Introduction to variable selection

Part 2: Wrappers for feature subset selection

Mikko Korpela

26th September, 2006

Outline of presentation

- Introduction
- Feature subset selection
  - Relevance and optimality of features
  - Filter approach, wrapper approach
- Experimental setting
- Search engines for wrapper approach
- Compound operators in state space
- Comparative results
- Overfitting
- Summary

Introduction

- In the article [1], feature selection used for classification
  - Most results probably applicable to regression
- Goal: Select feature subset that maximizes classification performance on an unseen test set
  - Different from choosing the relevant set of features!

[1] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2):273–324, 1997.

Feature subset selection

- Many machine algorithms degrade in performance when irrelevant features are present
- Correlated, relevant features may also be harmful
- Feature subset selection, a definition:
  - Find a subset of features that maximizes accuracy of classifier
  - No feature extraction or construction
- Feature selection not needed for Bayes classifier
- Practical algorithms face the bias–variance tradeoff

Feature subset selection (2)

- **Optimal feature subset** defined with respect to the induction algorithm

Definition 1
Given an inducer $\mathcal{I}$, and a dataset $\mathcal{D}$ with features $X_1, X_2, \ldots, X_n$, from a distribution D over the labeled instance space, an **optimal feature subset, $\mathbf{X_{opt}}$,** *is a subset of the features such that the accuracy of the induced classifier $\mathcal{C} = \mathcal{I}(\mathcal{D})$ is maximal.*

- Optimal feature subset not necessarily unique
- Problem: Distribution of data not known
  - ➡ Accuracy of classifier must be estimated from data

Relevance of features

- Many definitions of relevance suggested:

Definition 2
A feature X_i is said to be **relevant** *to a concept $\mathcal{C}$ if X_i appears in every Boolean formula that represents $\mathcal{C}$ and* **irrelevant** *otherwise.*

Definition 3
X_i *is* **relevant** *iff there exists some x_i and y for which $p(X_i = x_i) > 0$ such that*

$$p(Y = y \mid X_i = x_i) \neq p(Y = y) \ .$$

Definition 4
X_i *is* **relevant** *iff there exists some x_i, y, and s_i for which $p(X_i = x_i) > 0$ such that*

$$p(Y = y, S_i = s_i \mid X_i = x_i) \neq p(Y = y, S_i = s_i) \ .$$

Definition 5
X_i *is* **relevant** *iff there exists some x_i, y, and s_i for which $p(X_i = x_i, S_i = s_i) > 0$ such that*

$$p(Y = y \mid X_i = x_i, S_i = s_i) \neq p(Y = y \mid S_i = s_i) \ .$$

Relevance of features (2)

- Let's see how different definitions of relevance do in practice:

Example 1 (Correlated XOR) Let features $X_1, \ldots, X_5$ be Boolean. The instance space is such that X_2 and X_3 are negations of X_4 and X_5, respectively, *i.e.*, $X_4 = \overline{X_2}$, $X_5 = \overline{X_3}$. There are only eight possible instances, and we assume they are equiprobable. The (deterministic) target concept is

$$Y = X_1 \oplus X_2 \qquad (\oplus \text{ denotes XOR}) \ .$$

- Intuitively, X_1 is relevant
- Both X_2 and X_4 are relevant, but either one can be omitted
- Definitions of relevance fail miserably:

Definition	Relevant	Irrelevant
Definition 2	X_1	X_2, X_3, X_4, X_5
Definition 3	None	All
Definition 4	All	None
Definition 5	X_1	X_2, X_3, X_4, X_5

Relevance of features (3)

- Better definitions of relevance needed
- **Strong relevance** and **weak relevance** defined in terms of a Bayes classifier:

