CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

Recap

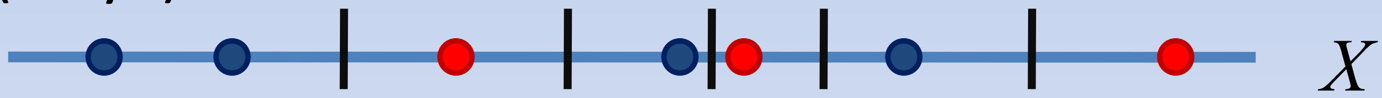
- To choose a test, we look for an attribute that provides i______ about the l___. A quantity that encodes this is the e______ of a random variable.
- E_____ is the expected length of the s_____ l_____ d_____ of a random variable.
- I_____ g_____ is the r_____ of e_____ of the class variable b___ and a_____ partitioning.
- What problem arises with nominal features and info gain?
- We can attempt to resolve this issue by adjusting the split criterion. GR(X)=_____/_____. This works because _______.
- We partition on continuous features by considering all tests of the form _____________.
- We only need to consider values that ________.

Today

- Decision Tree Induction (Ch 3, Mitchell)
- Overfitting and overfitting control

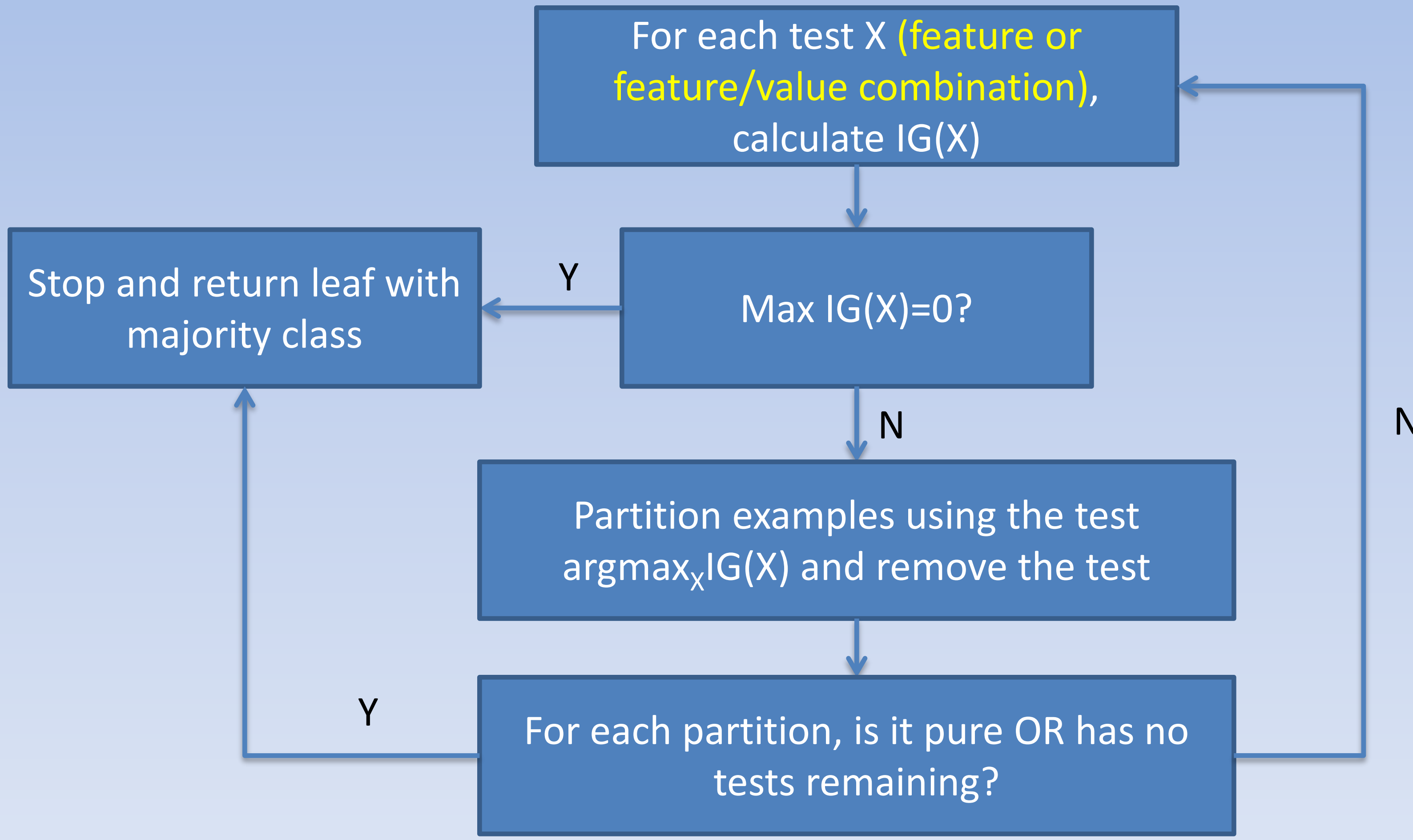
Continuous Attributes

- Cannot test for equality
- Consider all Boolean tests of the form $X \geq v$ (or $X \leq v$)
