Decision Tree Learning

Features

A method for approximating discrete-valued functions which is

1. robust to noisy data
2. capable of learning disjunctive expressions
   ➔ Searches a completely expressive hypothesis space
3. easy to understand by humans (especially its learned results!)

Inductive Bias: “Prefer small trees over large trees”

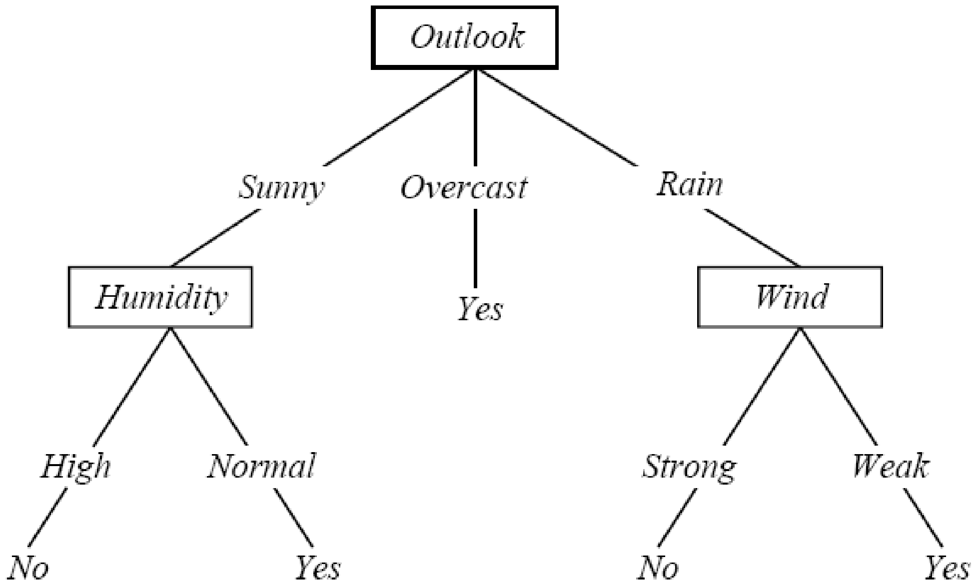
Outline

- Decision tree representation
- ID3 learning algorithm
- Entropy, Information gain
- Overfitting

Training Examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Decision Tree for *PlayTennis*



How to convert tree into disjunctions of conjunctions of constraints on attribute values of instances?

➔Convert into set of if-then rules

➔Each path from the root node to a leaf corresponds to a conjunctions of attribute tests

➔Tree is a disjunctions of conjunctions of constraints on attribute values

1. How Would you classify <outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong>?

Typical Datamining Task

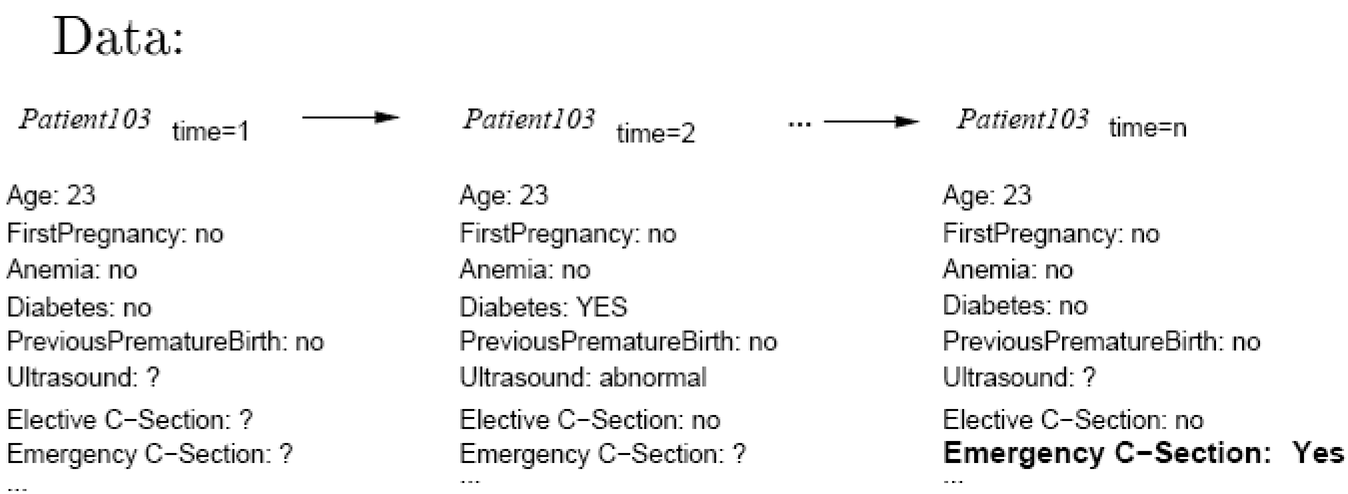
From lecture 1-1

Given:

- 9714 patient records each describing a pregnancy and birth
- Each patient record contains 215 features

Learn to predict:

- Classes of future patients at high risk for Emergency Cesarean Section



A Tree to Predict C-Section Risk

Learned from medical records of 1000 women, neg. examples are C-sections:

[833+,167-] .83+ .17-

- Fetal_Presentation = 1: [822+,116-] .88+ .12-
  - Previous_Csection = 0: [767+,81-] .90+ .10-
    - Primiparous = 0: [399+,13-] .97+ .03-
    - Primiparous = 1: [368+,68-] .84+ .16-
      - Fetal_Distress = 0: [334+,47-] .88+ .12-
        - Birth_Weight < 3349: [201+,10.6-] .95+ .05-
        - Birth_Weight >= 3349: [133+,36.4-] .78+ .22-
      - Fetal_Distress = 1: [34+,21-] .62+ .38-
  - Previous_Csection = 1: [55+,35-] .61+ .39-
- Fetal_Presentation = 2: [3+,29-] .11+ .89-
- Fetal_Presentation = 3: [8+,22-] .27+ .73-