Definition 5 (Strong relevance)
A feature X_i *is* **strongly relevant** *iff there exists some* x_i, y, *and* s_i *for which* $p(X_i = x_i, S_i = s_i) > 0$ *such that*

$$p(Y = y \mid X_i = x_i, S_i = s_i) \neq p(Y = y \mid S_i = s_i) \ .$$

Definition 6 (Weak relevance)
A feature X_i *is* **weakly relevant** *iff it is not strongly relevant, and there exists a subset of features* S'_i *of* S_i *for which there exists some* x_i, y, *and* s'_i *with* $p(X_i = x_i, S'_i = s'_i) > 0$ *such that*

$$p(Y = y \mid X_i = x_i, S'_i = s'_i) \neq p(Y = y \mid S'_i = s'_i) \ .$$

- In Example 1, X_1 strongly relevant, X_2 and X_4 weakly relevant, X_3 and X_5 **irrelevant**

Relevance and optimality

- Bayes classifier uses:
  - All strongly relevant features
  - Possibly some weakly relevant features
- For practical classifiers:
  - Relevance does not imply membership in optimal feature subset
  - Irrelevance does not imply that a feature should not be in optimal feature subset
  - Examples available, omitted here...

Filter approach

Figure 2: The feature filter approach, in which the features are filtered independently of the induction algorithm.

- Feature selection done as a preprocessing step
- Drawback: Effect of feature selection on induction algorithm not known
- Algorithm called **Relieved-F** (based on **Relief**) used in comparisons
    - Attempts to find all relevant features

Wrapper approach

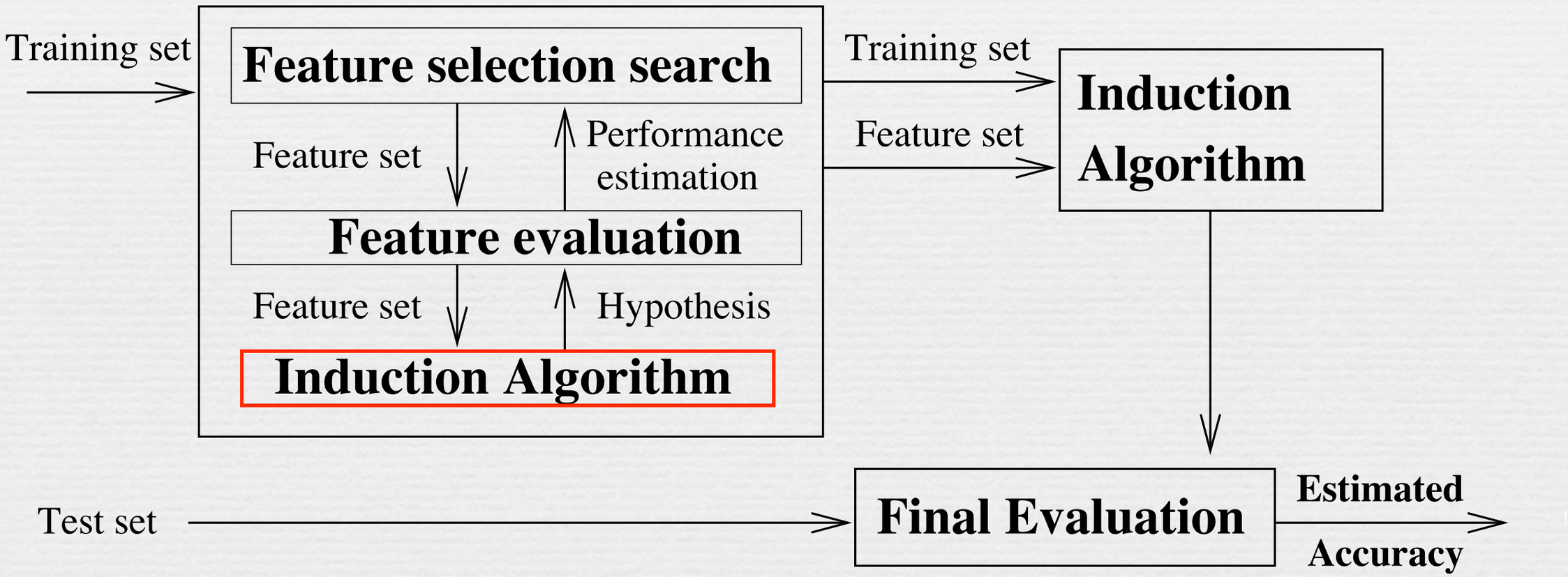


Figure 1: The wrapper approach to feature subset selection. The induction algorithm is used as a "black box" by the subset selection algorithm.