  - Only values of interest are those v that separate adjacent training examples with different classes (why?)



- Note: In this case, the attribute cannot be removed, though the test ((attribute, value) tuple) can be

ID3 Algorithm---Training phase



Example

Color	Area	Shape	Class Label
red	0.1	circle	1
blue	0.2	triangle	1
green	0.3	triangle	0
green	0.3	circle	0
green	0.4	square	0
red	0.4	triangle	1
blue	0.6	circle	0
red	0.7	square	0
blue	0.8	square	0

Example

Color	Area	Shape	Class Label
~~red~~	~~0.1~~	~~circle~~	~~1~~
~~blue~~	~~0.2~~	~~triangle~~	~~1~~
green	0.3	triangle	0
green	0.3	circle	0
green	0.4	square	0
red	0.4	triangle	1
blue	0.6	circle	0
red	0.7	square	0
blue	0.8	square	0

Example

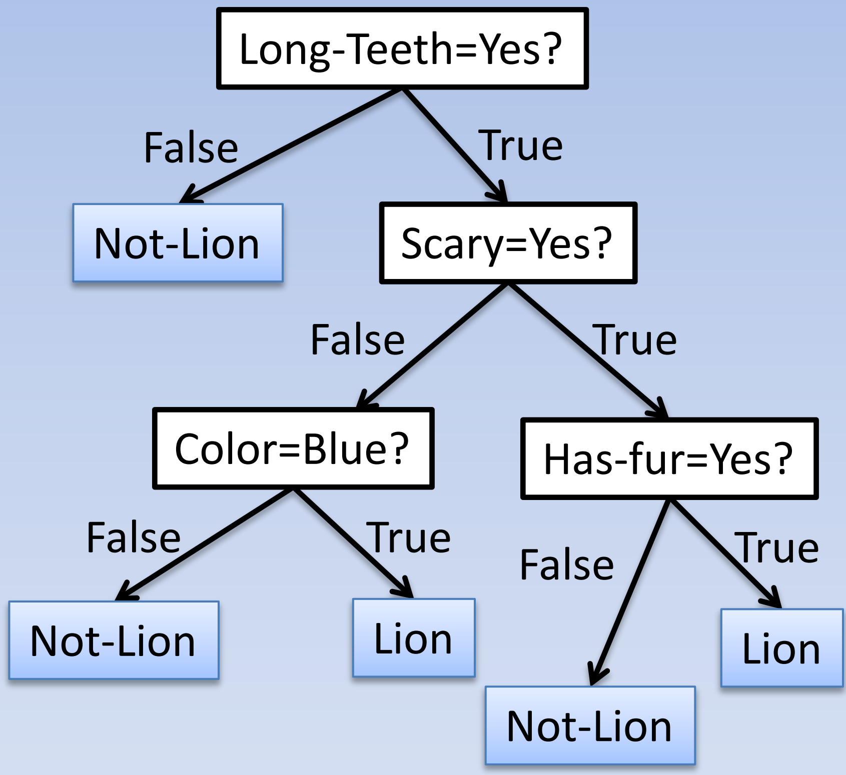
Color	Area	Shape	Class Label
~~red~~	~~0.1~~	~~circle~~	~~1~~
~~blue~~	~~0.2~~	~~triangle~~	~~1~~
~~green~~	~~0.3~~	~~triangle~~	~~0~~
~~green~~	~~0.3~~	~~circle~~	~~0~~
~~green~~	~~0.4~~	~~square~~	~~0~~
red	0.4	triangle	1
~~blue~~	~~0.6~~	~~circle~~	~~0~~
red	0.7	square	0
~~blue~~	~~0.8~~	~~square~~	~~0~~