- Decision tree representation:
  - Each internal node tests an attribute
  - Each branch corresponds to an attribute value
  - Each leaf node assigns a classification
- How would we represent:
  - AND, OR, XOR
  - $(\mathrm{A} \wedge \mathrm{B}) \vee (C \wedge \neg D \wedge E)$
  - *M* of *N*

$A \wedge B$

$A \vee B$

A *XOR* B

$(A \wedge B) \vee (C \wedge \neg D \wedge E)$

When to Consider Decision Trees

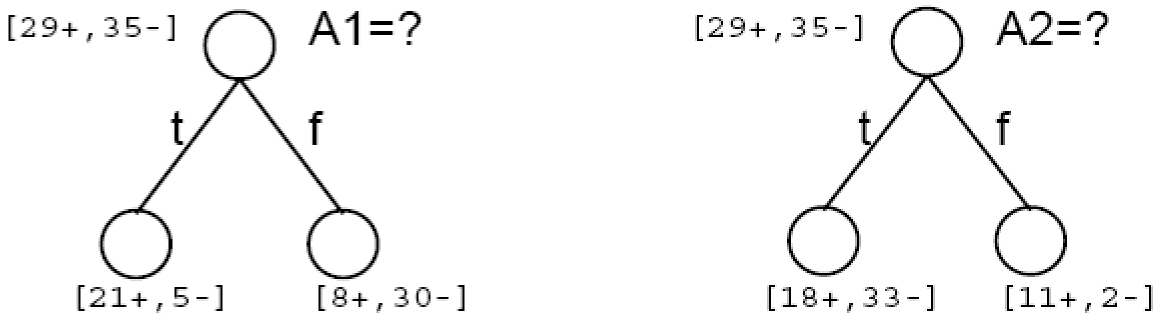
- Instances described by attribute values
- Target function is discrete valued
- Disjunctive hypotheses may be required
- Possibly noisy training data
- Training data may contain missing attribute values

- Examples:
  - Equipment or medical diagnosis
  - Credit risk analysis
  - Modeling calendar scheduling preferences

Top-down Induction of Decision Trees

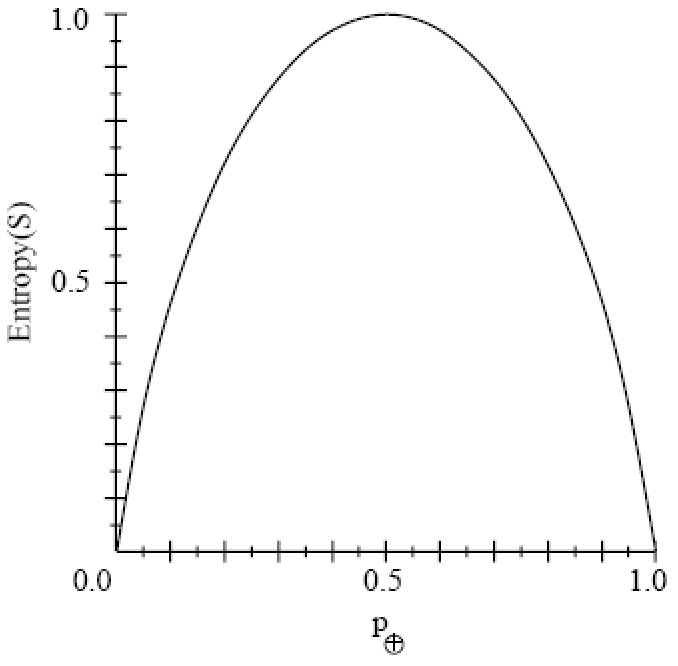
- Main loop:
  - $A \leftarrow$ the "best" decision attribute for next *node*
  - Assign A as decision attribute for *node*
  - For each value of A, create new descendant of *node*
  - Sort training examples to leaf nodes
  - If training examples perfectly classified, then STOP, else iterate over new leaf nodes

- Which attribute is best?



(Shannon) Entropy (1)

- *S* is a sample of training examples
- p_+ is the fraction of positive examples in *S*
- p_- is the fraction of negative examples in *S*
- Entropy measures the impurity of *S*



$$Entropy(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

(Shannon) Entropy (2)

Entropy(S) = expected number of bits needed to encode class + or - of randomly drawn member of *S* (under the optimal, shortest-length code)

Why?

Information theory: Optimal length code assigns $-\log_2 p$ bits to message having probability *p*.

So, expected number of bits to encode + or - of random member of *S*:

$$p_+(-\log_2 p_+) + p_-(-\log_2 p_-)$$

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Entropy for c-wise Classification