Wrapper approach (2): state space search

- Each state represents a feature subsets
- State is boolean vector with
  1 = feature present, 0 = feature absent
- Wrapper method searches state space trying to find best features
- "Black box" induction algorithm evaluates states

State	A Boolean vector, one bit per feature
Initial state	The empty set of features (0,0,0..., 0)
Heuristic/evaluation	Five-fold cross-validation repeated multiple times with a small penalty (0.1%) for every feature.
Search algorithm	Hill-climbing or best-first search
Termination condition	Algorithm dependent (see below)

Wrapper approach (3): connectedness of states

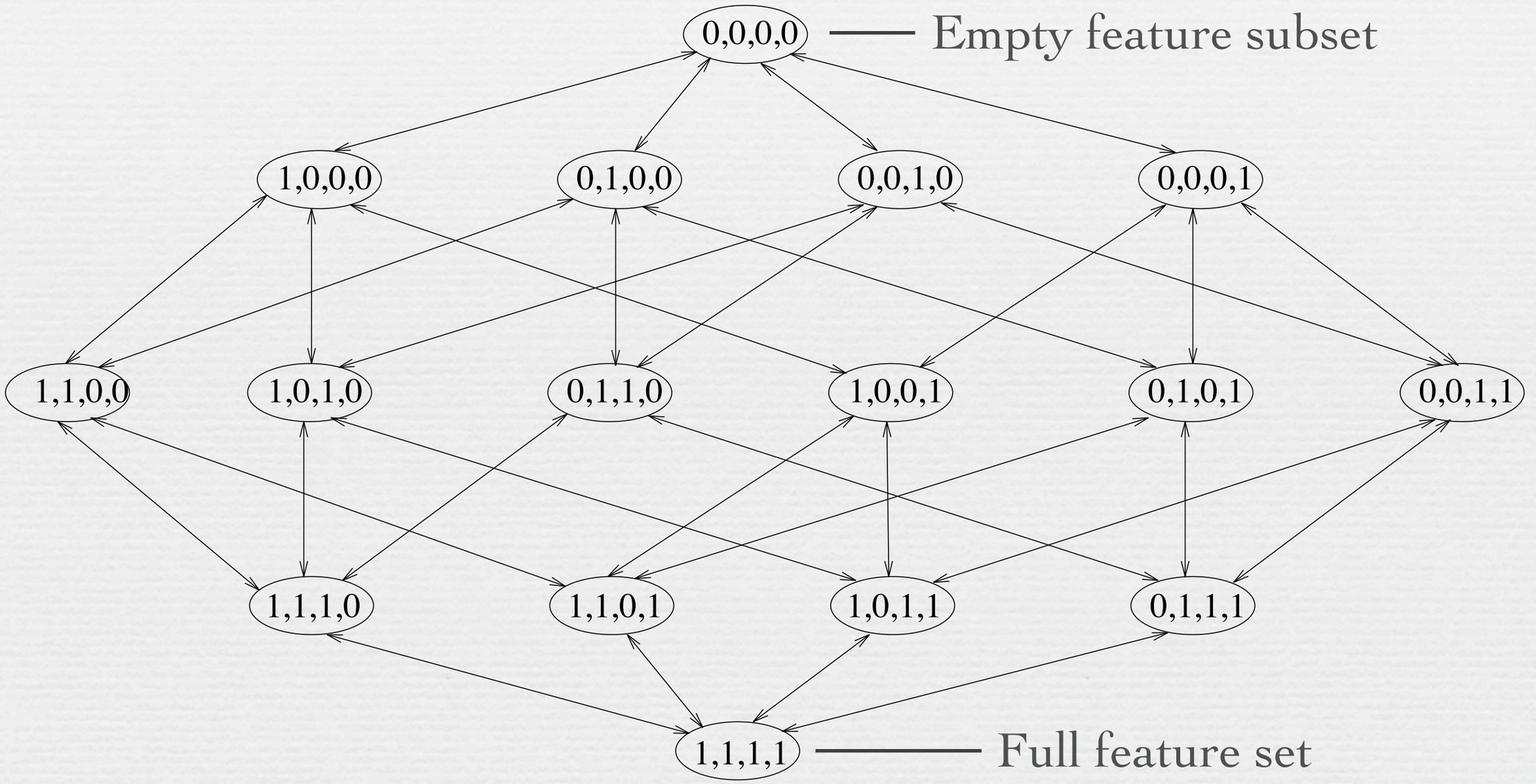


Figure 5: The state space search for feature subset selection. Each node is connected to nodes that have one feature deleted or added.

