# Lecture 5 DT cont.

Oct 8 2008

# Review of last lecture

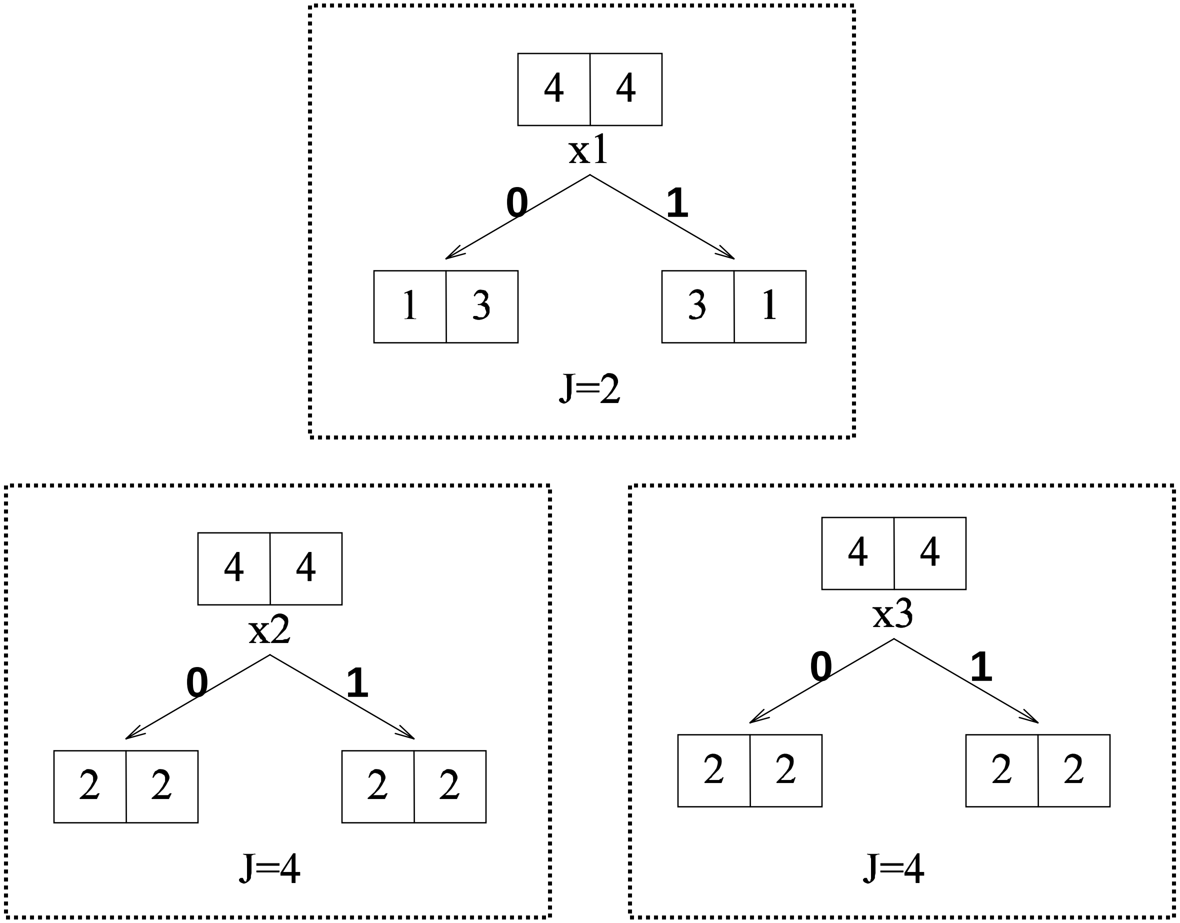
- What is decision tree
- What decision boundaries do decision trees produce
  - Syntactically different trees can represent the same decision boundaries
  - In such cases, we prefer smaller trees
  - flexible ***hypothesis space***
- How to learn a decision tree?
  - A greedy approach
  - At each step, choose the test that reduce the most uncertainty about class labels