An Issue

- Given enough features, ID3 will usually be able to fit training examples exactly (i.e. every leaf is pure), because the tree can be grown as much as needed

- But real data is noisy

Overfitting



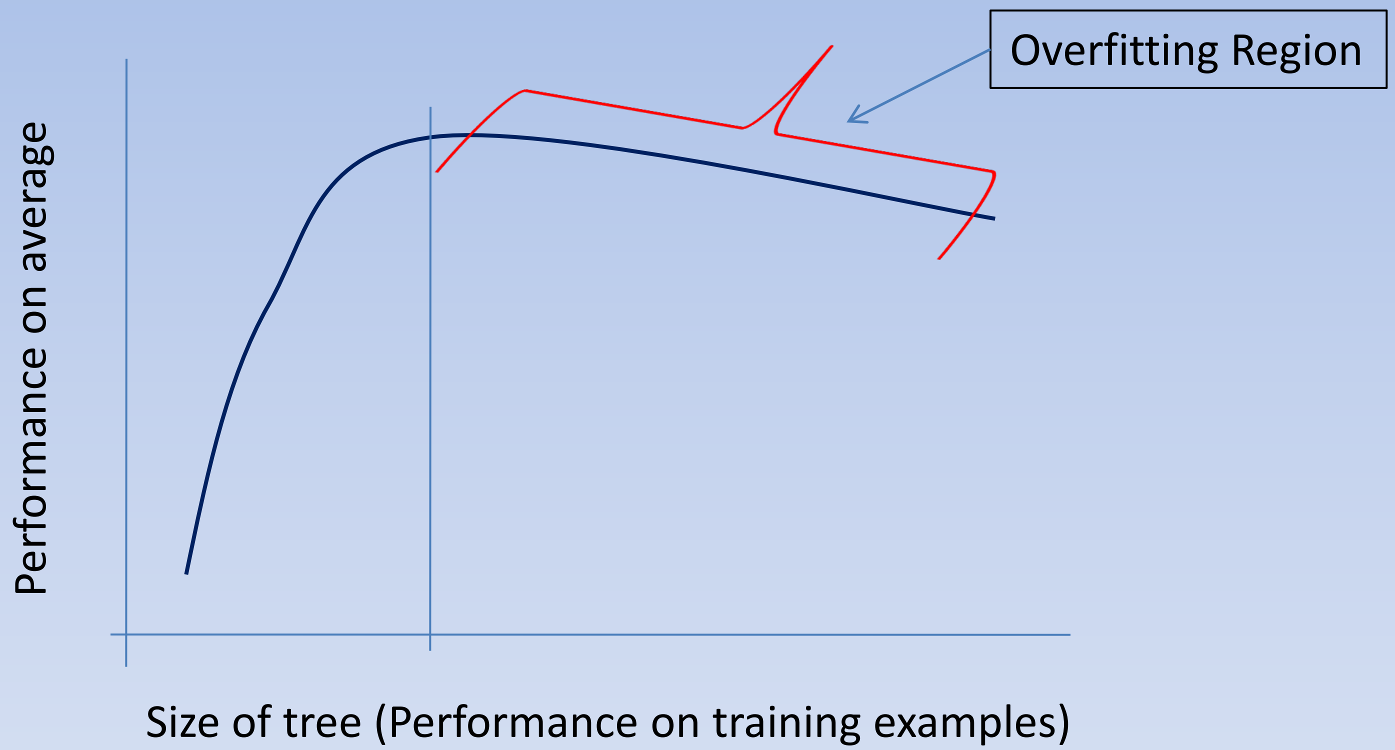
	Has-fur?	Long-Teeth?	Scary?	Color?	*Lion?*
$Animal_1$	Yes	No	No	Green	No
$Animal_2$	No	Yes	No	Black	No
$Animal_3$	Yes	Yes	Yes	Golden	Yes
$Animal_4$	Yes	Yes	No	Blue	Yes
$Animal_5$	Yes	Yes	Yes	Tawny	Yes