Experimental setting:

Datasets

Table 2: Summary of datasets. Datasets above the horizontal line are "real" and those below are artificial. CV indicates ten-fold cross-validation.

no.	Dataset	Features			no.	Train	Test	baseline
		all	nominal	continuous	classes	size	size	accuracy *
1	breast cancer	10	0	10	2	699	CV	65.52
2	cleve	13	7	6	2	303	CV	54.46
3	crx	15	9	6	2	690	CV	55.51
4	DNA	180	180	0	3	2000	1186	51.91
5	horse-colic	22	15	7	2	368	CV	63.04
6	Pima	8	0	8	2	768	CV	65.10
7	sick-euthyroid	25	18	7	2	2108	1055	90.74
8	soybean-large	35	35	0	19	683	CV	13.47
9	Corral	6	6	0	2	32	128	56.25
10	*m*-of-*n*-3-7-10	10	10	0	2	300	1024	77.34
11	Monk1	6	6	0	2	124	432	50.00
12	Monk2-local	17	17	0	2	169	432	67.13
13	Monk2	6	6	0	2	169	432	67.13
14	Monk3	6	6	0	2	122	432	52.78

* Accuracy when simply predicting the majority class

Experimental setting:
Induction algorithms

- Two families of induction algorithms used in the paper
  - (Induction algorithms build classifiers)

1. Decision tree algorithms
   - **C4.5**, builds trees top-down and prunes them
   - **ID3**, no pruning
2. Naive-Bayes

$$
\begin{aligned}
& p(Y = y \mid \vec{X} = \vec{x}) \\
& = p(\vec{X} = \vec{x} \mid Y = y) \cdot p(Y = y) / p(\vec{X} = \vec{x}) && \text{by Bayes rule} \\
& \propto p(X_1 = x_1, \ldots, X_n = x_n \mid Y = y) \cdot p(Y = y) && p(\vec{X} = \vec{x}) \text{ is same for all label values.} \\
& = \prod_{i=1}^{n} p(X_i = x_i \mid Y = y) \cdot p(Y = y) && \text{by independence}
\end{aligned}
$$

"Naive"

Search engines for wrapper approach: Hill-climbing search

- The simplest search technique
- Also called "greedy search" or "steepest ascent"
- Move to child with highest accuracy, terminate when no improvement

Table 3: A hill-climbing search algorithm

1. Let $v \leftarrow$ initial state.
2. Expand v: apply all operators to v, giving v's children.
3. Apply the evaluation function f to each child w of v.
4. Let $v' =$ the child w with highest evaluation $f(w)$.
5. If $f(v') > f(v)$ then $v \leftarrow v'$; goto 2.
6. Return v.

Search engines for wrapper approach: Best-first search

- More robust than hill climbing
- Select most promising node generated so far that hasn't been expanded

Table 6: The best-first search algorithm

1. Put the initial state on the OPEN list,
   CLOSED list $\leftarrow \emptyset$, BEST $\leftarrow$ initial state.
2. Let $v = \underset{w \in \text{OPEN}}{\arg\max}\ f(w)$ (get the state from OPEN with maximal $f(w)$).
3. Remove v from OPEN, add v to CLOSED.
4. If $f(v) - \epsilon > f(\text{BEST})$, then BEST $\leftarrow v$.
5. Expand v: apply all operators to v, giving v's children.
6. For each child not in the CLOSED or OPEN list, evaluate and add to the OPEN list.
7. If BEST changed in the last k expansions, goto 2.
8. Return BEST.

