Algorithms & Data Analysis

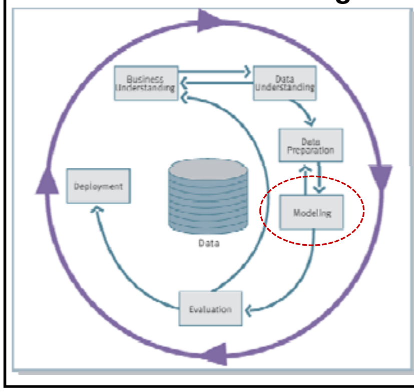
CISC 6950

Lecture 2

(these slides are based on the slides by Prof. F. Provost (Stern NYU) and E. Keogh (UC Riverside))

1

The Data Mining Process



Which customers should TelCo target with a special offer, prior to contract expiration?

2

What might a data mining model look like?

- There are different sorts of data mining. Here are just two examples

Rules: (a supervised segmentation)

- **If (income > $70K) & (age > 40) & (domicile="NE USA") then p(churn)=0.12**
- **If … then …**

Numeric function:

- **P(churn) = f(x1,x2,…, xk)**

3

What is a model?

A *simplified* representation* of reality created for a *specific purpose*.

**** simplification is based on some assumptions***

- Examples: map, blue prints
- Data Mining Example (from introduction module):
  "formula" for predicting probability of customer attrition at contract expiration

  →"classification model" or "class-probability estimation model"

4

Why model?

Progress from an intuitive approach to data-driven decision-making to one based on science & craft

- Frames data selection & acquisition
- Allows leverage of existing techniques & technology
- Improves consistency of analyses
- Helps to explore data interactively – understand impact of variables
- Helps with communication of results, "selling" ideas

5

Data Mining: Terminology

Induction (aka ***learning, inductive learning, model induction***)

A process by which a model (or other pattern) is generalized from factual data.

Example:

By analyzing data on past credit customers who have and have not defaulted one can generalize what characterizes customers who are likely to default, as opposed to those who are not.

Name	Balance	Age	Default

⇨

Pattern

If **Balance** >= 50K and **Age** > 45
Then **Default** = "no"
Else **Default** = "yes"

6

Data Mining: Terminology

Attributes Target

Name	Balance	Age	Default
Mike	123,000	30	Yes
Mary	51,100	40	Yes
Bill	68,000	55	No
Jim	74,000	46	No
Mark	23,000	47	Yes
Anne	100,000	49	No

Example

an example of this form sometimes is called a **"feature vector"**

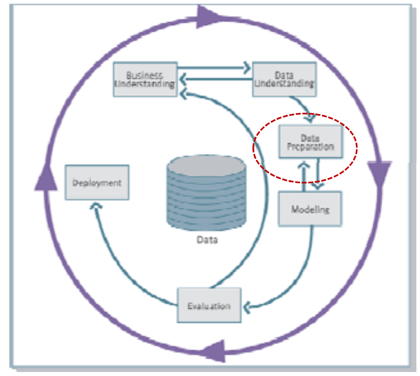
Example, Instance; Fact; Data point

typically described by a set of attributes (fields, variables, features) and a target variable (label).

A data set : A set of examples

table

The Data Mining Process



Data Mining: Terminology

A learner, inducer, induction algorithm

A method or algorithm used to generalize a model or pattern from a set of examples

Name	Balance	Age	Default
Mike	123,000	30	Yes
Mary	51,100	40	Yes
Bill	68,000	55	No
Jim	74,000	46	No
Mark	23,000	47	Yes
Anne	100,000	49	No

→ **Learner:** Induces a model from examples

↓

Classification Model

If **Balance** >= 50K and **Age** > 45
Then **Default** = "no"
Else **Default** = "yes"

Data Mining: Terminology

Contrast: regression modeling (rather than classification)

Target Variable

Name	Balance	Age	Order $
Mike	123,000	30	183
Mary	51,100	40	131
Bill	68,000	55	178
Jim	74,000	46	166
Mark	23,000	47	117
Anne	100,000	49	198

→ **Learner:** Linear regression

↓

Model

Amount = 0.002***Income**+2***Age**

Data Mining: Terminology

Supervised learning

- Model induction where the model describes a relationship between a set of independent attributes and a predefined dependent attribute – the "target"
- **AND,** the values for the target are available at induction time
- Most induction algorithms fall into the supervised learning category

Name	Balance	Age	Default
Mike	123,000	30	Yes
Mary	51,100	40	Yes
Bill	68,000	55	No
Jim	74,000	46	No
Mark	23,000	47	Yes
Anne	100,000	49	No

→ **Supervised learning of a classification model**

↓

Classification Model

If **Balance** >= 50K and **Age** > 45
Then **Default** = "no"
Else **Default** = "yes"

Training data, ***labeled*** data

Why trees?

