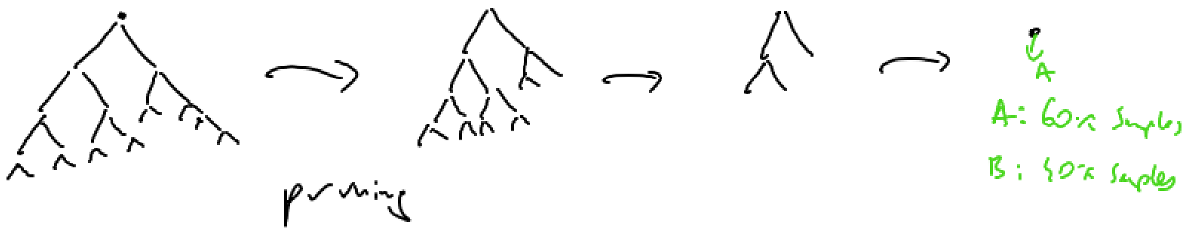
Decision trees

- constructing
- pruning (cost-complexity pruning)

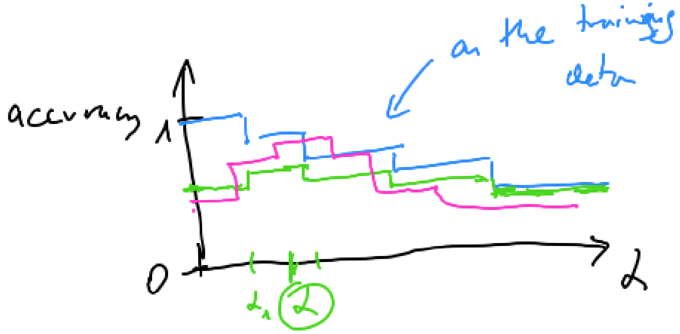
↑ depends on some $\alpha \geq 0$



How to choose a reasonable α for pruning?

- test ~~on~~ the testing data

usually one sees the following behavior:



- use cross-correlation: we divide the training dataset into K ~~dat~~ parts, $A_1, A_2, \dots, A_k$ (randomly)

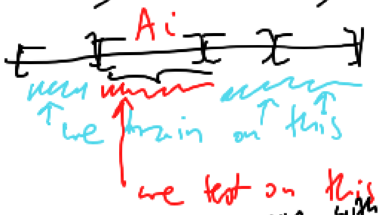
$A_i \cap A_j = \emptyset$ for $i \neq j$, $\bigcup A_i = \{\text{training dataset}\}$

for each $i = 1, 2, \dots, k$: train the tree on $\bigcup_{j \neq i} A_j$

and test it on A_i

check the performance of diff. α's

$\rightarrow$ find α $\rightarrow$ ~~ret~~train the tree on the original training set and prune with this α



Confusion matrix

	actual −	actual +
predicted −	a_{11}	a_{12}
predicted +	a_{21}	a_{22}

a_{11} = #{samples in class − correctly predicted}

a_{12} = #{ −"− + predicted as −}

e.g.

		actual −	actual +
predicted	−	93	5
	+	7	45

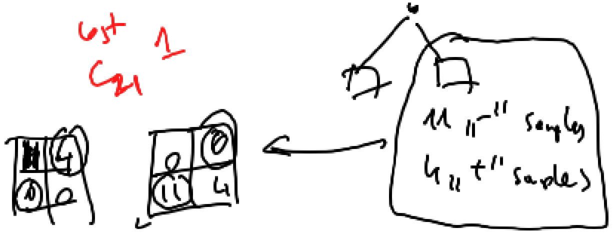
		actual −	actual +
predicted	−	96	11
	+	4	39

cost c_{12} 9

cost c_{21} 1

Often errors of one type are more costly than those of the other type

Decision trees (and many other classifiers) can be easily converted to cost-sensitive classifiers by changing the way we make final classification



