The Power of Decision Tables

Ronny Kohavi

25 April 1995

Talk Outline

① Introduction & Motivation

② Decision Tables.

③ Inducing decision tables.

④ Experimental results & recent experiments.

⑤ Summary.

Supervised Classification Learning

The task of the *induction algorithm* (learning algorithm), is to build a classifier from a dataset, such that the classifier has high prediction accuracy.

Input A dataset (training set) containing labelled instances i.i.d. over the labelled instance space.

Output A classifier mapping an unlabelled instance to a label.

Classifier's prediction accuracy The probability of correctly classifying a randomly selected instance.

The Classifier

1. Classifiers commonly use a "structure" to describe the hypothesized concept (decision trees, graphs, rules).
2. The structure determines what target concepts will be easy to represent (easy=small structure).
3. Most Boolean concepts will not have a polynomial-size representation in any reasonable structure (the Shannon effect).
4. Because learners usually attempt to find small structures (Occam's razor), this defines an inherent bias.

Which structures are appropriate for real-world problems?

Motivation

1. Feature subset selection work (ML94) showed many trees were complete.

2. Little use of the power of the decision tree.

3. Could it be that decision tables are a good representation?

4. Is the generalization power coming from feature selection?

Decision Tables

1. A **decision table** has two parts: a schema and a body.
2. The **schema** is a set of feature names.
3. The **body** is simply a list of instances, each one containing the features values for the features in the schema.
4. To classify, match the schema features. If not found, simply predict the **default class.**

physician-fee-freeze	mx-missile	export-admin-to-South-Africa	label
y	y	y	republican
y	n	y	republican
n	n	y	democrat
y	n	n	republican
n	y	y	democrat
n	n	u	democrat

Default: democrat

The Free Parameters

The free parameters in inducing a decision table:

1. The default class: the majority class in the training set.
2. The instances: all combinations found in the training set.
3. The schema features: feature subset selection.

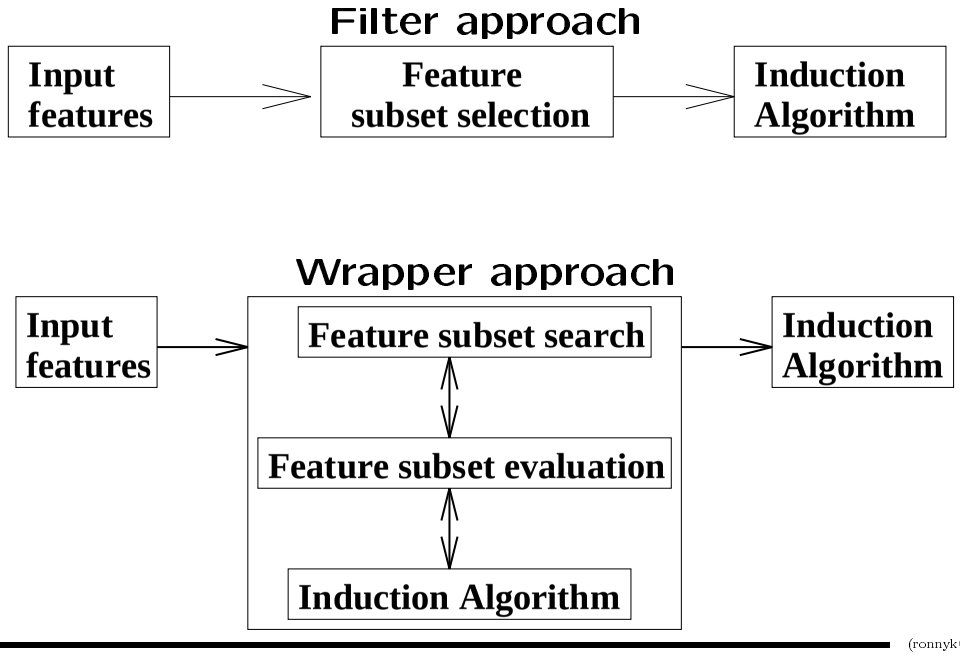
A set of **optimal features,** S^***,** consists of the features such that if the training set is projected on S^*, we get the decision table with the highest possible accuracy.

Problems: not operational, space of 2^n features is too large.