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

- p_i = proportion of S belonging to class i
- Maximum value = $\log_2 c$

- Measures the expected encoding length measured in *bits*

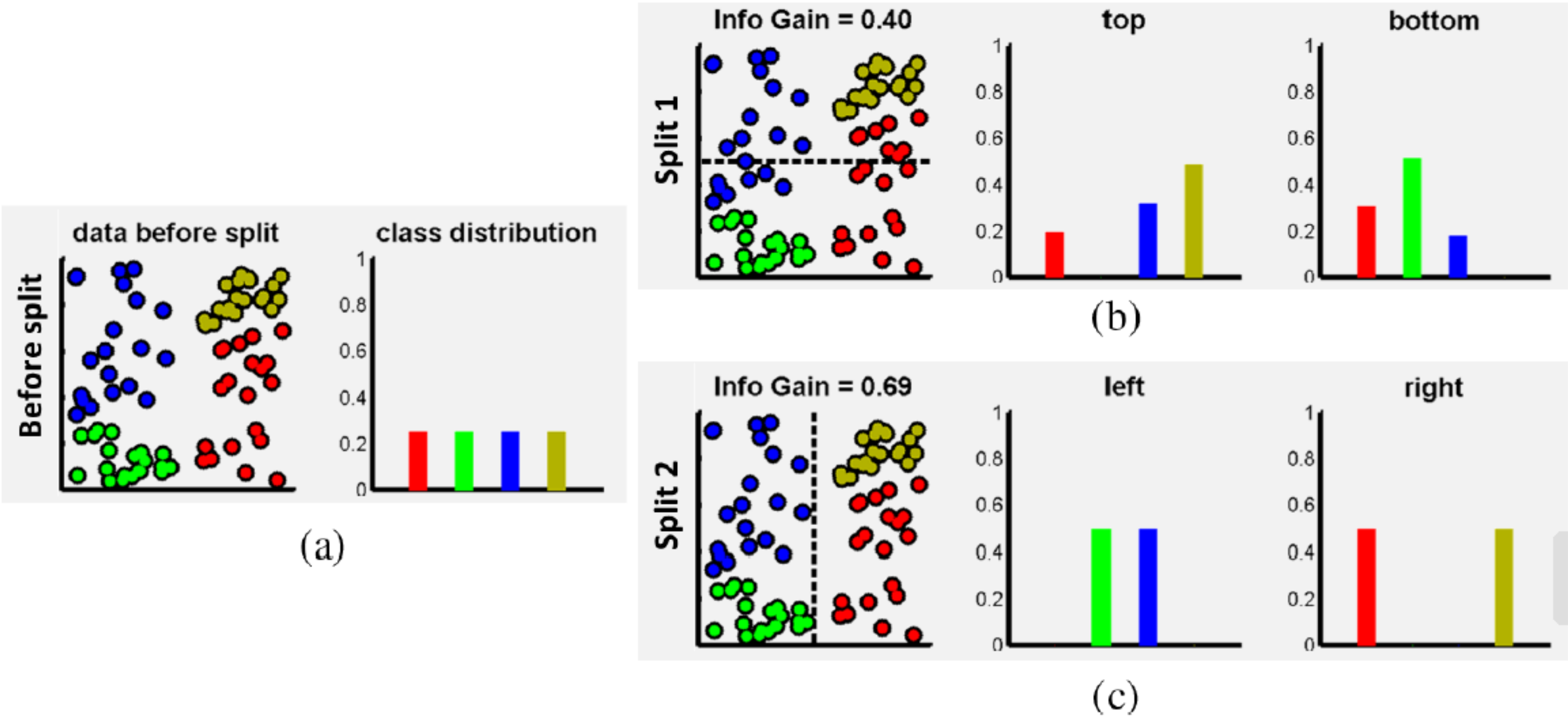


Fig. 2.5 Information gain for discrete, non-parametric distributions. (a) Dataset $\mathcal{S}$ before a split. (b) After a horizontal split. (c) After a vertical split. In this example the vertical split produces purer class distributions in the child nodes. Classes are colour coded.

Quelle: Antonio Criminisi, Jamie Shotton, Ender Konukoglu . *Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning*. In Foundations and Trends® in Computer Graphics and Vision, Vol. 7: No 2-3, pp 81-227, 2011.