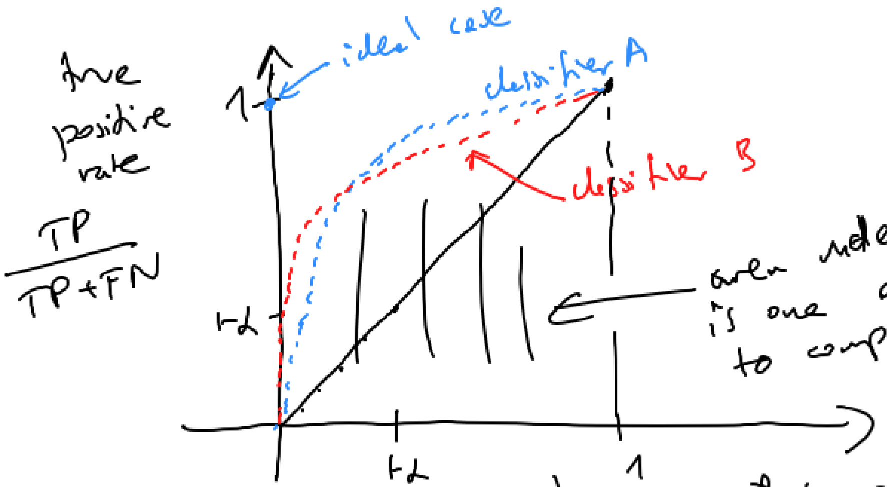
one may classify this as of class "+" if

$$\frac{\#\{\text{samples } +\}}{\#\{\text{samples}\}} \geq \frac{c_{12}}{c_{12}+c_{21}} = p^*$$

$\frac{4}{15}$ $\qquad$ $\frac{1}{10}$

So far we used $p^* = \frac{1}{2}$

ROC curve
↑
(receiver operating characteristic)



area under this curve (AUC) is one of the ways to compare classifiers

false positive rate

$\frac{FP}{FP+TN}$

actual

predicted	−	+
−	TN	FN
+	FP	TP

$\frac{TP}{TP+FN}$

trivial classifier

predicted

	−	+
−	$\alpha \cdot N_1$	$\alpha \cdot N_2$
+	$(1-\alpha)N_1$	$(1-\alpha)N_2$

1-α

$\frac{FP}{FP+TN} = \frac{(1-\alpha)N_2}{N_2} = 1-\alpha$

trivial classifier:

choose some α

predict a sample to be of

class − with prob α

+ with prob 1-α

actu

Pr.	−	+
−		$\alpha \cdot N$
+		$(1-\alpha)N$

Bagging (bootstrap aggregation)

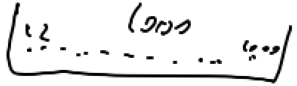
A general method, but often used with trees.

1) take B training sets

2) build a separate model for each

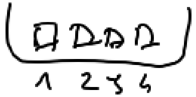
3) average / take majority vote of resulting predictions

$\uparrow$ regression $\qquad \uparrow$ classification



Ad (1): Draw $\tilde{n}$ samples from the original set of n observation with replacement

(usually $\tilde{n} = n$)



Repeat B times to obtain B training sets

If n is large and we draw $\tilde{n} = n$ samples with replacement, then we may expect $\approx 0.63 \cdot n$ ~~of the original samples~~ unique samples

$$\underset{\left(1 - \frac{1}{e}\right)}{\underset{\uparrow}{0.63}}$$

1: $(1, 3, 1, 4)$

2: $(3, 3, 2, 1)$

3: $(4, 2, 3, 1)$

new training sets

For bagging one may use out-of-bag error for example to find the pruning threshold α

- Each sample occurs in ca. 63% of the training datasets

  and does not occur in ca. 37% -"- ———

  So this sample x may be used to check the performance of ca. 37% of the models that were trained on a training set <u>not</u> containing x

Random forests

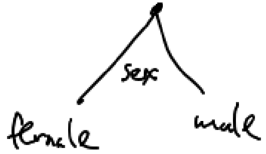
$x \in \mathbb{R}^n$ — n predictors

They are like bagging, but with 2) replaced by

2') select some random subset of p predictors out of n
and build a model using just these p predictors
(and a training set selected in step 1)

the advantage: different models will have smaller correlation

example: passenger of Titanic



reasonable choices for p: $p \approx \lfloor \sqrt{n} \rfloor$ for classification
$p \approx \lfloor n/3 \rfloor$ for regression

(but p is a hyperparameter that can be also subject to optimisation)

In step 2) or 2)' one may take a full tree without pruning
or also a small tree (heavily pruned)