Overfitting

- If a learned concept h has
  - Higher performance (lower error) on the training examples, BUT
  - Lower performance (higher error) on average across all examples

than some alternative concept h' in the same hypothesis space, h is said to have overfit to the training examples

Overfitting



Controlling Overfitting

- Introduce a restriction on the hypothesis space to prevent overly-complex hypotheses from being learned
  - Early Stopping
  - Post Pruning

Early Stopping

- Standard algorithm stops growing the tree when $IG(X)=0$ for all X
- Early stopping stops growing the tree when $IG(X) \leq \varepsilon$, for some chosen ε
- Sensitive to choice of ε
- Easy to implement, but does not work very well in practice

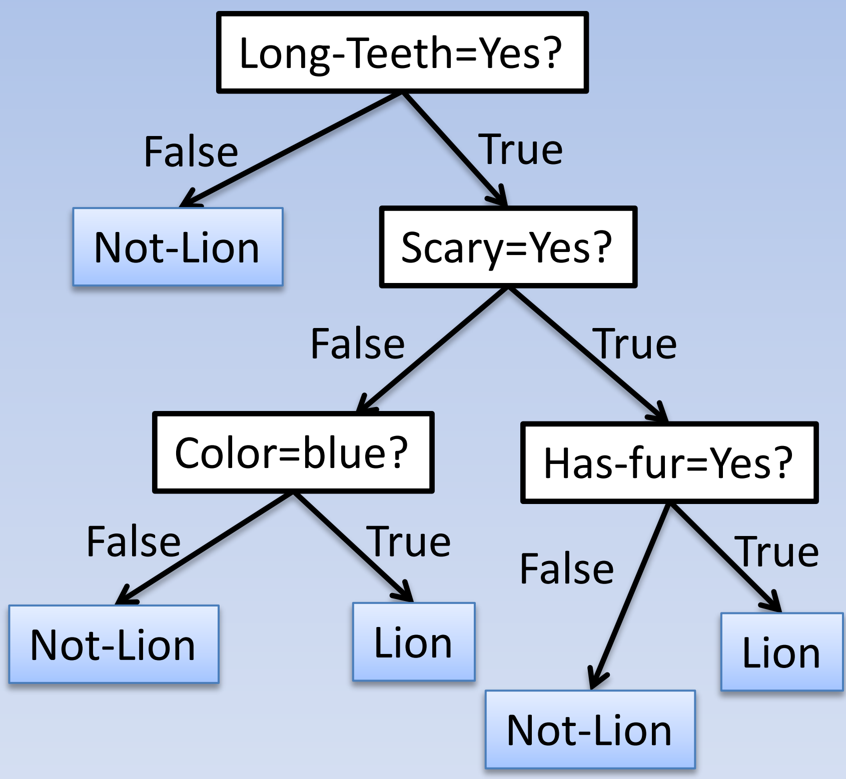
Greedy post-pruning

- Hold aside some training examples at start (validation set)
- Grow tree as usual on remainder
- Then run a *greedy pruning* algorithm