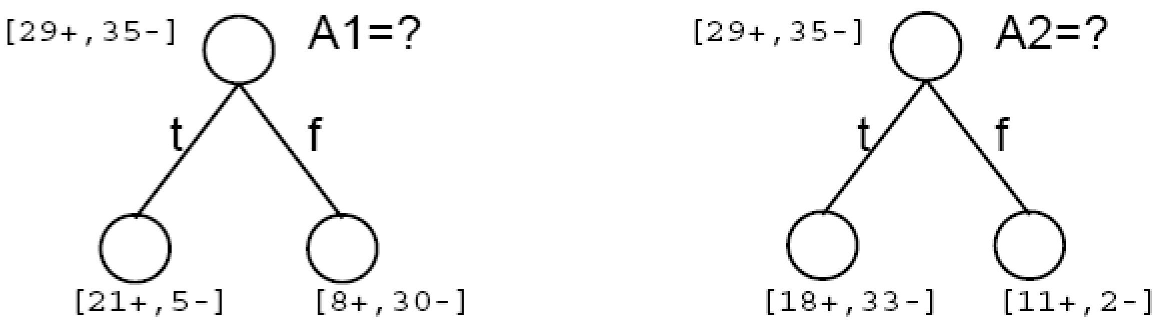
Information Gain

- *Gain(S,A)* = expected reduction in entropy due to sorting on *A*

Original entropy

Entropy after partitioning

$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

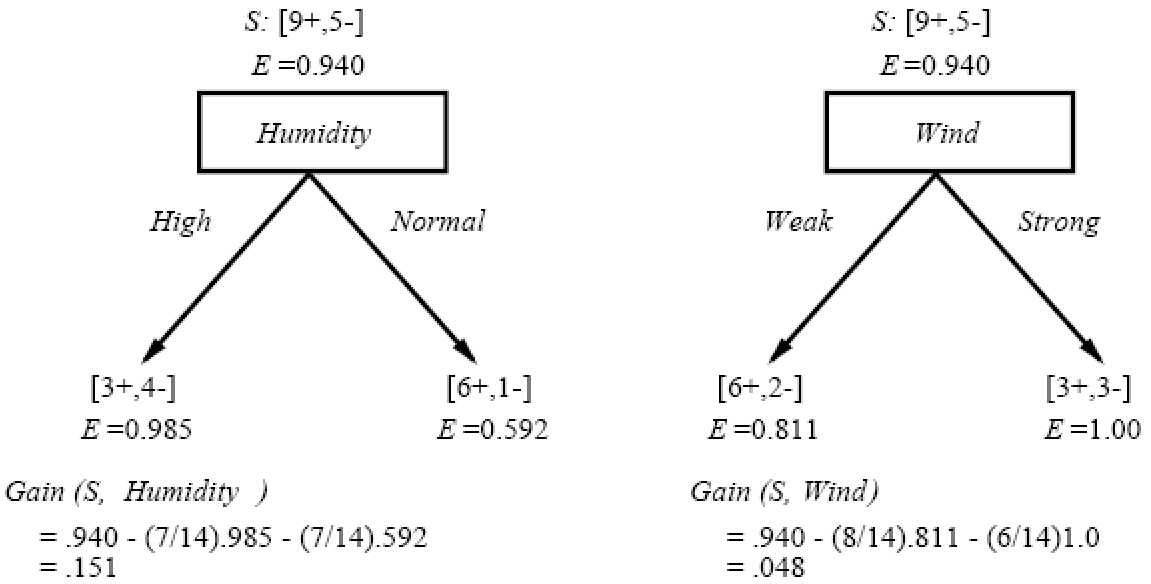


➔ Let's compute all Entropy and Gain values

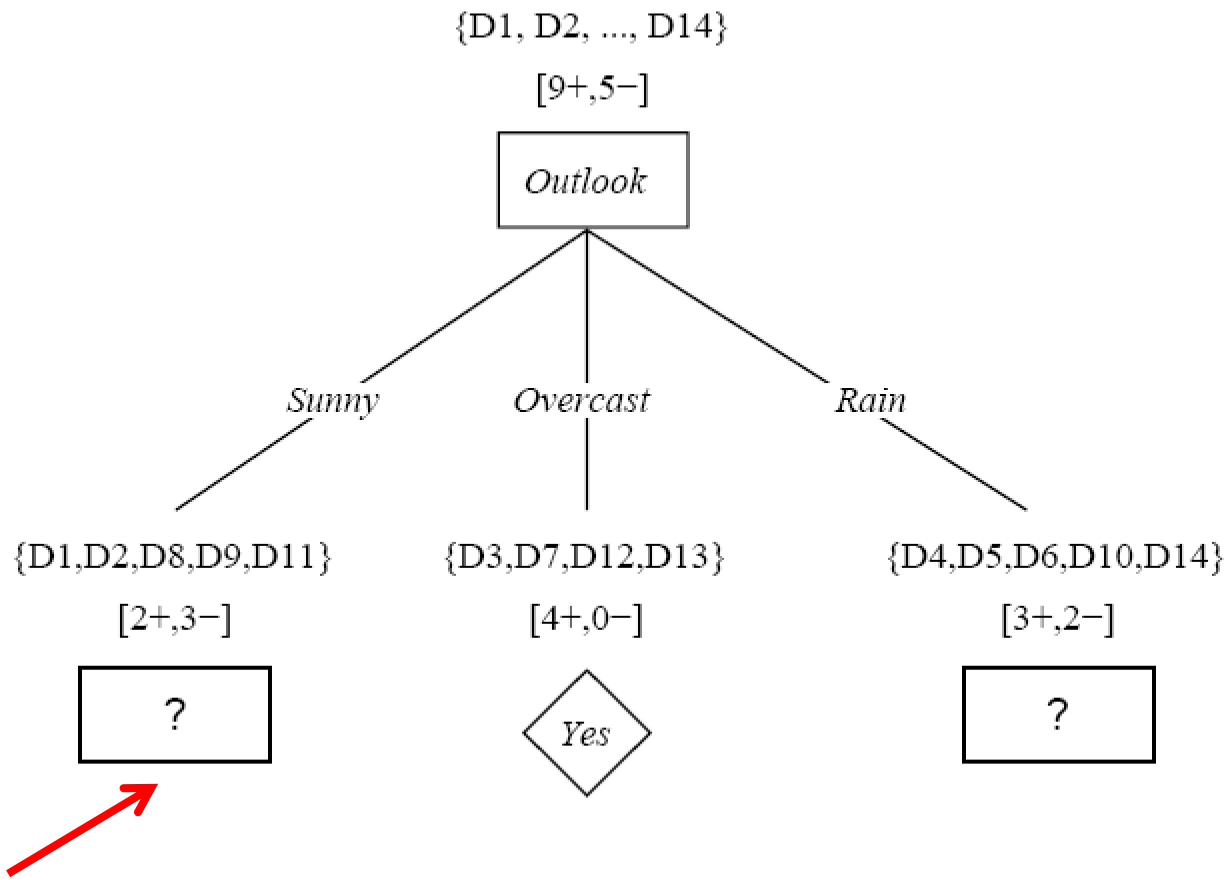
Training Examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

- Which attribute is the best classifier?



Selecting the Next Attribute



- Which attribute should be tested here?

