POLITECNICO DI MILANO

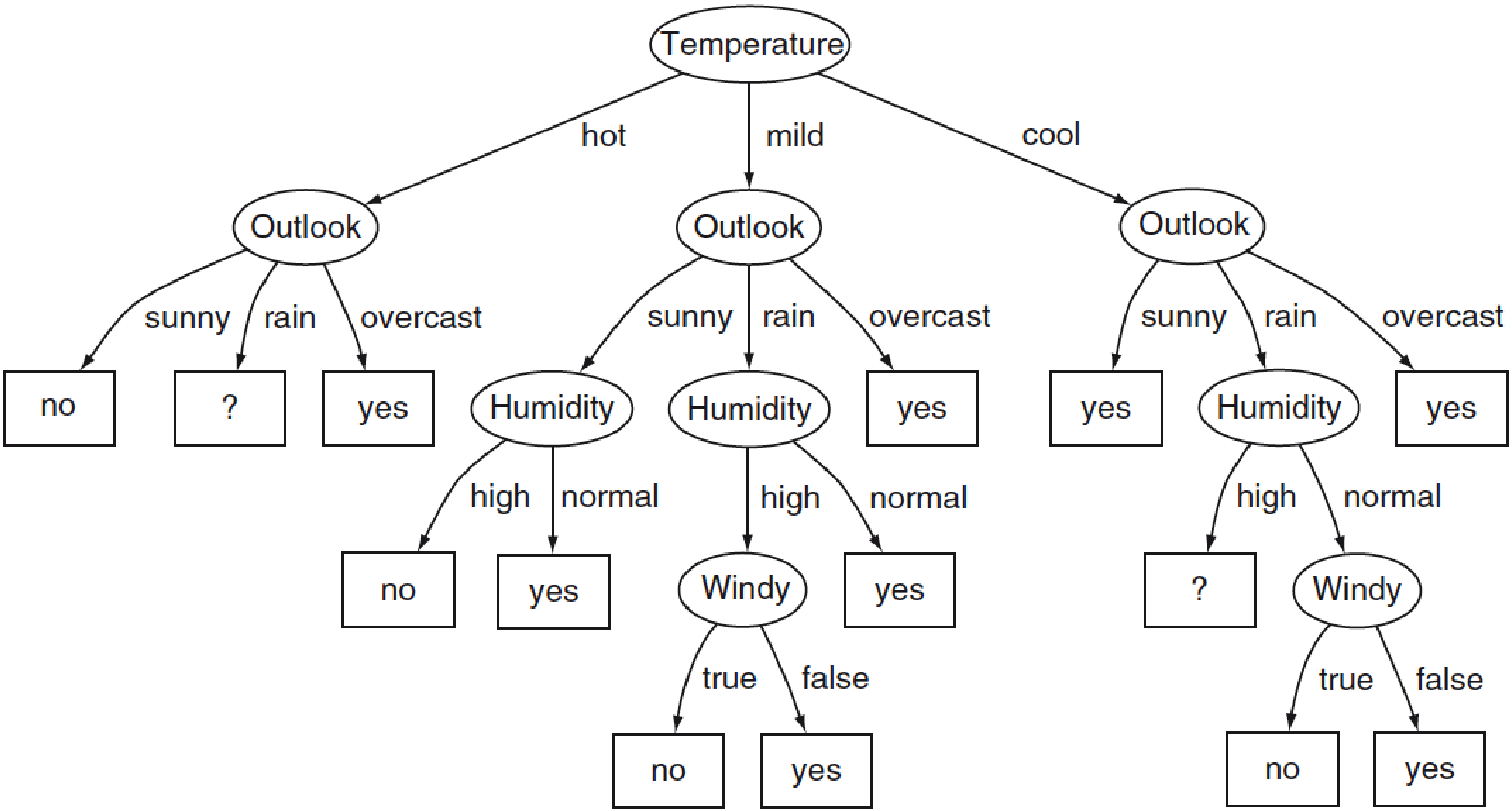
Decision Trees Pruning

Information Retrieval and Data Mining

Prof. Matteo Matteucci

Generalization and Overfitting in Decision Trees

The Perfect Fit …



Avoiding Overfitting in Decision Trees

- The generated tree may overfit the training data
  - Too many branches may reflect anomalies noise or outliers
  - Result is in poor accuracy for unseen samples
- Pre-pruning
  - Halt tree construction early
  - Do not split a node if this would result in the goodness measure falling below a threshold (difficult to choose)
- Post-pruning
  - Remove branches from a “fully grown” tree
  - Use a set of data different from the training data to decide which is the “best pruned tree”

