Cornell University

Decision Tree Learning

CS4780/5780 – Machine Learning
Fall 2011

Thorsten Joachims
Cornell University

Reading: Mitchell Sections 2.1, 2.2, 2.5-2.5.2, 2.7, Chapter 3

Outline

- Hypothesis space
- Version space
- Inductive learning hypothesis
- List-then-eliminate algorithm
- Decision tree representation
- Classifying with a decision tree
- ID3 decision tree learning algorithm
- Entropy, Information gain
- Overfitting

Cornell University

Hypothesis Space

correct (3)	color (2)	original (2)	presentation (3)	binder (2)	A+Homework
complete	yes	yes	clear	no	yes / +1
complete	no	yes	clear	no	yes / +1
partial	yes	no	unclear	no	no / -1
complete	yes	yes	clear	yes	yes / +1

Instance Space X: Set of all possible objects described by attributes.

Target Function f: Maps each instance $x \in X$ to target label $y \in Y$ (hidden).

Hypothesis h: Function that approximates f.

Hypothesis Space H: Set of functions we allow for approximating f.

Training Data S: Set of instances labeled with target function f.

Consistency

Definition: *A hypothesis h is* **consistent** *with a set of training examples S of target concept f if and only if $h(x) = y$ for each training example $(x, y) \in S$.*

$$Consistent(h, S) \equiv [\forall (x, y) \in S : h(x) = y]$$

correct (3)	color (2)	original (2)	presentation (3)	binder (2)	A+Homework
complete	yes	yes	clear	no	yes
complete	no	yes	clear	no	yes
partial	yes	no	unclear	no	no
complete	yes	yes	clear	yes	yes

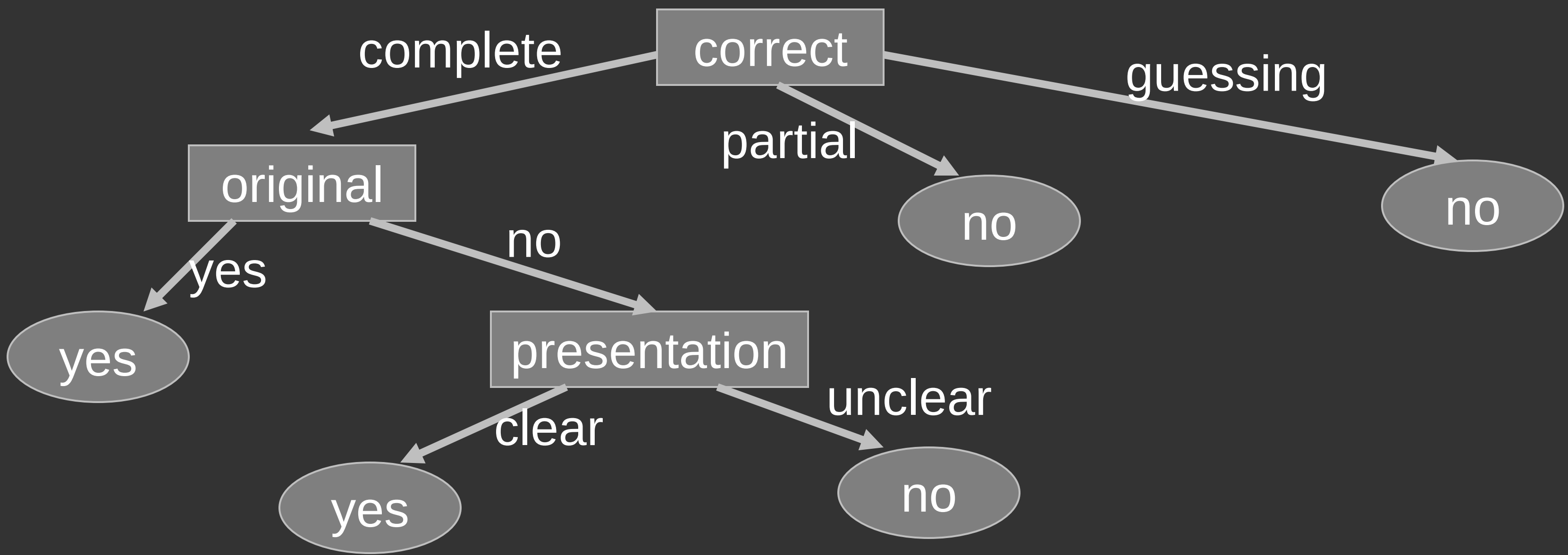
Version Space

Definition: *The* **version space**, $VS_{H,S}$, *with respect to hypothesis space* H *and training examples* S, *is the subset of hypotheses from* H *consistent with all training examples in* S.

