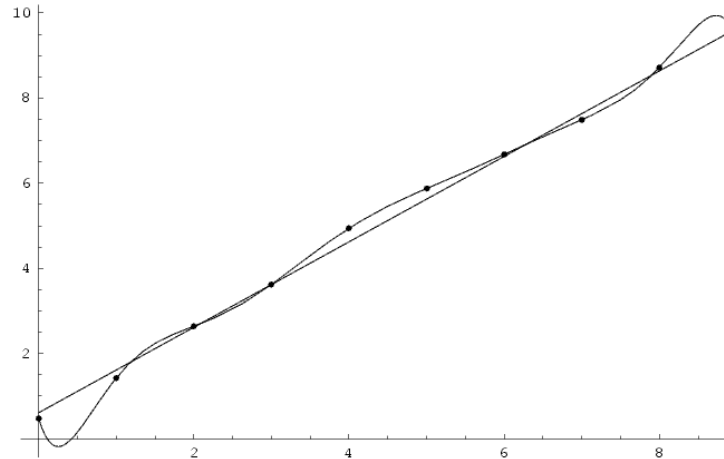


Real-world decision tree algorithms

- handle continuous data (e.g., temperature)
- avoid **overfitting** the data
- deal with:
 - weighted attribute costs
 - data with missing attribute values
- perform **pruning**

Overfitting



- Which regression curve is a better fit?
- Problem: might be fitting noise rather than data
- Pruning: prevent overfitting in decision tree by avoiding recursive split on attributes that are not obviously relevant
- But how to determine irrelevance?

Statistical Significance Test

- Take a sample of size v , consisting of p positive and n negative examples
- Divide the sample into subsets based on classification of the attribute
- for each subset, let p_i and n_i be the number of positive and negative examples
- Calculate expected # of positive and negative examples, assuming attribute is irrelevant

$$\hat{p}_i = p \times \frac{p_i + n_i}{p + n} \qquad \hat{n}_i = n \times \frac{p_i + n_i}{p + n}$$

Statistical Significance Test

- Calculate expected # of positive and negative examples, assuming attribute is irrelevant

$$\hat{p}_i = p \times \frac{p_i + n_i}{p + n} \quad \hat{n}_i = n \times \frac{p_i + n_i}{p + n}$$

- Calculate total deviation:

$$D = \sum_i \frac{(p_i - \hat{p}_i)^2}{\hat{p}_i} + \frac{(n_i - \hat{n}_i)^2}{\hat{n}_i}$$

- If attribute is “irrelevant”, can prune it from tree
- Advantage: such a tree deals well with noise

Cross-validation

- need to validate the tree with a test set
- what happens if test set gives errors?

Test Set Results

<u>color</u>	<u>decision</u>	
red	F	x
red	F	x
green	T	x
red	T	✓
green	F	✓

New Results

<u>color</u>	<u>decision</u>
red	F
red	F
green	T
red	T
green	F

Example: Golf Decision

Outlook	Temperature	Humidity	Windy	Decision
sunny	85	85	false	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play

Hint

- start by considering binary variables (e.g., outlook, windy)
- when they no longer help provide unambiguous classification to all remaining examples:
 - consider continuous variables
 - choose a useful split point in their range (e.g. temperature > 83)

C4.5 Decision Tree Output

Decision Tree:

outlook = overcast: Play (4.0)

outlook = sunny:

| humidity <= 75 : Play (2.0)

| humidity > 75 : Don't Play (3.0)

outlook = rain:

| windy = true: Don't Play (2.0)

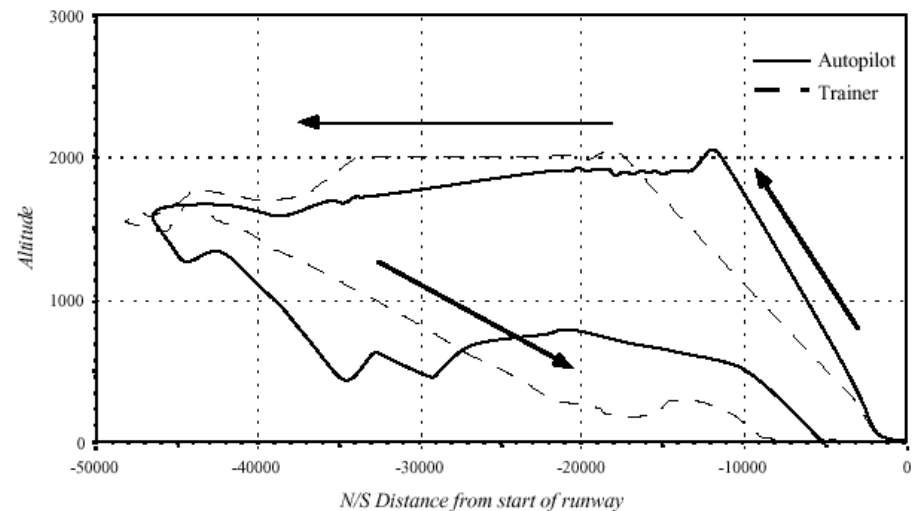
| windy = false: Play (3.0)