- Decision trees, or *classification trees*, are one of the most popular data mining tools (along with linear/logistic regression)
- They're:
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap
- Almost all data mining packages include DTs
- They have advantages for model comprehensibility, which is important for:
  - model evaluation
  - communication to non-DM-savvy stakeholders

Data Mining: Terminology

A learner, inducer, induction algorithm

- A method or algorithm used to generalize a model or pattern from a set of examples

Name	Balance	Age	Default
Mike	123,000	30	Yes
Mary	51,100	40	Yes
Bill	68,000	55	No
Jim	74,000	46	No
Mark	23,000	47	Yes
Anne	100,000	49	No

→ **Tree induction:** Induces a classification tree from examples

↑ ***Training*** data, ***labeled*** data

13

Classification Tree: Representation and Terminology

Employed ROOT

Yes → Class=NOT Default

No → **Balance** NODE

<50K → Class=NOT Default

>=50K → **Age**

<45 → Class=NOT Default

>=45 → Class= Default

LEAF

14

Classification Tree: Divide and Conquer

A series of nested tests (recursive partitioning)

- **Nodes**
  - Each "non-terminal" node represents a test on one or more attributes
  - Tests on nominal attribute: number of splits (branches) is number of possible values <u>or</u> one value vs. rest
  - Numeric attributes are discretized
- **Leaves:**
  - A class assignment (E.g, Default /Not default)
  - Also provide a distribution over all possible classes (e.g., default with probability 0.25, not default with prob. 0.75)

Employed

Yes → Class=NOT Default

No → **Balance**

<50K → Class=NOT Default

>=50K → **Age**

<45 → Class=NOT Default

>=45 → Class= Default

15

How a Classification Tree is Used for Classification?

Consider an unseen example

- To determine the class of a new example: E,g., **<u>Mark, age 40, works for CitiBank, balance 88K.</u>**
- The example is routed down the tree according to values of attributes tested successively.
- At each node a test is applied to one attribute.
- When a leaf is reached, the example is assigned to a class—or alternatively to a distribution over the possible classes (e.g., default with probability 0.25, not default with prob. 0.75).

Employed

No → Yes

Yes → **Balance**

<50K → No

>=50K → **Age**

<45 → No

>=45 → Yes

16

Definition Recap.

- Decision tree is a classifier in the form of a tree structure
  - Decision node: specifies a test on a single attribute
  - Leaf node: indicates the value of the target attribute
  - Arc/edge: split of one attribute
  - Path: a disjunction of test to make the final decision
- Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.

17

Decision Tree Classification

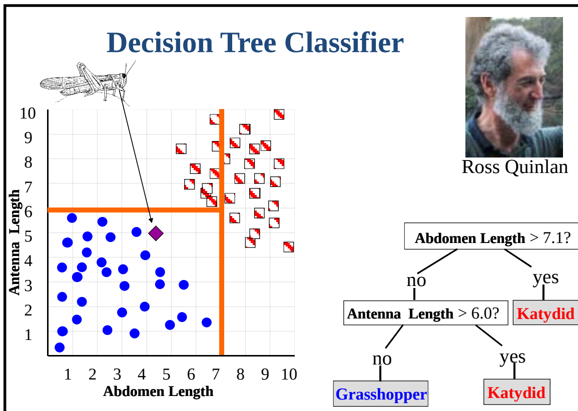
- Decision tree
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution
- Decision tree generation consists of two phases
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

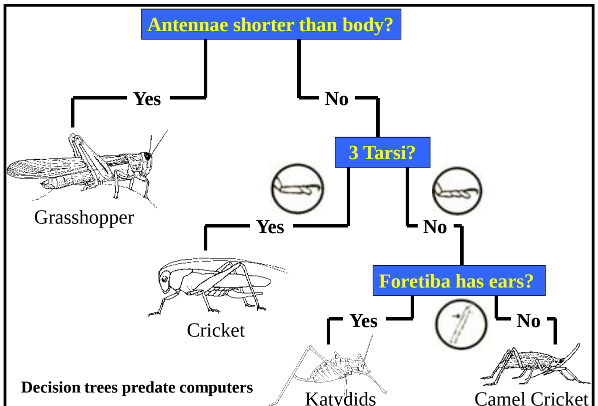
18

Classification Tree Induction

- Objective:
- Based on customer attributes, partition the customers into **subgroups that are less impure – with respect to the class** (i.e., such that in each group most instances belong to the same class)

Decision Tree Classifier





Decision trees predate computers

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

attribute

attribute value

decision

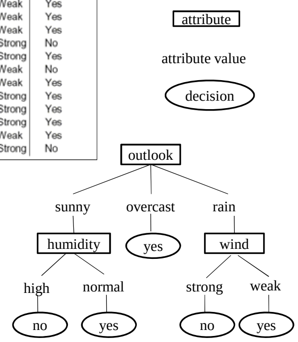
How do we construct the decision tree?