$$VS_{H,S} \equiv \{h \in H | Consistent(h, S)\}$$

Cornell University

Decision Tree Example: A^{+}Homework



correct (3)	color (2)	original (2)	presentation (3)	binder (2)	A+Homework
complete	yes	yes	clear	no	yes
complete	no	yes	clear	no	yes
partial	yes	no	unclear	no	no
complete	yes	yes	clear	yes	yes

Cornell University

Top-Down Induction of DT (simplified)

Training Data: $S = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$

TDIDT(S,y_{def})

- IF(all examples in S have same class y)
  - Return leaf with class y (or class y_{def}, if S is empty)
- ELSE
  - Pick A as the "best" decision attribute for next node
  - FOR each value v_i of A create a new descendent of node
    - $S_i = \{(\vec{x}, y) \in S : \text{attribute A of } \vec{x} \text{ has value } v_i)\}$
    - Subtree t_i for v_i is TDIDT(S_i,y_{def})
  - RETURN tree with A as root and t_i as subtrees

Example: TDIDT

TDIDT(S,y_{def})

- IF(all ex in S have same y)
  - Return leaf with class y (or class y_{def}, if S is empty)
- ELSE
  - Pick A as the "best" decision attribute for next node
  - FOR each value v_i of A create a new descendent of node
    - $S_i = \{(\vec{x}, y) \in S : \text{attrib A of } \vec{x} \text{ is } v_i)\}$
    - Subtree t_i for v_i is TDIDT(S_i,y_{def})
  - RETURN tree with A as root and t_i as subtrees

Example Data S:

C O P	A^+
$\vec{x}_1 = (\text{c}, \text{y}, \text{c})$	$y_1 = +1$
$\vec{x}_2 = (\text{c}, \text{n}, \text{u})$	$y_2 = -1$
$\vec{x}_3 = (\text{c}, \text{y}, \text{u})$	$y_3 = +1$
$\vec{x}_4 = (\text{c}, \text{n}, \text{c})$	$y_4 = +1$
$\vec{x}_5 = (\text{p}, \text{y}, \text{c})$	$y_5 = -1$
$\vec{x}_6 = (\text{g}, \text{y}, \text{c})$	$y_6 = -1$
$\vec{x}_7 = (\text{c}, \text{y}, \text{c})$	$y_7 = +1$
$\vec{x}_8 = (\text{c}, \text{y}, \text{u})$	$y_8 = +1$
$\vec{x}_9 = (\text{p}, \text{y}, \text{c})$	$y_9 = -1$
$\vec{x}_{10} = (\text{c}, \text{y}, \text{c})$	$y_{10} = +1$

Which Attribute is "Best"?

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."

Cornell University

```
• 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
```

Total size of tree:

- 299 nodes

Note: word stems expanded for improved readability.

Learning as Prediction

Definition: *A particular* **instance of a learning problem** *is described by a probability distribution* $P(X,Y)$.

Definition: *A* **sample** $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n))$ *is independently identically distributed (i.i.d.) according to* $P(X,Y)$.

Sample Error and Generalization Error

Definition: *The* **error on sample** S $Err_S(h)$ *of a hypothesis* h *is* $Err_S(h) = \frac{1}{n}\sum_{i=1}^{n} \Delta(h(\vec{x}_i), y_i)$.

Definition: $\Delta(a, b)$ *is the 0/1-loss function*

$$\Delta(a,b) = \begin{cases} 0 & if(a == b) \\ 1 & else \end{cases}$$

Definition: *The* **prediction/generalization/true error** $Err_P(h)$ *of a hypothesis* h *for a learning task* $P(X,Y)$ *is*

$$Err_P(h) = \sum_{\vec{x} \in X, y \in Y} \Delta(h(\vec{x}), y) P(X = \vec{x}, Y = y).$$

Cornell University

Learning as Prediction Overview

Real-world Process
$P(X,Y)$

drawn i.i.d.

drawn i.i.d.

