Machine learning

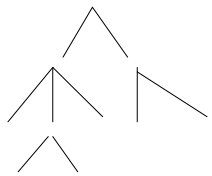
Maarten van Someren

University of Amsterdam

Main points

- Decision tree learning as an example of ML
- ML and inference
- Metamodel: training data, hypothesis space, "learning bias"
- A few more methods at a glance
- Overfitting
- Why does it work? No Free Lunch

What is a decision tree?



Basic Algorithm

ID3(*Examples*, *TargetAttr*, *Attributes*):

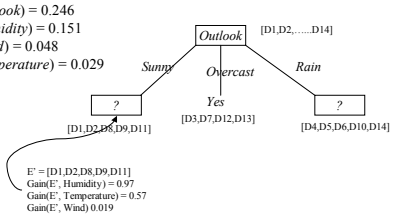
- Create a *Root* for the tree;
- If all *Examples* are positive (negative), label Yes (No);
- If *Attributes* is empty, label with most common class value in *Examples*;
- Select the best A from *Attributes*, and for each value vi of A,
  - add a new branch for A=vi, let *E'* be the examples with A=vi;
  - if E' is empty, then add a leaf node, else add a node=ID3(*E'*, *TargetAttr*, *Attributes-{A}*)

Example

- [*Day, Outlook, Temp, Humidity, Wind,* ***PlayTennis***]
  - (D1 Sunny Hot High Weak No)
    (D2 Sunny Hot High Strong No)
    (D3 Overcast Hot High Weak Yes)
    (D4 Rain Mild High Weak Yes)
    (D5 Rain Cool Normal Weak Yes)
    (D6 Rain Cool Normal Strong No)
    (D7 Overcast Cool Normal Strong Yes)
    (D8 Sunny Mild High Weak No)
    (D9 Sunny Cool Normal Weak Yes)
    (D10 Rain Mild Normal Weak Yes)
    (D11 Sunny Mild Normal Strong Yes)
    (D12 Overcast Mild High strong Yes)
    (D13 Overcast Hot Normal Weak Yes)
    (D14 Rain Mild High Strong No)

An Example Tree Built

Gain(*S, Outlook*) = 0.246
Gain(*S, Humidity*) = 0.151
Gain(*S, Wind*) = 0.048
Gain(*S, Temperature*) = 0.029



Best attribute

- Best predictor
- Criteria:
  - Predict most frequent class + count prediction errors
  - Information gain

Information gain

- Entropy:
  - S is a sample of training examples
  - P+ is the proportion of positive examples in S
  - P- is the proportion of negative examples in S
- Entropy measures the impurity of S
- Entropy(S) = - (p+ . 2logp+) – (p- . 2logp-)
- Entropy(S) = expected number of bits needed to encode class (p+ or p-)of randomly drawn member of S (under the optimal, shortest-length code)

- Information theory: optimal length code assigns - ^{2}logp bits to message having probability p.
- So, expected number of bits to encode class of random member of S:
- (P+ - ^{2}logp+) + (p- . -^{2}logp-)
- Equals the information

Information GAIN

- Gain(S,A) = expected reduction in entropy due to sorting on A
- = sum of entropies in subsets, weighed by their proportion
- = entropy(S) – Σ_v |v|/|S|.Entropy(S_v)

Numerical variables

- Define intervals (and use these as tests in trees)
- Try out all possible splits
- Carry variable along for further splits

Missing data

- Data table incomplete
- We cannot assign each datapoint to a leaf and continue)
- Solutions:
  1. Estimate missing value
  2. “split” case into parts (!?)

Example applications

- Many; most popular DM method
- Properties of cell -> cancer or not
- “Behavioral cloning”: learn decision tree from state – expert action pairs
- Controlling simulated airplane
- Controlling steel roller
- Customer goes to other company or not

Metamodel, ontology

- Data (examples, instances)
  - Predictors, description; predicted, target
  - Features, attributes, (predictive) variables
  - (target) class (variable)
- Hypothesis, (prediction) model
- Hypothesis set/language
- Learning task (estimation, data mining, exploratory data analyis task): classification, prediction
- Performance task: classification, prediction

Evaluating a decision tree

- Why? Research, application, during learning
- Predictions on new data from same domain
- Evaluation Measures:
  - Accuracy: number of correct predictions
  - Precision/accuracy/F1
  - Area under learning curve
  - Comprehensibility, usefulness, …
- NB: during learning: compare extending decision tree: extension should be best ***and*** new tree should be better than old tree!

Is a tree with fewer prediction errors on the training data really better?