# Choosing the test based on training error

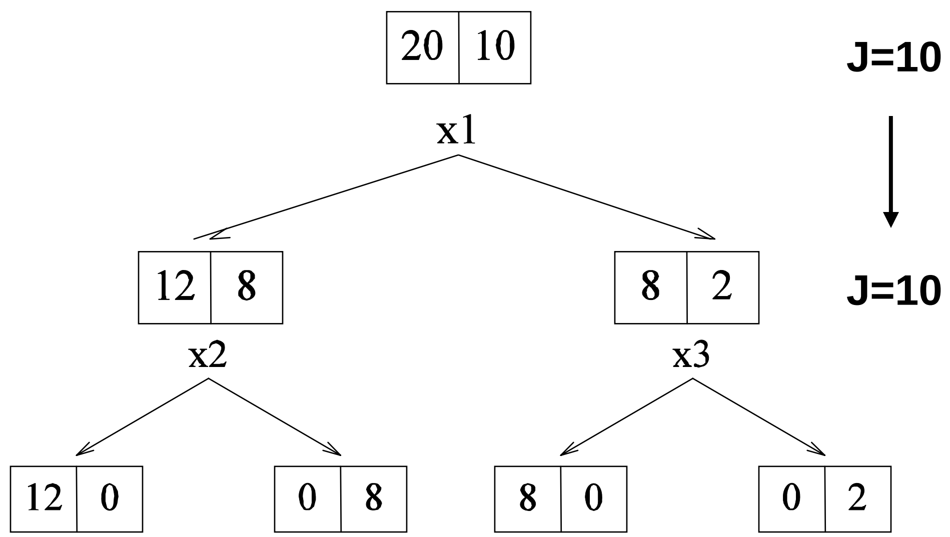
- Perform 1-step look-ahead search and choose the attribute that gives the lowest error rate on the training data

| $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

**Training examples**



Unfortunately, this measure does not always work well, because it does not detect cases where we are making "progress" toward a good tree



# A Better Heuristic from Information Theory

- Let *X* have the following probability distribution

| *P(X = 0)=$p_0$* | *P(X = 1)=$p_1$* |
|---|---|
| 0.2 | 0.8 |

- The <u>entropy</u> of ***X***, denoted *H(**X**)*, is defined as

$$H(X) = -P_0 \log_2 P_0 - P_1 \log_2 P_1$$

$$H(X) = -P_0 \log_2 P_0 - \ldots - P_k \log_2 P_k$$ if there are k possible values

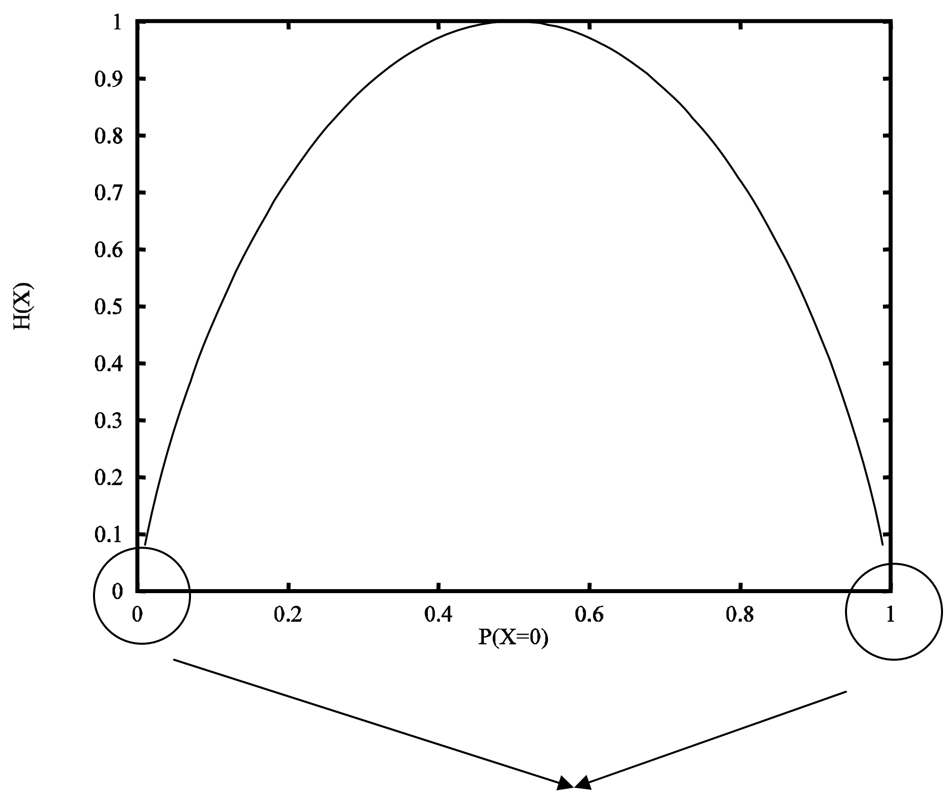
- **− log*P(X=$x$)*** measures the <u>surprise</u> of value ***$x$*:**