Train Sample S_{train}
$(x_1,y_1), \ldots, (x_n,y_n)$

S_{train}

Learner

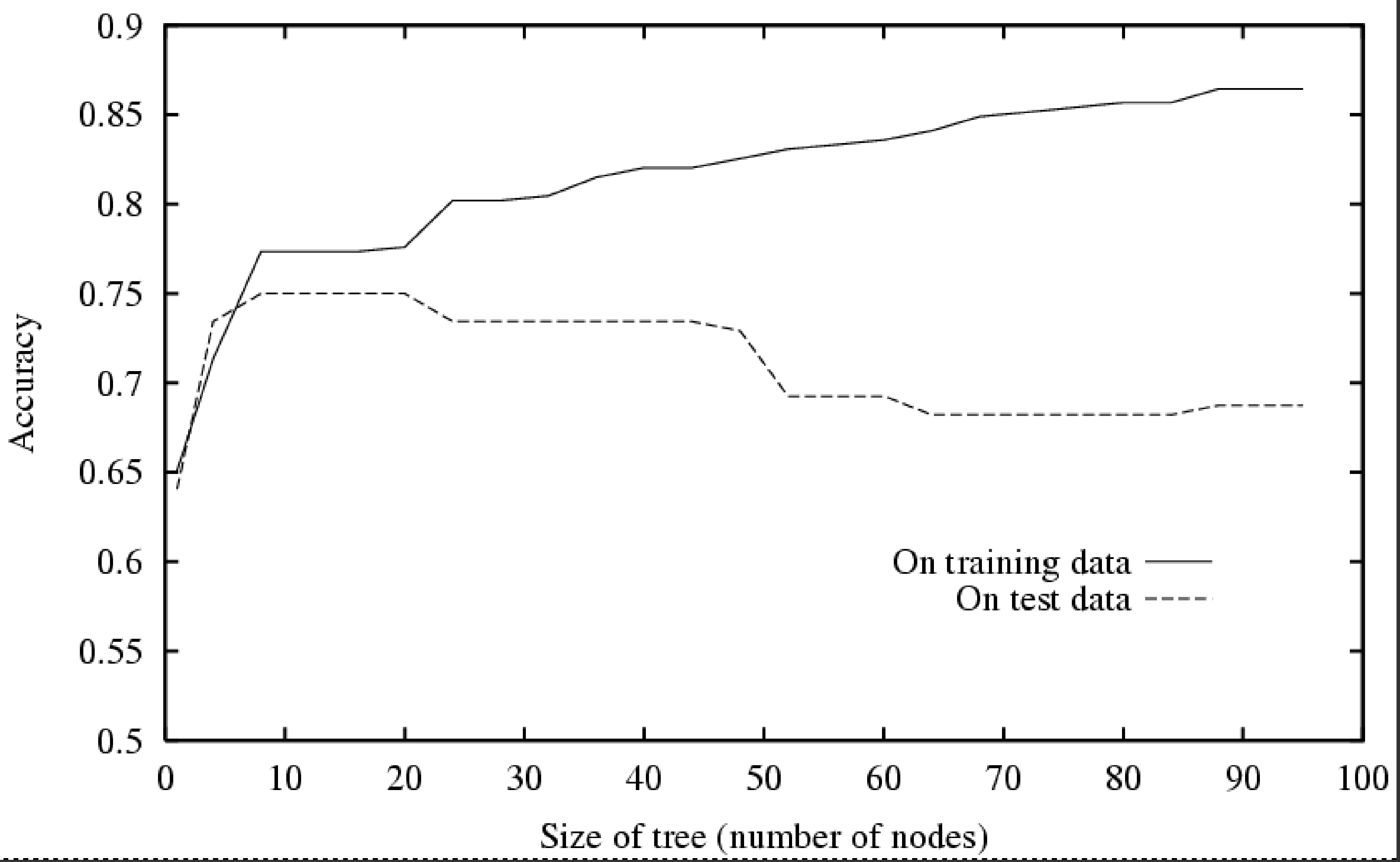
h

Test Sample S_{test}
$(x_{n+1},y_{n+1}), \ldots$

- Goal: Find h with small prediction error $Err_P(h)$ over $P(X,Y)$.
- Strategy: Find (any?) h with small error $Err_{S_{train}}(h)$ on training sample S_{train}.