- Answer: **NO** (!!!?)
- There may be two hypotheses h1 and h2 such that:

  error-train(h1) > error-train(h2)

  and

  error-domain(h2) < error-domain(h2)

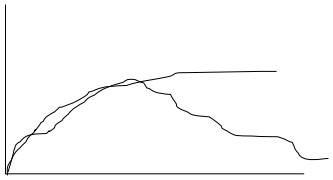
Overfitting: intuition

- Model, tree may model noise for part of the instance space
- Suppose at a current leaf we have:
  - #pos = 6; #neg = 1
- Two options:
  - Find additional attributes that splits
  - Stop and assign “pos” to leaf
- If the domain is noisy then “pos” may be a better estimate for the entire part of the domain under this node

Overfitting: more intuition

- caseID as attribute:
  - classifies perfectly on training data
  - Useless as predictor
- Numerical attribute:
  - Many values
  - Classifies very well on data
  - Far less useful as predictor

With/without pruning: accuracy by complexity



Not only with decision trees

- Neural networks:
  - Causes of overfitting:
    - Large intermediate layer
    - Many iterations during training
- Bayesian models:
  - Complex graph structures – many parameters

Preventing overfitting

- Reduced error pruning: use independent test set
  - But: may overfit the test set
- Statistical significance
  - But: little data in lower parts of tree
- Thresholds: minimal number of data
- Many data ….

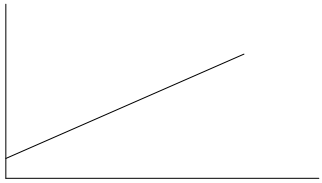
"Learning bias"

- Many candidate hypotheses are consistent with data
- Method prefers one over the rest = bias
- Bias can be:
  - Strict / preference
  - Explicit / implicit (in procedure)

Bias in decision tree learner:

- If
  - *class distibutions are hyperrectangles*, and
  - *Pruning* works correctly, and
  - *Myopic search* makes good choices
- Then TDIDT will work optimally

When does it not work (well)?



When does it not work (well)?

E.g. Parity function
- N binary attributes (1 / 0)
- "even" number of "1"s --> positive
- "odd" number of "1"s --> negative

1	2	3	4	class
1	1	1	1	pos
1	1	1	0	neg
1	1	0	1	neg

Problem: no single predictive attribute!?

Many more methods

- Bayesian-1: estimate P(hypothesis | data) and take the (1) hypothesis with the maximum
- Bayesian-2: estimate P(target | training data, new example) (more complex and expensive)
- Neural networks: go Bayesian or biological
- Support Vector Machines / kernel methods: work from (function of) similarity between data; good for high-dimensional problems

Examples of Bias

- (fixed or initial) topology of neural and Bayesian networks
- *Linear* prediction functions
- Decision tree learning?

- Often: bias for simple models:
  - Decision trees
  - Neural networks
- Why? Are simpler models better on average?

Bias and overfitting

- *Strong* bias → restricted hypothesis set, less overfitting
- But bias may be false …
- And vv.
- Example: linear / polynomial, small / large tree or neural net

Different learning tasks

- Offline / batch
- Incremental
- Dynamic domain / concept drift
- Vector / sequence
- Supervised / unsupervised (e.g. clustering)
- Data types: numbers, nominal (member of set)

Clustering

- Unsupervised
- Data
- Goal is not predict one variable from others but group data into “clusters” that are similar
- Note: cluster can be used for prediction:
  1. New object: find out which cluster it matches
  2. predict most frequent values
- No “target” predicted variable

Clustering methods

- Similarity/distance measure
- Heuristic algorithm
  - Fixed number of clusters
  - Fixed minimal similarity for clusters
  - "Iterative improvement"

Examples

- Many, many
- Customers
- Stars
- Ships
- ….

Why does ML work?

- What does "work" mean?
  - Find correct hypothesis?
  - Find "best" hypothesis given data? What is the "best hypothesis given the data"?
  - ??
- Does "decision tree learning" always work? No
- What makes it fail?
  - Lack of data
  - Data not representative
  - Overfitting (or: underfitting)
  - Mismatch bias / domain

No Free Lunch

- Hypothesis / prediction is explicitly or implicitly based on assumption
- Kind of heuristic: may be false
- specifically, suppose we look at all possible worlds
- Data → hypothesis → (+ new example) prediction
- In 50% of possible worlds this prediction will be false
- This holds for ***all*** learning methods; no method is on average better than ***any*** other method

Learning as reasoning

- ML:
  - (many) data (+ bias) → hypothesis
  - Hypothesis + new example → prediction for unknown feature of new example
- also possible:
  - Direct from data to prediction: "instance –based" or "lazy"; match new data with old

From a logical perspective

- Inductive vs. Deductive: Hypothesis and predictions are not deduced from data; may be false if data are true
- Logical models of learning:
  - ILP: hypothesis ➔ data (instead of vv) (like "abduction"); find hypothesis
    - Eg. Decision tree + new data →class of new data
  - Rational reconstruction: data + bias ➔ candidate hypotheses
    - Training data + "bias" ➔ decision tree

From a statistical perspective

- ML = estimation; compare estimating mean from sample
- E.g. decision tree estimates distribution of classes over data

From a mathematical perspective

- Function fitting / approximation
- Hypothesis ~ function

Main points

- Decision tree learning as an example of ML
- Metamodel: training data, hypothesis space, “learning bias”
- A few more methods at a glance
- Overfitting
- Bias
- Why does it work? No Free Lunch!?
- ML as inference