  If ***P**(X=$x$)* is small, $x$ is a surprising value to take,
  **−log*P***(*$x$*) is large

- **Entropy** can be considered as the average <u>surprise</u> of a random variable, which is also referred to as the uncertainty of a random variable

# Entropy

- Entropy is a concave function downward




Minimum uncertainty occurs when $p_0$=0 or 1

# Mutual Information

- If we use entropy to measure uncertainty, we end up measuring the <u>mutual information</u> between a candidate test variable *X* and class label *Y:*

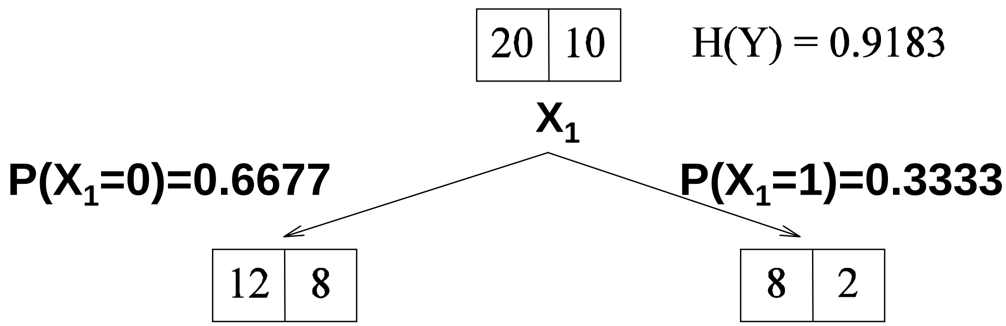
$$I(X,Y) = H(Y) - H(Y \mid X)$$

Uncertainty of Y

Remaining uncertainty of Y after knowing the value of X

- H(Y|X) is called the conditional entropy of Y given X
  - Measures the uncertainty of Y after knowing the value of X

$$H(Y \mid X) = \sum_x P(X = x) H(Y \mid X = x)$$
$$= -\sum_x \underbrace{P(X = x)}_{\text{The probability of X=x}} \sum_y \underbrace{P(Y = y \mid X = x) \log P(Y = y \mid X = x)}_{\text{The uncertainty of Y when X=x}}$$




$H(Y|X_1=0)$
$=-0.6*\log 0.6-0.4*\log 0.4$
$=0.9710$

$H(Y|X_1=1)$
$=-0.8*\log 0.8-0.2*\log 0.2$
$=0.7219$

$H(Y|X_1)= 0.6667*0.9710 + 0.3333*0.7219 = 0.8873$

**I(X1,Y)=0.0304**

# Information Gain

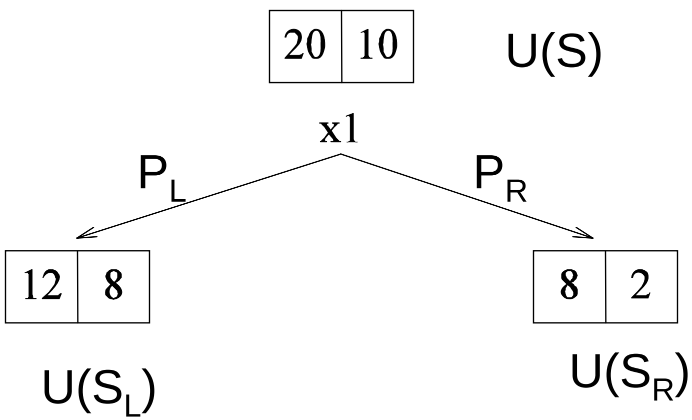
- This is called the **information gain** criterion: choose X that maximizes mutual information between X and y

$$\arg\max_j I(X_j;Y) = \arg\max_j H(Y) - H(Y \mid X_j)$$
$$= \arg\min_j H(Y \mid X_j)$$