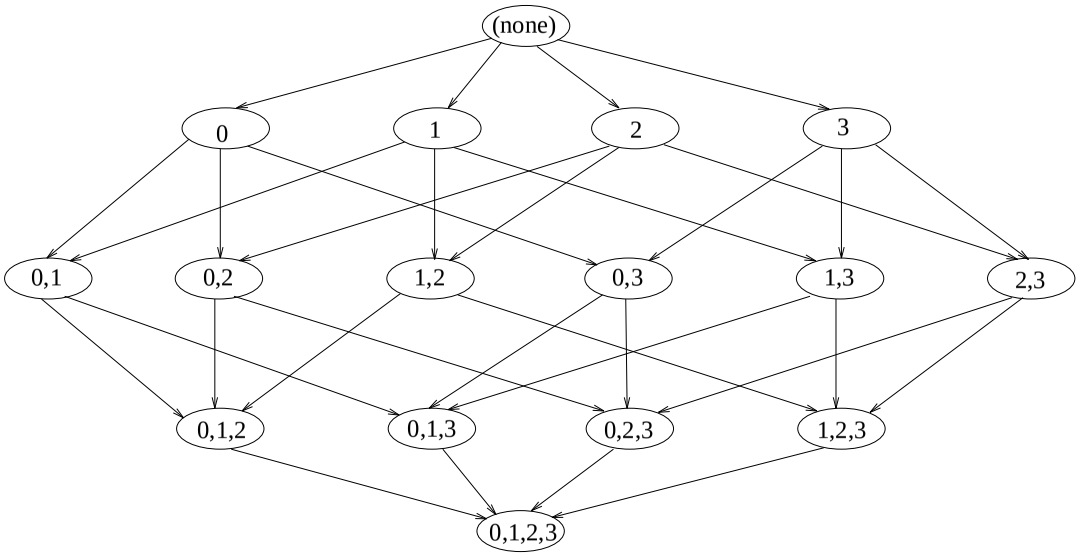
A minimal S^* does not contain irrelevant features and it may not even contain relevant ones.

Feature Subset Selection: The Wrapper Approach

We use the *wrapper approach* described by John, Kohavi, and Pfleger in ML-94.



The Search Space



The Wrapper Approach Instantiation

The following options chosen:

1. The feature subsets define a state space where *add feature* and *delete feature* are the operators.
2. A **best-first search** is conducted in the state space.
3. The heuristic function is the estimated prediction accuracy, determined by cross-validation.
4. The search starts with the empty set of features and terminates if 10 consecutive node expansions do not yield any improvement.

Incremental Cross-Validation

1. Given a schema, decision tables can be updated incrementally by adding and deleting instances from the table.
2. Incremental induction algorithms are well suited for cross-validation. Instead of running the induction algorithm k times, we *delete* the fold instances, classify them, put them back.
3. The time to cross-validate a decision table (with a given schema) and a dataset is linear in the number of instances and independent of k, the number of folds.
4. Since 10-fold CV takes as much time as leave-one-out, should we do leave-one-out?

Experimental Results

Continuous domain is one with at least one continuous feature.

Domains	Insignificant Difference	C4.5 better	tables better
Discrete	7	4	5
Continuous	9	12	1

In splice-junction DNA, the accuracy was 94.6%±0.7% for tables, compared with 92.3%±0.8% for C4.5. The table's schema contained 11 bits (out of 180).

Discretization must be done to make decision tables competitive in continuous domains.

Discretization

1. A comparison of discretization methods for C4.5 and Naive-Bayes will appear in ML-95 [Dougherty, Kohavi, and Sahami].
2. All methods were approximately the same, but the method proposed by Fayyad & Irani in IJCAI-93 was the best.

New results (not in paper)

Domains	Insignificant Difference	C4.5 better	tables better
Discrete	7	4	5
Continuous	8	6	7

Summary

1. Simple decision tables can be appropriate for many real-world datasets, especially if discretized.
2. In some cases the size of the table makes it easy to comprehend.
3. Feature subset selection is critical for real-world problems. A lot of the inductive power comes from knowing what to leave out.
4. The current FSS algorithm is slow, but incremental cross-validation makes it more practical.
5. After discretization, this is a truly competitive algorithm.
6. Run time is slower to select schema, but updating is much faster.