Pre-Pruning

- Usually based on statistical significance test
- Stop growing the tree when there is no statistically significant association between any attribute and the class at a particular node
- High risk of premature halt
  - If initially no individual attribute exhibits any interesting information about the class
  - The structure will become visible only in fully expanded tree
  - Pre-pruning won’t expand the root node

Post-Pruning

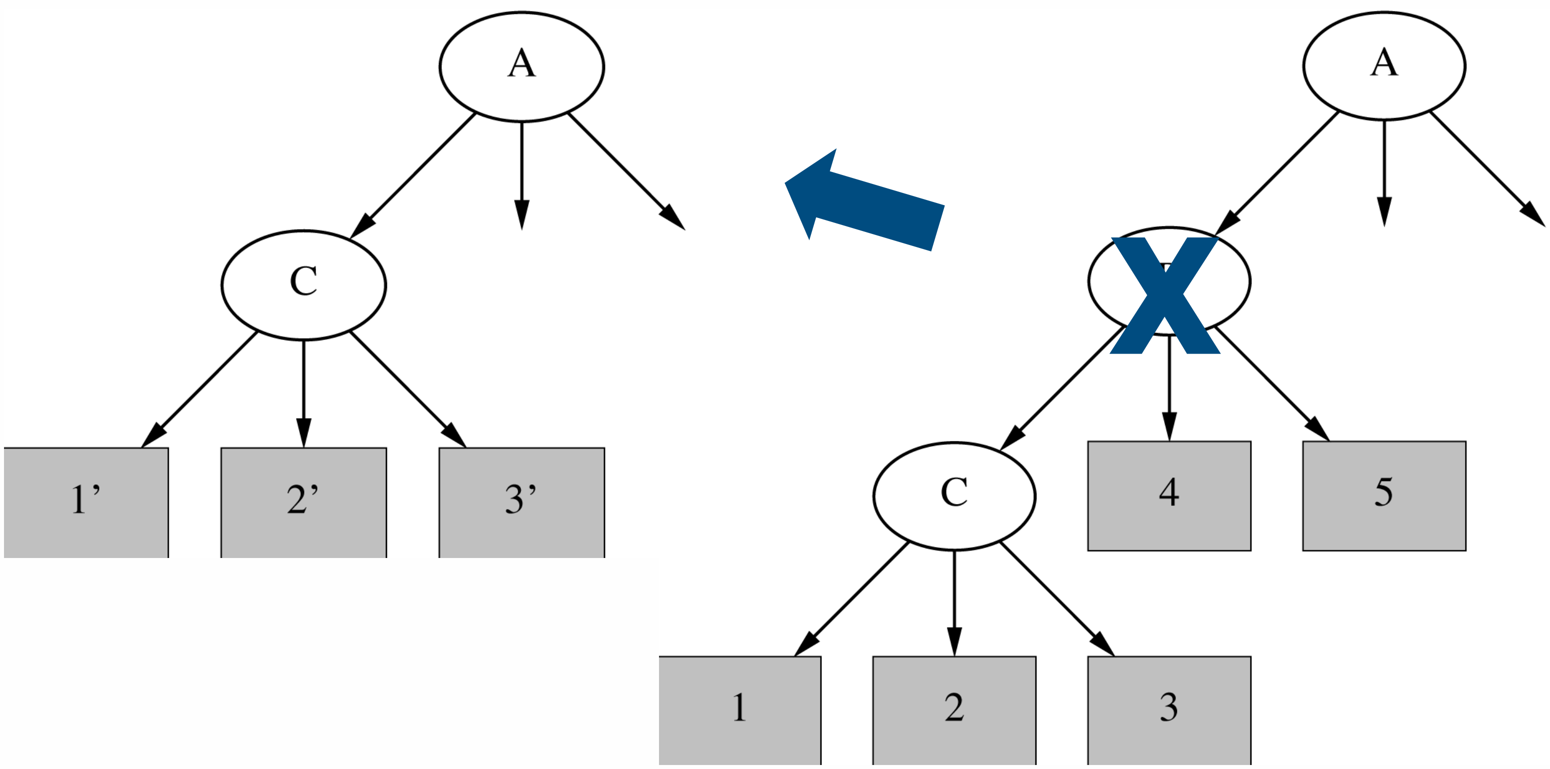
- First, build full tree, then prune it
  - Fully-grown tree shows all attribute interactions
  - But some subtrees might be due to chance effects