- Information gain is just one of the methods for selecting tests in decision tree learning

- There are other methods as well, but they use the same general approach based on different uncertainty measures

# Choosing the Best Feature: A General View




Benefit of split =

$$U(S) - [\underbrace{P_L * U(S_L) + P_R * U(S_R)}_{\text{Expected Remaining Uncertainty (Impurity)}}]$$

| Measures of Uncertainty | |
|---|---|
| Error | $\min\{p, 1-p\}$ |
| Entropy | $-p \log p - (1-p) \log 1 - p$ |
| Gini Index | $2p(1-p)$ |

# Issues with Multi-nomial Features

- Multi-nomial features: more than 2 possible values
- Comparing two features, one is binary, the other has 100 possible values, which one you expect to have higher mutual information with Y?
  - The conditional entropy of Y given this feature will be low
  - But is this meaningful?
  - This bias will inherently prefer such multinomial features to binary features
  - Method 1: To avoid this, we can rescale the conditional entropy:

$$\arg\min_j \frac{H(Y \mid X_j)}{H(X_j)} = \arg\min_j \frac{\sum_x P(X_j = x) H(Y \mid X_j = x)}{-\sum_x P(X_j = x) \log P(X_j = x)}$$

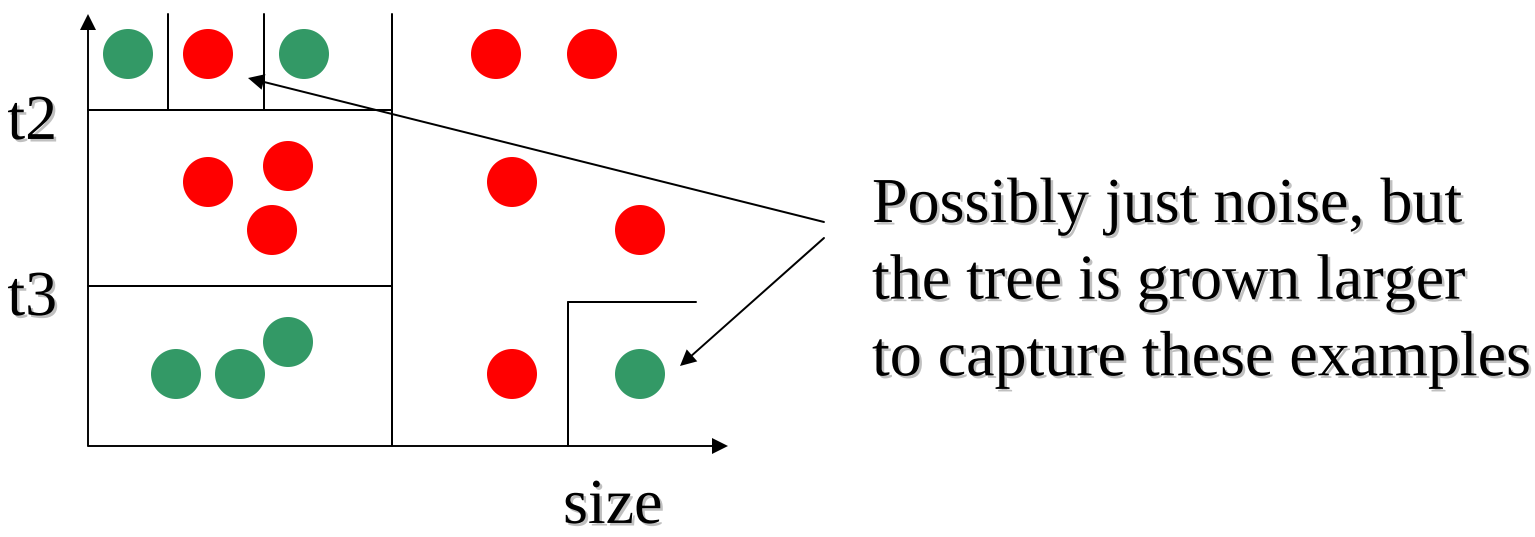
  - Method 2: Test for one value versus all of the others
  - Method 3: Group the values into two disjoint sets and test one set against the other

