). One of the features is the time of day the course is offered. If I just choose this one feature and build a decision tree, here is how the training examples cluster at the leaves:



   (a) How would you classify a new example with value **evening** for the feature **time**?

   (b) What is the overall *training error* if I use the majority class label at each leaf?

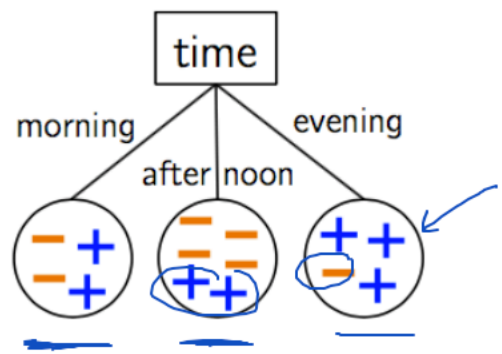
3. If a decision tree is overfitting, is the *depth* more likely to be low or high?

Reading check-in: work individually for a few minutes

1. Match the decision tree component on the left with its corresponding data component on the right.

   - internal nodes — class labels
   - branches — feature names
   - leaves — feature values

2. Say I am trying to predict if a student will like a course (+) or dislike it (−). One of the features is the time of day the course is offered. If I just choose this one feature and build a decision tree, here is how the training examples cluster at the leaves:



   (a) How would you classify a new example with value **evening** for the feature **time**?

   + (like)

   (b) What is the overall *training error* if I use the majority class label at each leaf?

   $$\text{error} = \frac{2+2+1}{14} = \frac{5}{14} \qquad \text{acc}: \frac{9}{14}$$

3. If a decision tree is overfitting, is the *depth* more likely to be low or high?

   high

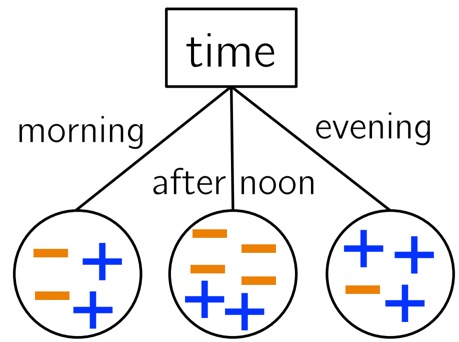
Reading Check-in

1)
- internal nodes
- branches
- leaves

class labels
feature names
feature values

2) (a) +

(b) 5/14



3) high

Outline for September 18

- Recap high level Decision Tree algorithm
- Entropy and information gain
- Continuous features
- Lab 2 implementation suggestions

Entropy

Idea: avg # bits needed to transmit info

Year	prob (p)	Idea	Cumulative prob	Binary		
Senior	0.5					
Junior	0.25					
Sophomore	0.125					
First year	0.125					

Entropy

Idea: avg # bits needed to transmit info

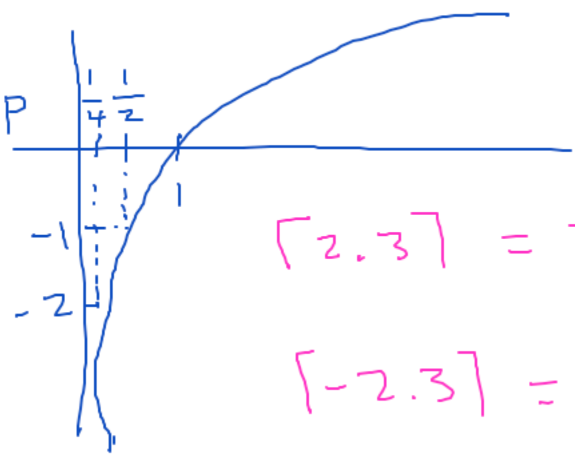
Year	prob (p)	~~Idea~~	Cumulative prob	Binary	$-\lceil \log_2(p) \rceil$	code
Senior	0.5	~~0~~	0	0.000..	1	0
Junior	0.25	~~1~~	0.5	0.100..	2	10
Sophomore	0.125	~~01~~	0.75	0.110..	3	110
First year	0.125	~~10~~	0.875	0.111..	3	111

110 110 10 0 0 111 0

binary ⇒ decimal

$$\cdots \square \cdot 2^2 + \square \cdot 2^1 + \square \cdot 2^0 + \square \cdot 2^{-1} + \square \cdot 2^{-2} \cdots$$