- Two pruning operations
  - Subtree raising
  - Subtree replacement

- Possible strategies to select the subtree
  - Error estimation
  - Significance testing
  - MDL principle

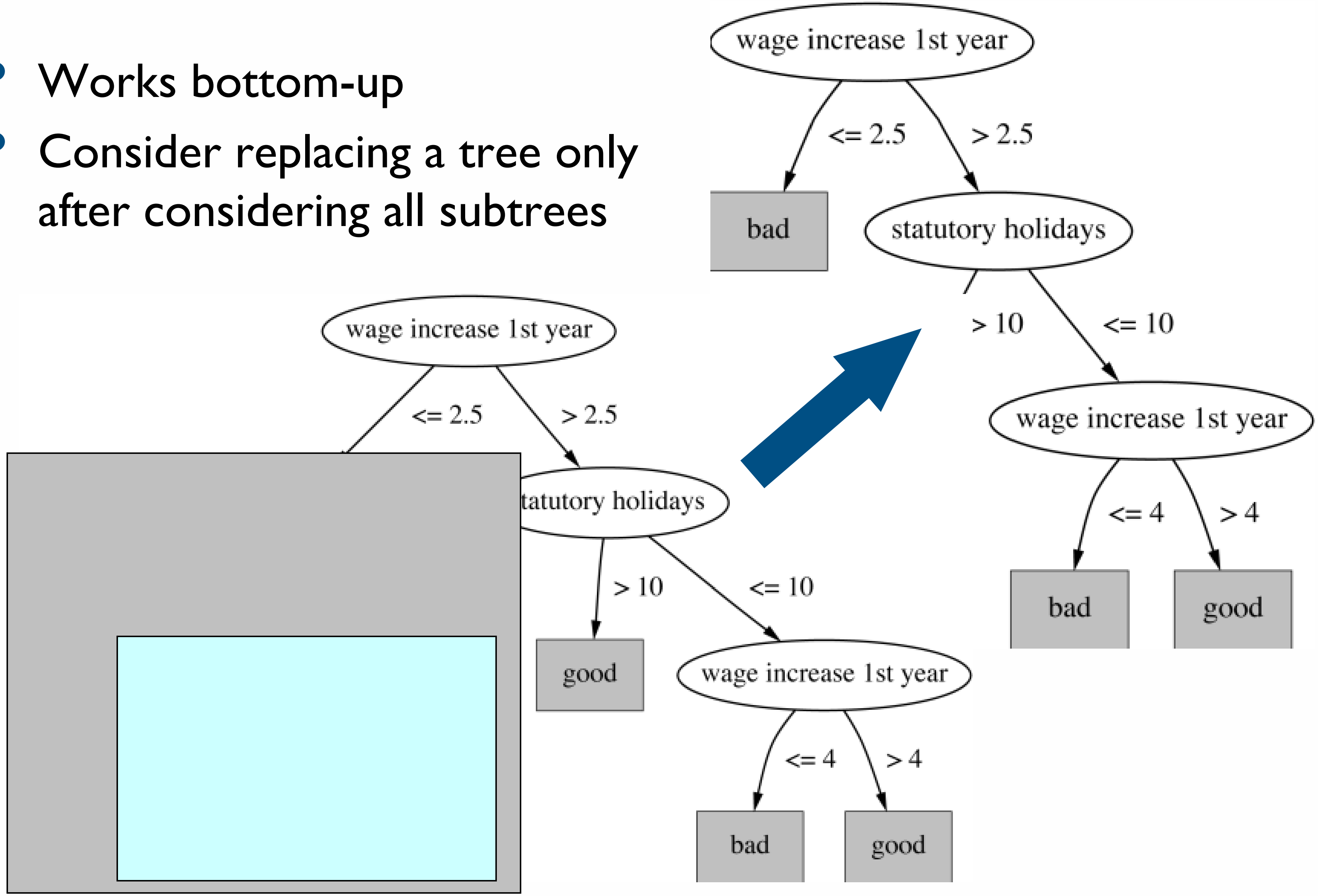
Subtree Raising

- Delete node and redistribute instances
  - Redistribution is slower than replacement