# Continuous Features

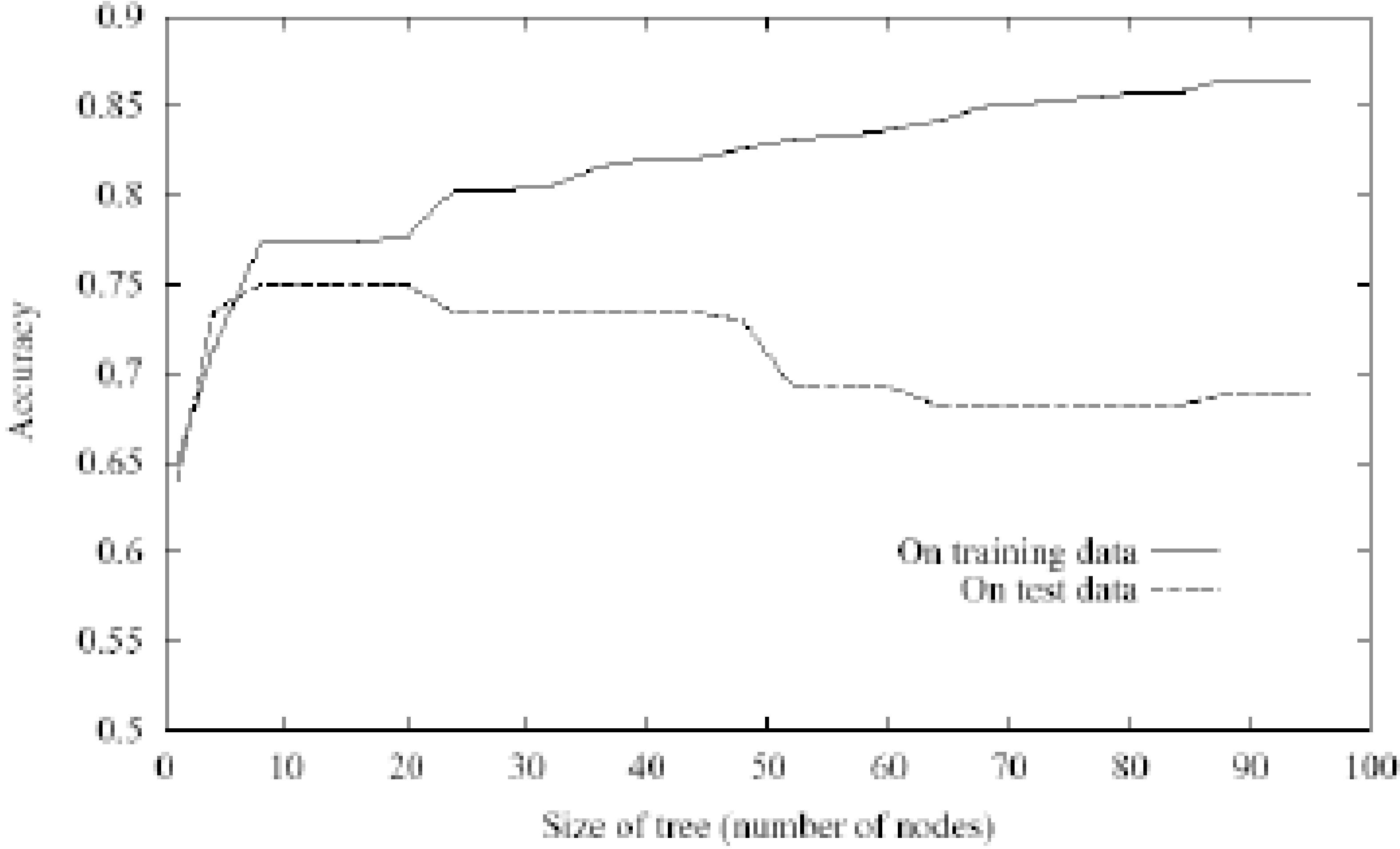
- Test against a threshold
- How to compute the best threshold $\theta_j$ for $X_j$?
  - Sort the examples according to $X_j$.
  - Move the threshold $\theta$ from the smallest to the largest value
  - Select $\theta$ that gives the best information gain
  - Trick: only need to compute information gain when class label changes

# Over-fitting

- Decision tree has a very flexible hypothesis space
- As the nodes increase, we can represent arbitrarily complex decision boundaries
- This can lead to over-fitting