Information Gain Calculation

- $S_{sunny} = \{D1,D2,D8,D9,D11\}$

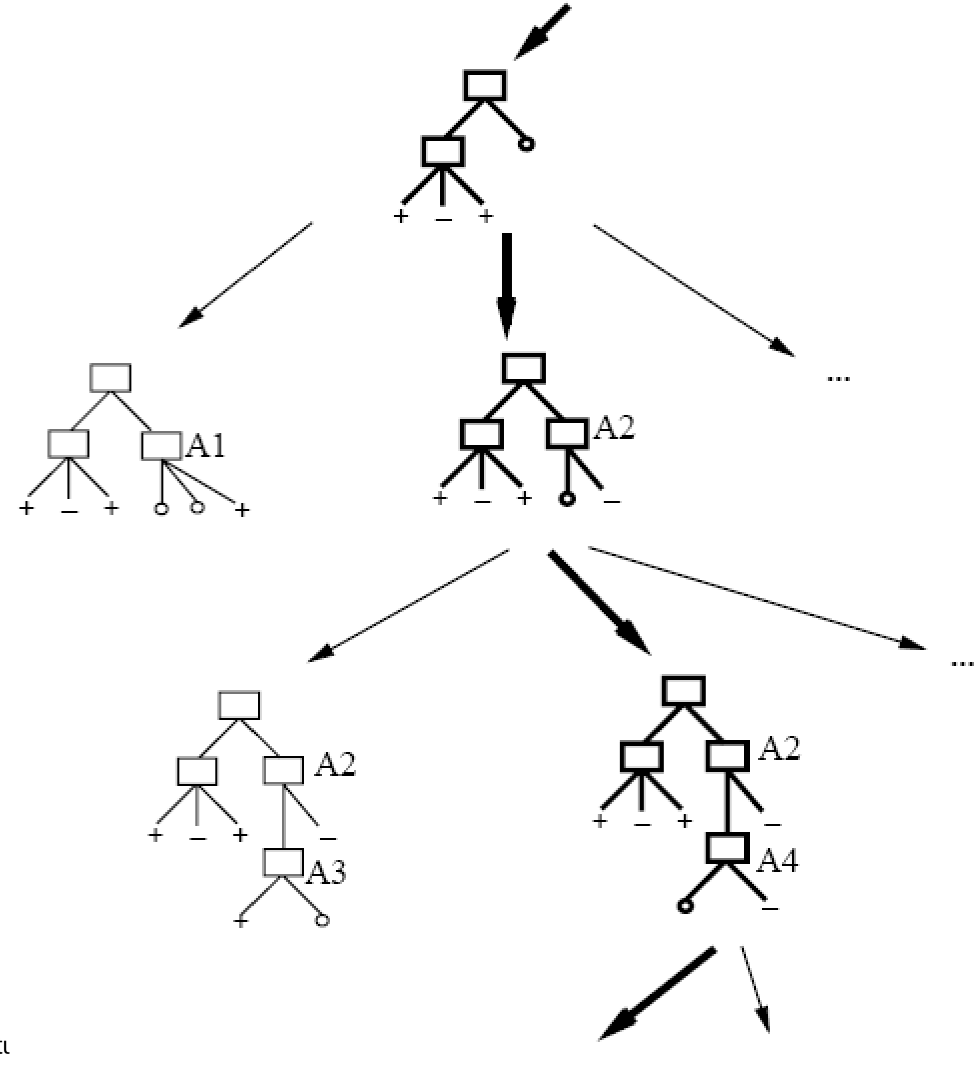
$$Gain\ (S_{sunny}\ , Humidity) = .970 - (3/5)\ 0.0 - (2/5)\ 0.0 = .970$$

$$Gain\ (S_{sunny}\ , Temperature) = .970 - (2/5)\ 0.0 - (2/5)\ 1.0 - (1/5)\ 0.0 = .570$$

$$Gain\ (S_{sunny}\ , Wind) = .970 - (2/5)\ 1.0 - (3/5)\ .918 = .019$$

- Humidity has highest information gain

Hypothesis Space Search by ID3 (1)



Hypothesis Space Search by ID3 (2)

- Hypothesis space is complete!
  - Target function surely is in there...
- Outputs a single hypothesis (which one?)
  - Can't play 20 questions...
- No back tracking
  - Local minima...
- Statistically-based search choices
  - Robust to noisy data...
- Inductive bias:
  - Approximately: „prefer shortest tree“

Inductive Bias in ID3

Note *H* is the power set of instances *X*

- Unbiased?

Not really...

- Preference for short trees, and for those with high information gain attributes near the root
- Bias is a *preference* for some hypotheses, rather than a *restriction* of hypothesis space *H*
- Occam's razor: prefer the shortest hypothesis that fits the data

Occam‘s Razor: Why Prefer Short Hypotheses?

- In favor:
  - There are fewer short hypotheses than long hypotheses
  - A short hypothesis that fits the data is unlikely to be coincidence
  - A long hypothesis that fits the data might be coincidence
- Opposed:
  - There are many ways to define small sets of hypotheses

    e.g., all trees with a prime number of nodes that use attributes beginning with “Z”
  - What is short depends on language

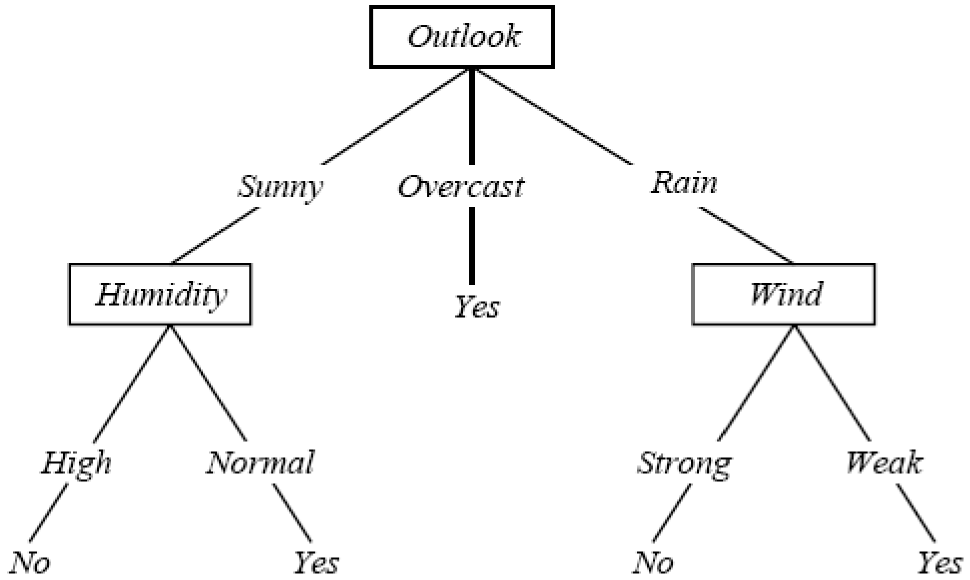
Universität Augsburg University

Consider adding noisy training example #15:

(<u>Outlook, Temperature, Humidity, Wind)</u>

Sunny, Hot, Normal, Strong, PlayTennis=No

What is the effect on earlier tree?