Subtree Replacement

- Works bottom-up
- Consider replacing a tree only after considering all subtrees



Estimating Error Rates (for Pruning)

- Prune only if it reduces the estimated error
  - Error on the training data is NOT a useful estimator (Why it would result in very little pruning?)
  - A hold-out set might be kept for pruning ("reduced-error pruning")

- Example (C4.5's method)
  - Derive confidence interval from training data
    - Standard Bernoulli-process-based method
    - Shaky statistical assumptions (based on training data)
  - Use a heuristic limit, derived from this, for pruning

Mean and Variance of Expected Errors

- Mean and variance for a Bernoulli trial are p and $p(1-p)$
- Expected error rate $f = S/N$ for large enough N follows a Normal distribution:

$$f \sim N(p \, , \, p(1-p)/N)$$

- The C% confidence interval [-z< X < z] for random variable with 0 mean is given by:

$$P[-z<X<z] = C$$

- With a symmetric distribution,

$$C = 1\text{-}2xP[X>z]$$

Confidence Limits Normal Distribution

- Confidence limits for the normal distribution with 0 mean and unit variance is …

- Thus:

$$P[-1.65<X<1.65] = 90\%$$

- To use this we have to reduce our random variable f to have 0 mean and unit variance

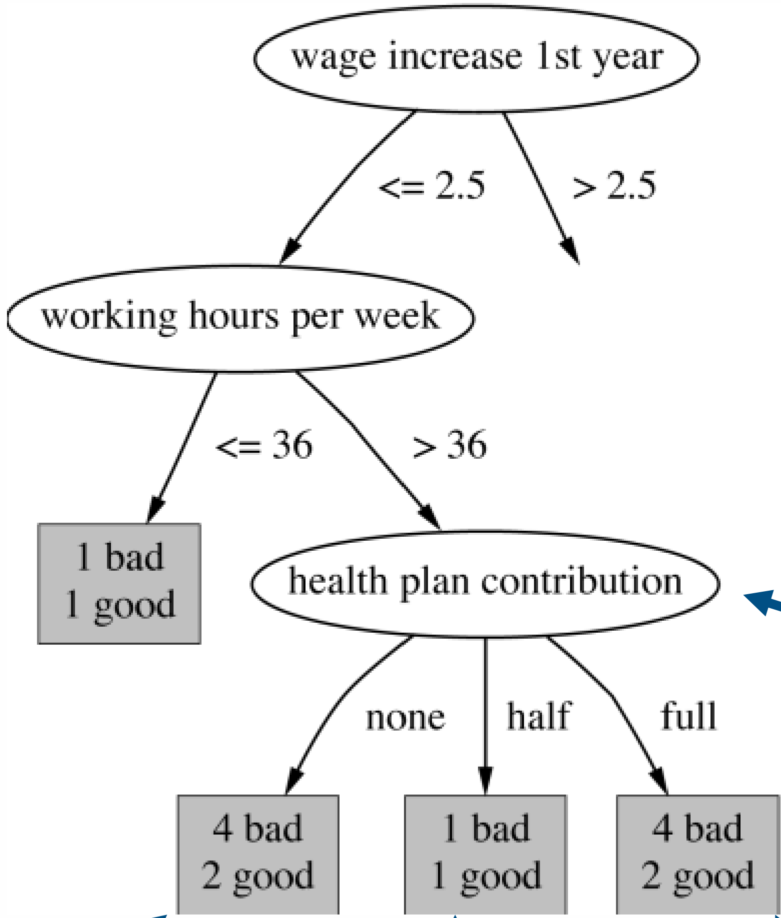
$\Pr[X \geq z]$	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
25%	0.69
40%	0.25

C4.5's Pruning Method

- Given the error f on the training data, the upper bound for the error estimate for a node is computed as

$$e = \left(f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) \Bigg/ \left(1 + \frac{z^2}{N} \right)$$

- If c = 50% then z = 0.69 (from normal distribution)
  - f is the error on the training data
  - N is the number of instances covered by the leaf



f = 5/14
e = 0.46
e < 0.51
so prune!

f=0.33
e=0.47

f=0.5
e=0.72

f=0.33
e=0.47

Combined using ratios 6:2:6 gives 0.51