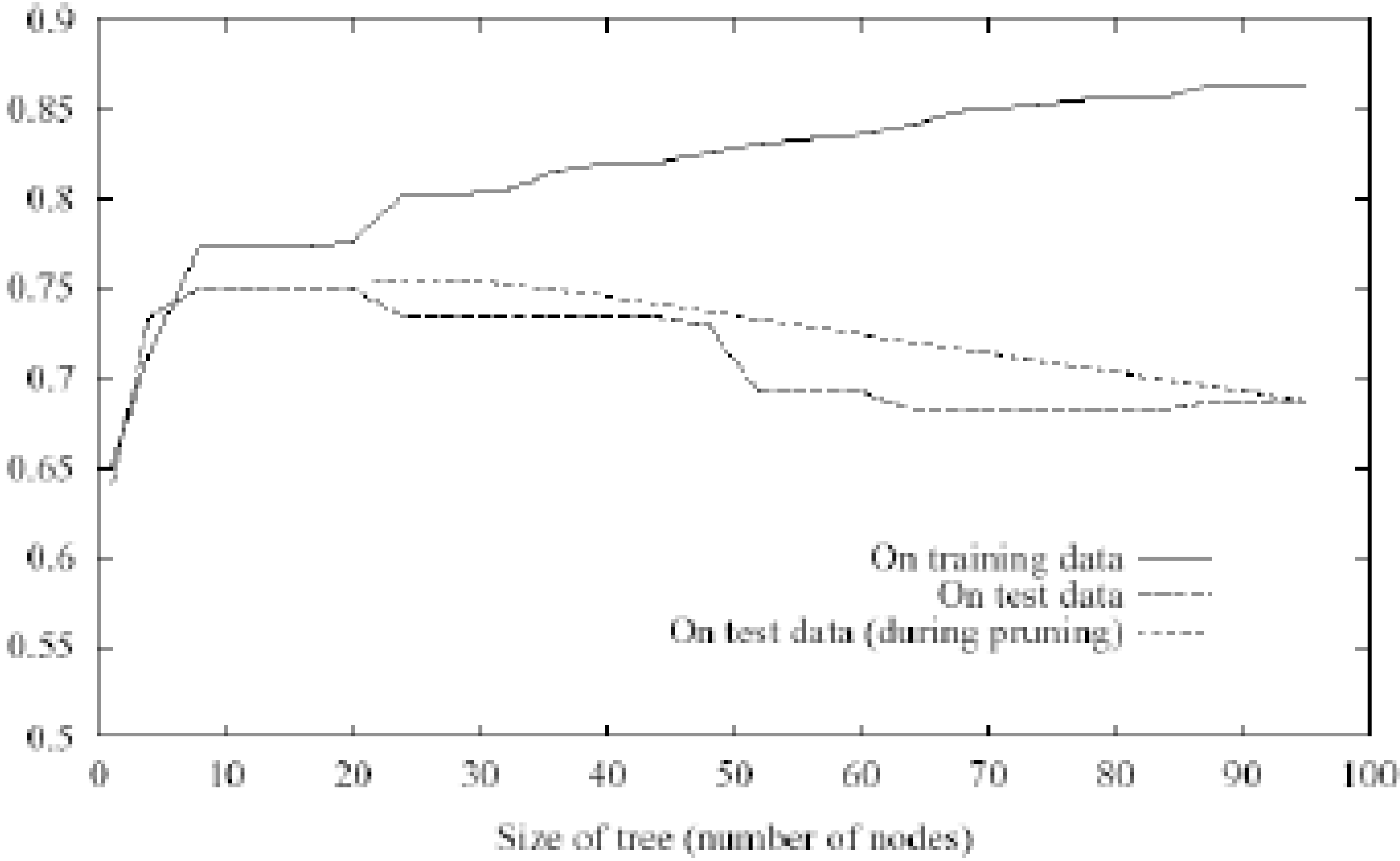
# Over-fitting



# Avoid Overfitting

- Early stop
  - Stop growing the tree when data split does not offer large benefit
- Post pruning
  - Separate training data into **training set** and **validating set**
  - Evaluate impact on validation set when pruning each possible node
  - Greedily prune the node that most improves the validation set performance

# Effect of Pruning



# Revisit some of the issues

- Is decision tree robust to outliers?
- Is decision tree sensitive to irrelevant features?
- Is decision tree computational efficient?