Greedy post-pruning

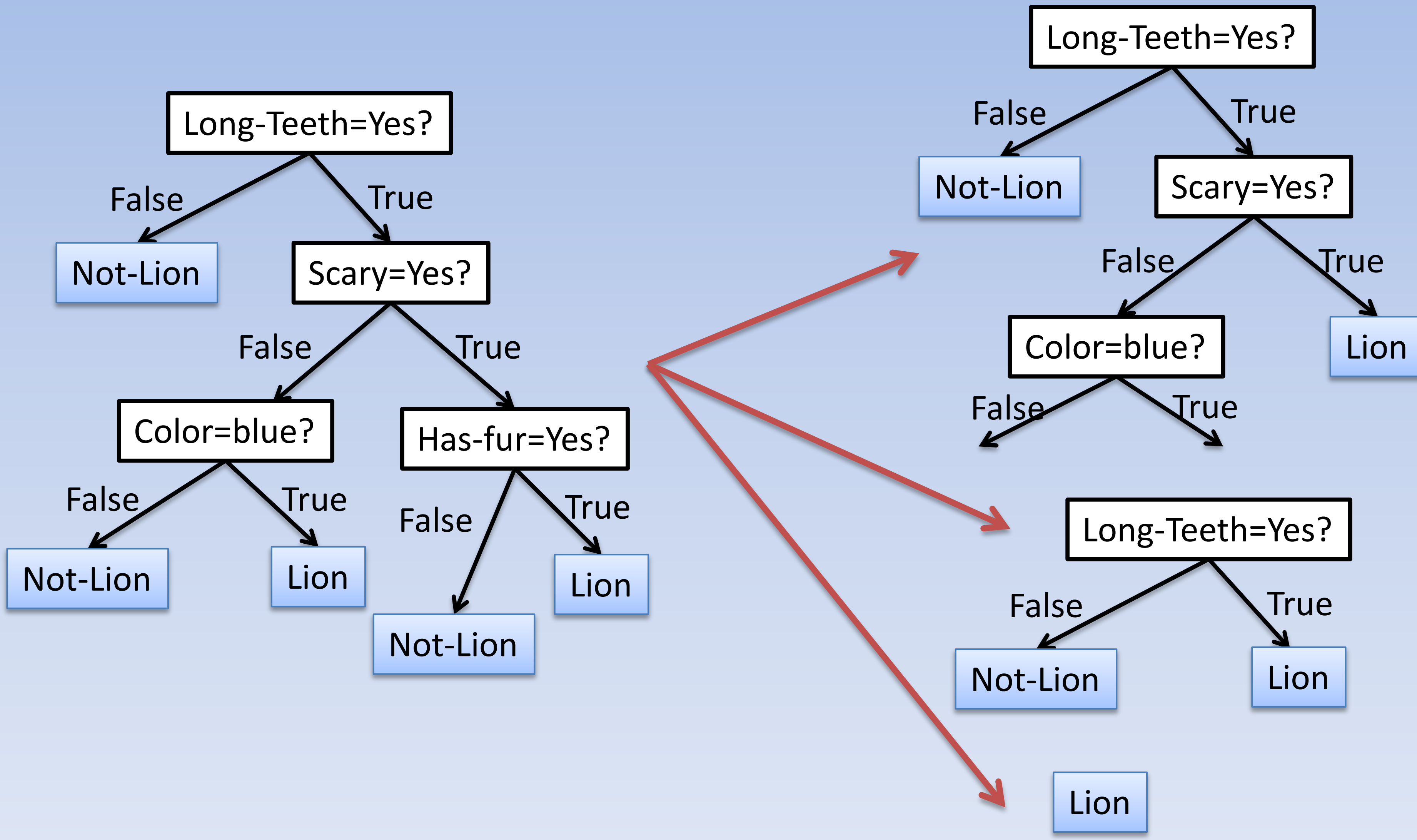
- For each internal node, construct a tree without that node
  - Convert node to leaf by predicting majority class
  - Delete subtree below node
- Evaluate this *pruned* tree on the validation set
- Find the single node that improves performance the most over the unpruned tree and remove it
- Repeat steps above until no node removal improves performance

Greedy Post Pruning



	Has-fur?	Long-Teeth?	Scary?	Color?	*Lion?*
$Animal_1$	Yes	No	No	Green	No
$Animal_2$	No	Yes	No	Black	No
$Animal_3$	Yes	Yes	Yes	Golden	Yes
$Animal_4$	Yes	Yes	No	Blue	Yes
$Animal_5$	Yes	Yes	Yes	Tawny	Yes
$Animal_6$	No	Yes	No	Blue	No

Greedy Post Pruning



Greedy Post Pruning