Overfitting

- Consider error of hypothesis *h* over
  - Training data: $error_{train}(h)$
  - Entire distribution of data: $error_{D}(h)$

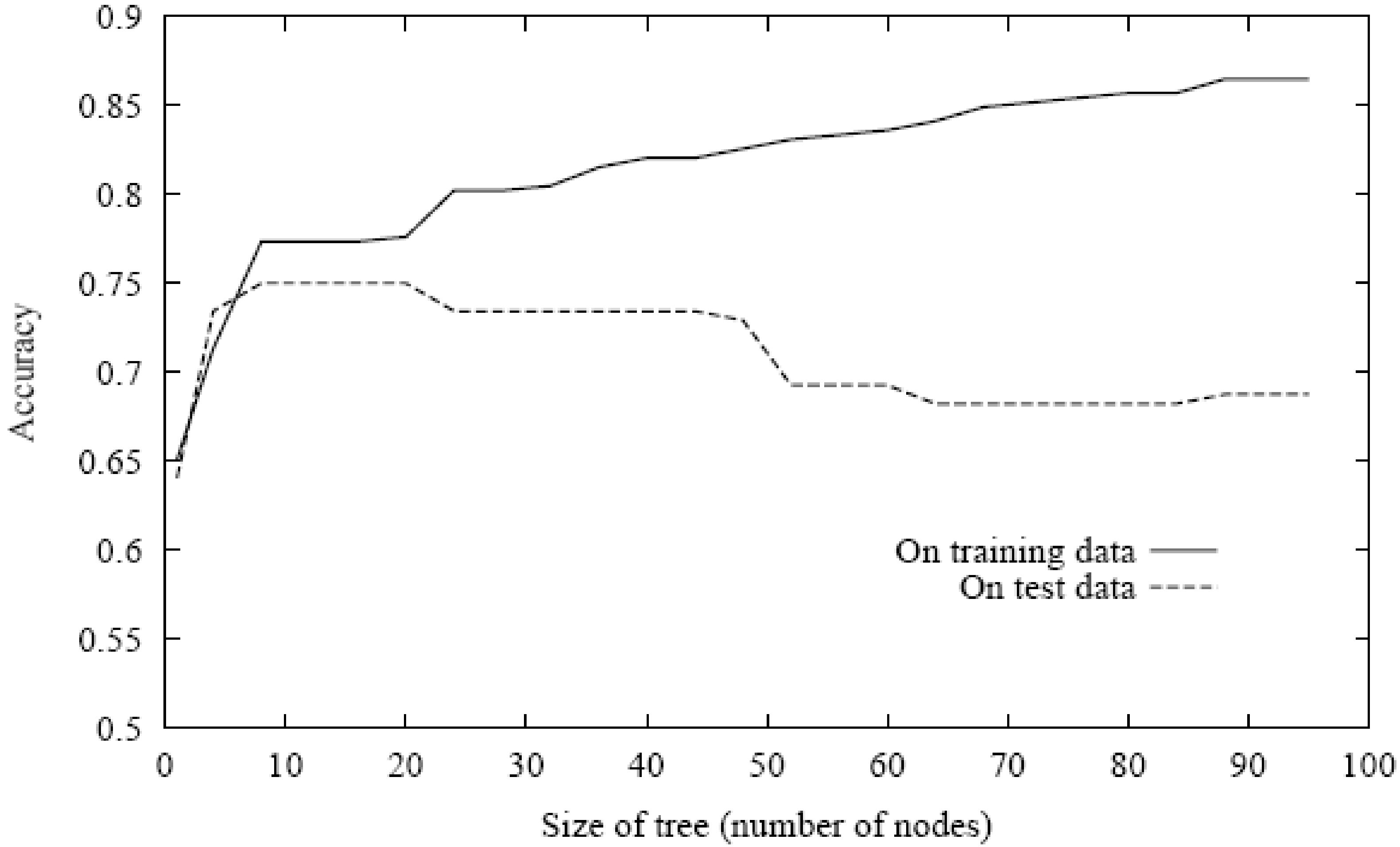
- Hypothesis $h \in H$ overfits training data if there is an alternative hypothesis $h' \in H$ such that