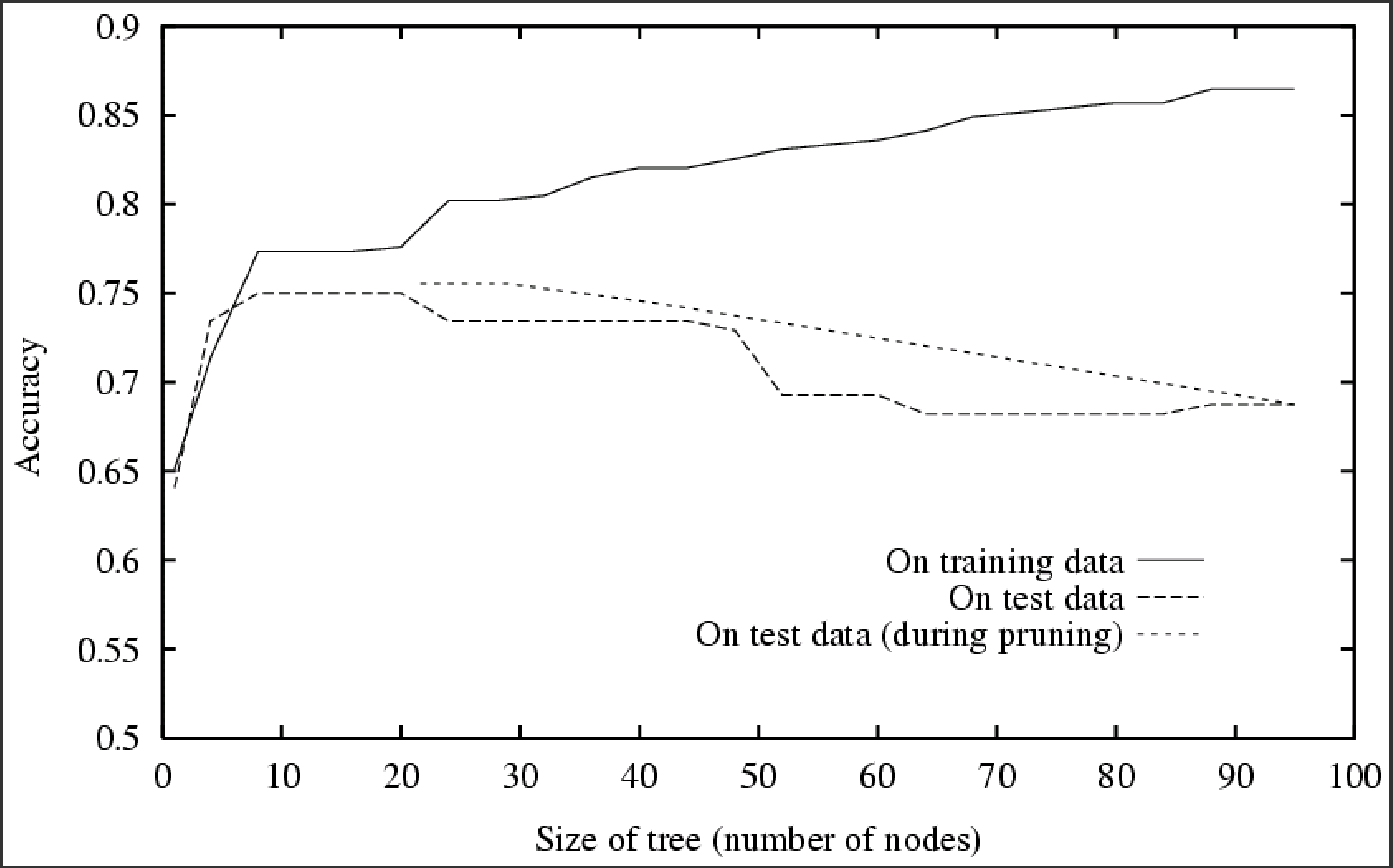
- Training Error: Error $Err_{S_{train}}(h)$ on training sample.
- Test Error: Error $Err_{S_{test}}(h)$ on test sample is an estimate of $Err_P(h)$.

Overfitting



- Note: Accuracy = 1.0-Error

Reduced-Error Pruning



Text Classification Example Results

- Unpruned Tree:
  - Size: 437 nodes   Training Error: 0.0%   Test Error: 11.0%
- Early Stopping Tree:
  - Size: 299 nodes   Training Error: 2.6%   Test Error: 9.8%
- Post-Pruned Tree:
  - Size: 167 nodes   Training Error: 4.0%   Test Error: 10.8%
- Rule Post-Pruning:
  - Size: 164 tests   Training Error: 3.1%   Test Error: 10.3%
  - Examples of rules
    - IF vs = 1 THEN - [99.4%]
    - IF vs = 0 & export = 0 & takeover = 1 THEN + [93.6%]