$\frac{1}{2}$ $\frac{1}{4}$



$\lceil 2.3 \rceil = 3$

$\lceil -2.3 \rceil = -3$

$5 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$

$\Rightarrow 101$

$5.5 \Rightarrow 101.1$

Entropy

$$H(Y) = - \sum_{c \in vals(Y)} p(Y=c) \overbrace{\log_2 p(Y=c)}^{\# \text{ bits}}$$

$$H(year) = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \left(\frac{1}{8} \cdot 3\right) 2$$

$$= \underline{1.75} \text{ bits}$$

$$= \cancel{\frac{1+2+3+3}{4}} \qquad = 2.25$$

Conditional Entropy

one feature name

value

$$H(Y|X=v) = -\sum_{c \in vals(Y)} p(Y=c|X=v)\log_2 p(Y=c|X=v)$$

A B

A + B

$$p(\text{yes}|\text{outlook}=\text{sun}) = \frac{P(\text{yes AND sun})}{P(\text{sun})}$$

B

v = sun

$\frac{5}{14}$

sun
rain
over
cast

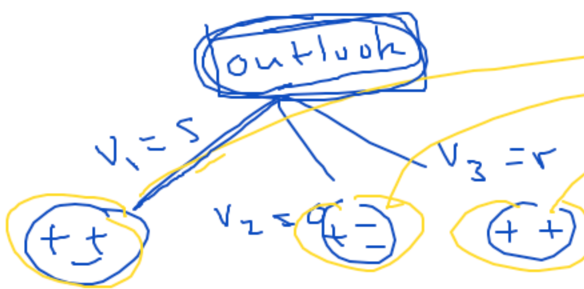
label $\in$ {yes, no}

outlook

low

$$H(Y|X) = \sum_{v \in vals(X)} P(X=v)\, H(Y|X=v)$$

entropy of label



Info Gain : $H(Y) - H(Y|X)$

high

Handout 4: work with your group!
(second question only)

Handout 4

Try first and then check your answers!

Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	Animated	Long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	Animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes

$\mathrm{P(Li = yes)} =$ 2/3

$\mathrm{H(Li)} =$ 0.92

$\mathrm{H(Li \mid T)} = 0.61$

$\mathrm{H(Li \mid Le)} = 0.61$

$\boxed{\mathrm{H(Li \mid D)} = 0.36}$ MIN ENTROPY

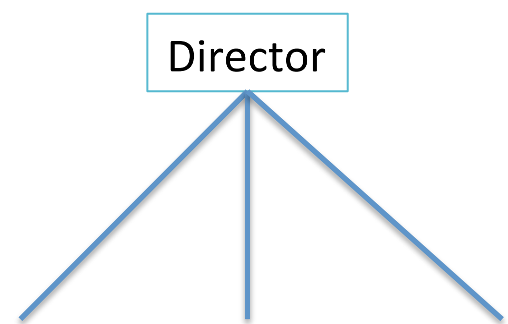
$\mathrm{H(Li \mid F)} = 0.85$

$\mathrm{Gain(Li,\ T)} =$ 0.92 – 0.61 = 0.31

$\mathrm{Gain(Li,\ Le)} =$ 0.92 – 0.61 = 0.31

$\boxed{\mathrm{Gain(Li,\ D)} = 0.92 - 0.36 = 0.56}$ MAX INFO GAIN

$\mathrm{Gain(Li,\ F)} =$ 0.92 – 0.85 = 0.07



Start of the tree

Outline for September 18

- Recap high level Decision Tree algorithm
- Entropy and information gain
- Continuous features
- Lab 2 implementation suggestions

Continuous Features

(do this for the TRAIN only!)

X	Y
10	Y
7	Y
8	N
3	Y
7	N
12	Y
2	Y

1) Sort examples based on given feature

2	3	7	7	8	10	12
Y	Y	Y	N	N	Y	Y

2) Different label with same feature value, collapse to "None"

2	3	7	8	10	12
Y	Y	None	N	Y	Y

3) Whenever label changes, make a feature (use avg)

2	3	7	8	10	12
Y	Y	None	N	Y	Y

X <= 7.5

X <= 5

X <= 9

Outline for September 18

- Recap high level Decision Tree algorithm
- Entropy and information gain
- Continuous features
- Lab 2 implementation suggestions

Implementation Suggestions

- Start slow with entropy! Build up function by function
- Think back to trees in data structures
- Distinguish between data (X,y) and options for data (values for each feature, classes for y)