$$error_{train}(h) < error_{train}(h')$$

and

$$error_{D}(h) > error_{D}(h').$$

Overfitting in Decision Tree Learning



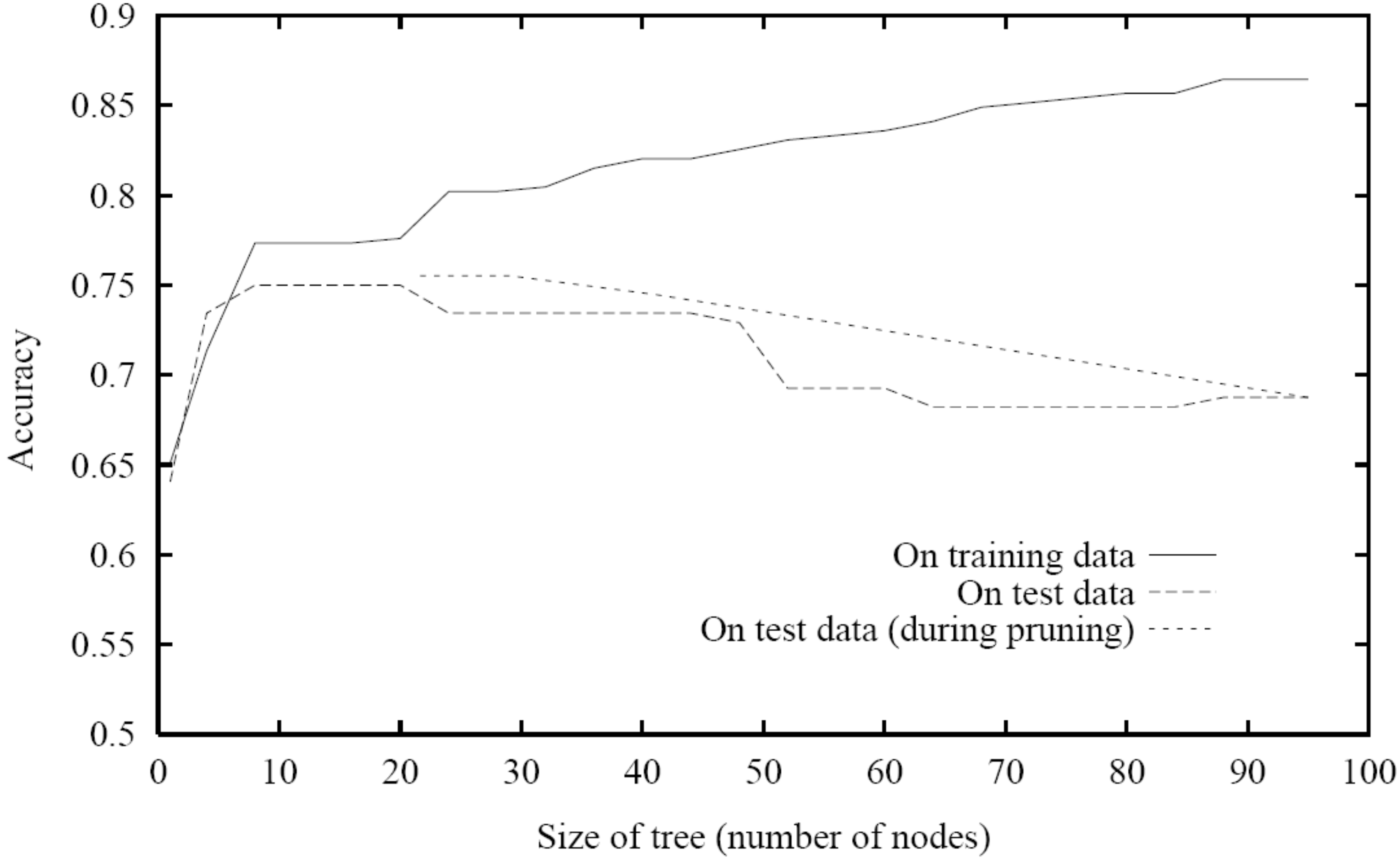
Avoiding Overfitting

- How can we avoid overfitting?
  - stop growing when data split is not statistically significant
  - grow full tree, then post-prune
- How to select „best“ tree:
  - Measure performance over training data
  - Measure performance over separate validation data set
  - Minimum Description Length: minimize *size(tree) + size(misclassifications(tree))*

Reduced Error-Pruning

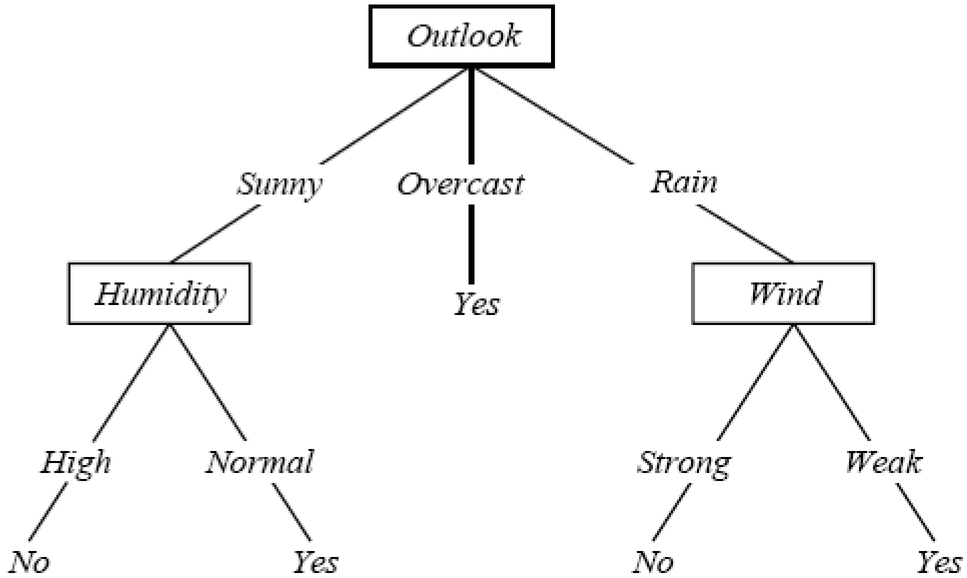
- Split data into *training* and *validation* set
- Do until further pruning is harmful:
  1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
  2. Greedily remove the one that most improves *validation* set accuracy

Effect of Reduced Error-Pruning



lecture slides for textbook Machine Learning, T. Mitchell, McGraw Hill, 1997

Converting a Tree to Rules



- IF (*Outlook=Sunny*) and (*Humidity=High*) THEN *PlayTennis=No*
- IF (*Outlook=Sunny*) and (*Humidity=Normal*) THEN *PlayTennis=Yes*
- …

Issues with Decision Trees: 2. Continuous Valued Attributes

- Create a discrete attribute to test continuous
  - Temperature = 82.5
  - (Temperature > 54) = true ELSE false

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

Issues with Decision Trees: 3. Attributes With Many Values

Universität Augsburg University

- Problem: Gain measure has a natural bias towards attributes with many values
  - if attribute has many values, *Gain* will select it
  - Imagine using Date = Jun 3 1996 as attribute
- Use alternative method for selecting attributes
- One approach: use GainRatio instead:

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where S_i is subset of *S* for which *A* has value v_i

$\log_2 c$

Discourages the selection of attributes with many uniformly distributed values

E.g. two-class problem in 2D ($\boldsymbol{x} \in \mathbb{R}^{\mathbf{2}}$):

$$f(\boldsymbol{x}) = I_{x_2 > x_1}$$

This linear, but not axis aligned decision function needs to be approximated by **many** nodes. It cannot be efficiently learned. It also needs many training examples in order to be approximately learned.