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they can be discretized in advance)
  - Examples are partitioned recursively based on selected attributes.
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

Top-Down Decision Tree Induction

- Main loop:
  1. A ← the “best” decision attribute for next node
  2. Assign A as decision attribute for node
  3. For each value of A, create new descendant of node
  4. Sort training examples to leaf nodes
  5. If training examples perfectly classified,
     Then STOP, Else iterate over new leaf nodes

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Picking a Good Split Feature

- Goal is to have the resulting tree be as small as possible, per Occam's razor.
- Finding a minimal decision tree (nodes, leaves, or depth) is an NP-hard optimization problem.
- Top-down divide-and-conquer method does a greedy search for a simple tree but does not guarantee to find the smallest.
  - General lesson in ML: "Greed is good."
- Want to pick a feature that creates subsets of examples that are relatively "pure" in a single class so they are "closer" to being leaf nodes.
- There are a variety of heuristics for picking a good test, a popular one is based on information gain that originated with the ID3 system of Quinlan (1979).

Principled Criterion

- Selection of an attribute to test at each node - choosing the most useful attribute for classifying examples.
- ***How***?
- Information gain
  - measures how well a given attribute separates the training examples according to their target classification
  - This measure is used to select among the candidate attributes at each step while growing the tree

Information Theory

- Think of playing "20 questions": I am thinking of an integer between 1 and 1,000 -- what is it? What is the first question you would ask?
- What question will you ask?
- Why?

- Entropy measures how much ***more*** information you need before you can identify the integer.
- Initially, there are 1000 possible values, which we assume are equally likely.
- What is the ***maximum*** number of question you need to ask?

Information Gain as A Splitting Criteria

- Select the attribute with the highest information gain (information gain is the expected reduction in entropy).
- Assume there are two classes, *P* and *N*
  - Let the set of examples ***S*** contain *p* elements of class *P* and *n* elements of class *N*
  - The amount of information, needed to decide if an arbitrary example in *S* belongs to *P* or *N* is defined as

$$E(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

0 log(0) is defined as 0

Entropy

- Entropy (disorder, impurity) of a set of examples, S, relative to a binary classification is:

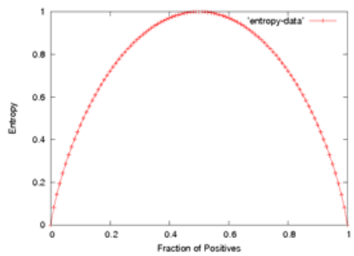
$$Entropy(S) = -p_1 \log_2(p_1) - p_0 \log_2(p_0)$$

where p_1 is the fraction of positive examples in S and p_0 is the fraction of negatives.
- If all examples are in one category, entropy is zero (we define 0⋅log(0)=0)
- If examples are equally mixed (p_1=p_0=0.5), entropy is a maximum of 1.
- Entropy can be viewed as the number of bits required on average to encode the class of an example in *S* where data compression (e.g. Huffman coding) is used to give shorter codes to more likely cases.
- For multi-class problems with c categories, entropy generalizes to:

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2(p_i)$$

Entropy Plot for Binary Classification

- The entropy is 0 if the outcome is ***certain***.
- The entropy is maximum if we have no knowledge of the system (or any outcome is equally possible).



Entropy of a 2-class problem with regard to the portion of one of the two groups

Information Gain

- Is the expected reduction in entropy caused by partitioning the examples according to this attribute.
- is the number of bits saved when encoding the target value of an arbitrary member of *S*, by knowing the value of attribute *A*.

Information Gain in Decision Tree Induction

- Assume that using attribute A, a current set will be partitioned into some number of child sets
- The encoding information that would be gained by branching on *A*

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

Note: entropy is at its minimum if the collection of objects is completely uniform

Continuous Attribute?

- Each non-leaf node is a test, its edge partitioning the attribute into subsets (easy for discrete attribute).
- For continuous attribute
  - Partition the continuous value of attribute A into a discrete set of intervals
  - Create a new boolean attribute A_c , looking for a threshold c,

$$A_c = \begin{cases} true & \text{if } A_c < c \\ false & \text{otherwise} \end{cases}$$

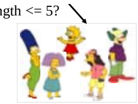
How to choose c ?