Evaluation on training data (14 items):

Before Pruning		After Pruning		Estimate
Size	Errors	Size	Errors	
8	0(0.0%)	8	0(0.0%)	(38.5%) <<

Sammut: Learning to Fly

- take off and fly to altitude of 2000 feet
- level out, fly to distance of 32000 feet
- turn right to 330°
- at 42,000 feet, turn back to runway
- line up on the runway
- descend on the runway, keeping in line
- land



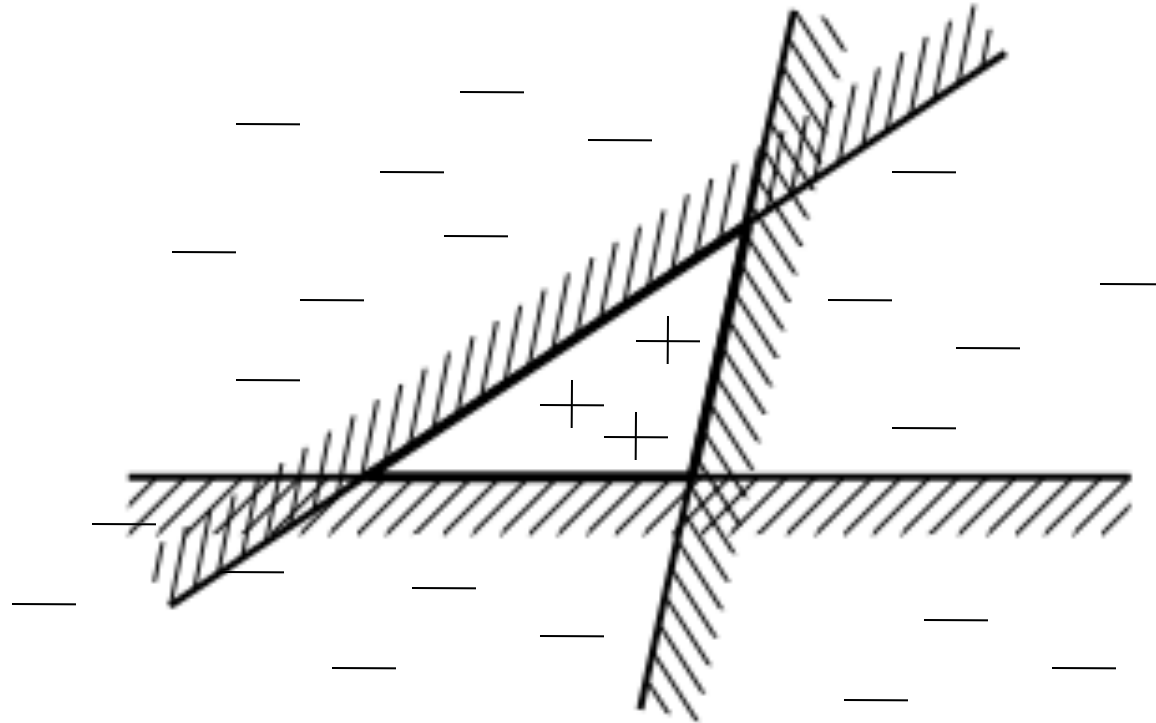
Issues Faced

- pruning
- branching
- dealing with real-world issues
 - noise in data
 - causality
 - delay between sensing and reaction
 - different strategies accomplish same goal

Ensemble Learning

- We've seen how to learn to make predictions from a single hypothesis, e.g., decision trees
- What if we generated an ensemble of hypotheses and used their combination to make predictions?
- If errors made by each hypothesis are independent, the probability that a majority of them will make the same error is very small

Three linear hypotheses in ensemble



Boosting

- Weighted training set: each example has a weight $w_{ij} \geq 0$ representing its importance during learning
- Start learning with $w_{ij} = 1$ for all examples in training set \rightarrow hypothesis h_1
- Increase w_{ij} for all misclassified examples in h_1 and decrease for others; learn again \rightarrow hypothesis h_2
- ...
- Until M hypotheses generated

How Boosting Works

height of rectangle indicates weight

size of decision tree indicates weight of hypothesis in ensemble