Unknown Attribute Values in Training Data

What if some examples have missing values of *A*?

Use training example anyway, sort through tree

1. If node *n* tests *A*, assign most common value of *A* among other examples sorted to node *n*
2. Assign most common value of *A* among other examples with same target value
3. Assign probability p_i to each possible value v_i of *A*
   – assign fraction p_i of example to each descendant in tree

Classify new examples in same fashion.

Density Estimation with Gaussian (1)

Model n -dimensional data $\{(\boldsymbol{x}, y)\}, \boldsymbol{x} \in \mathbb{R}^n$ by n-variate Gaussian:

$$G =..$$

In the continuous case we define the differential entropy

$$H(S) = -\int_y p(y)\, log\big(p(y)\big)\, dy$$

which becomes for the multi-variate Gaussian

$$H(S) = \frac{1}{2}\log\big((2\pi e)^n\big) \cdot \det(\Sigma)$$

Density Estimation with Gaussian (2)

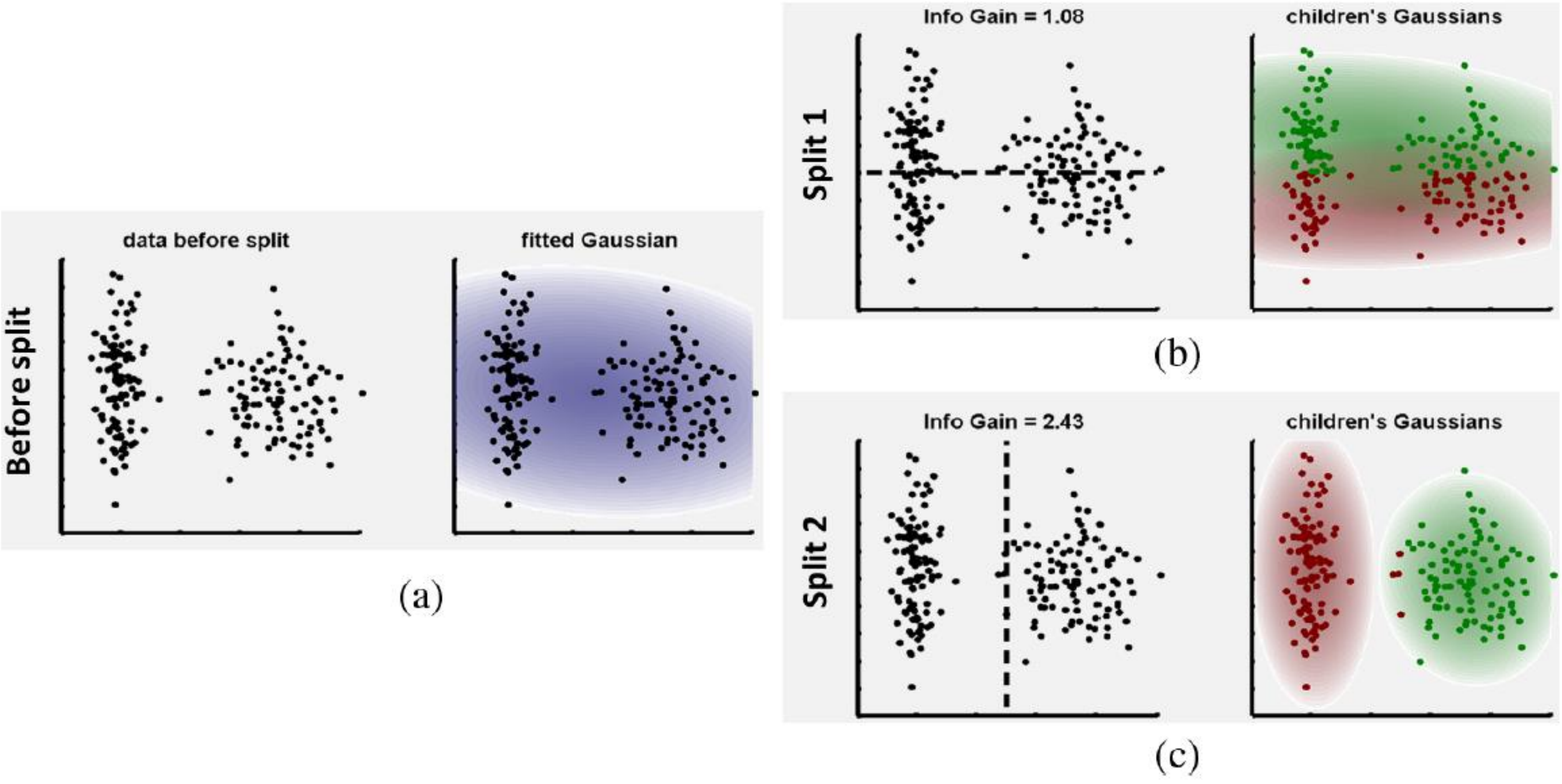


Fig. 2.6 Information gain for continuous, parametric densities. (a) Dataset $\mathcal{S}$ before a split. (b) After a horizontal split. (c) After a vertical split. A vertical split produces better separation and a correspondingly higher information gain.

Quelle: Antonio Criminisi, Jamie Shotton, Ender Konukoglu . *Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning*. In Foundations and Trends® in Computer Graphics and Vision, Vol. 7: No 2-3, pp 81-227, 2011.