Person	Hair Length	Weight	Age	Class
Homer	0"	250	36	M
Marge	10"	150	34	F
Bart	2"	90	10	M
Lisa	6"	78	8	F
Maggie	4"	20	1	F
Abe	1"	170	70	M
Selma	8"	160	41	F
Otto	10"	180	38	M
Krusty	6"	200	45	M
Comic	8"	290	38	?

$$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

Entropy(4**F**,5**M**) = -(4/9)log$_2$(4/9) - (5/9)log$_2$(5/9)
= 0.9911

yes / Hair Length <= 5? / no



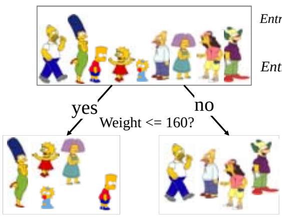
Let us try splitting on *Hair length (whether it is <=5)*

Entropy(1**F**,3**M**) = -(1/4)log$_2$(1/4) - (3/4)log$_2$(3/4)
= **0.8113**

Entropy(3**F**,2**M**) = -(3/5)log$_2$(3/5) - (2/5)log$_2$(2/5)
= **0.9710**

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

Gain(Hair Length <= 5) = 0.9911 – (4/9 * **0.8113** + 5/9 * **0.9710**) = **0.0911**



$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$

Entropy(4**F**,5**M**) = -(4/9)log$_2$(4/9) - (5/9)log$_2$(5/9)
= 0.9911

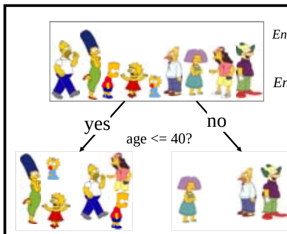
Let us try splitting on *Weight* *(whether it is <=160)*

Entropy(4**F**,1**M**) = -(4/5)log$_2$(4/5) - (1/5)log$_2$(1/5)
= **0.7219**

Entropy(0**F**,4**M**) = -(0/4)log$_2$(0/4) - (4/4)log$_2$(4/4)
= *0*

$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$

Gain(Weight <= 160) = 0.9911 – (5/9 * 0.7219 + 4/9 * 0) = **0.5900**



$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$

Entropy(4**F**,5**M**) = -(4/9)log$_2$(4/9) - (5/9)log$_2$(5/9)
= 0.9911

Let us try splitting on *Age* *(whether it is <=40)*

Entropy(3**F**,3**M**) = -(3/6)log$_2$(3/6) - (3/6)log$_2$(3/6)
= *1*

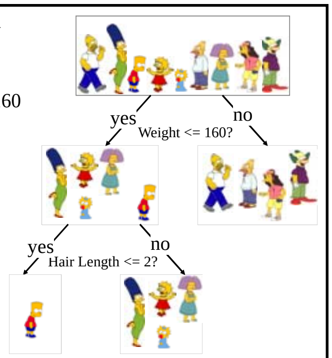
Entropy(1**F**,2**M**) = -(1/3)log$_2$(1/3) - (2/3)log$_2$(2/3)
= ***0.9183***

$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$

Gain(Age <= 40) = 0.9911 – (6/9 * 1 + 3/9 * 0.9183) = **0.0183**

Of the 3 features we had, *Weight* was best. But while people who weigh over 160 are perfectly classified (as males), the under 160 people are not perfectly classified… **RECURSION**!

This time we find that we can split on *Hair length*, and we are done!



We don't need to keep the data around, just the test conditions.

How would these people be classified?

Weight <= 160?
- yes → **Hair Length <= 2?**
  - yes → **Male**
  - no → **Female**
- no → **Male**

It is trivial to convert Decision Trees to rules…

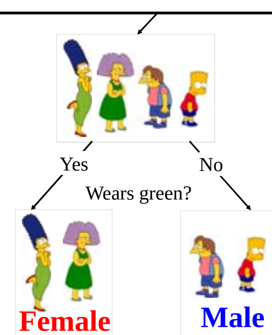
Weight <= 160?
- yes → **Hair Length <= 2?**
  - yes → **Male**
  - no → **Female**
- no → **Male**

Rules to Classify Males/Females

If *Weight* **greater than** 160, classify as **Male**
Elseif *Hair Length* **less than or equal** to 2, classify as **Male**
Else classify as **Female**

The worked examples we have seen were performed on small datasets. However with small datasets there is a great danger of overfitting the data…

When you have few data points, there are many possible splitting rules that perfectly classify the data, but will not generalize to future datasets.



For example, the rule "Wears green?" perfectly classifies the data, so does "Mothers name is Jacqueline?", so does "Has blue shoes"…