"Stale search"

Compound operators in state space

- Topology of search space previously defined by addition or deletion of a single feature at a time
  - Search can be quite slow
- Compound operators combine several additions or deletions into one operation
  - Dynamically created after standard set of children (single additions and deletions) evaluated
  - Search can advance faster
  - Backward feature selection search **now feasible**

Compound operators in state space: Example

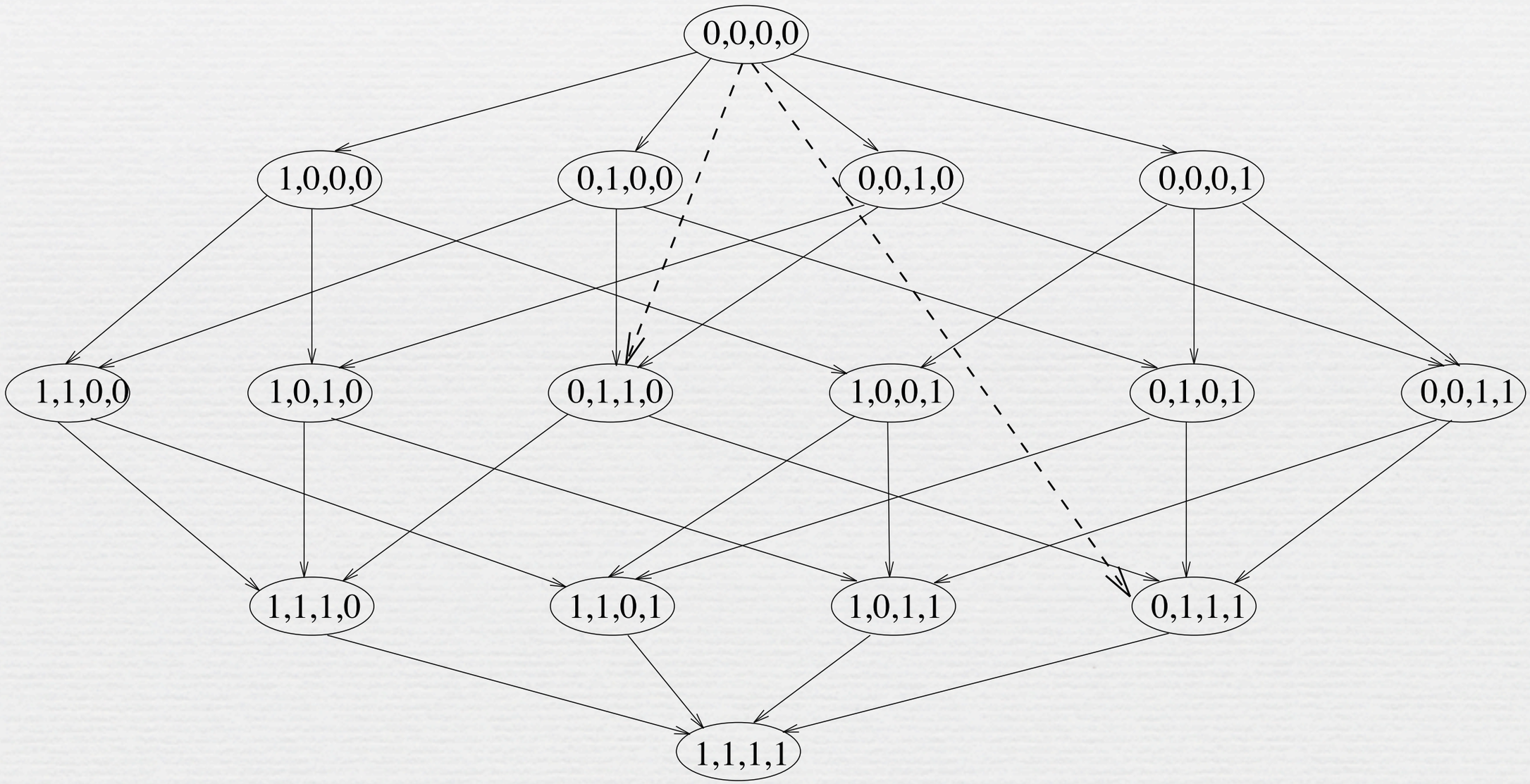
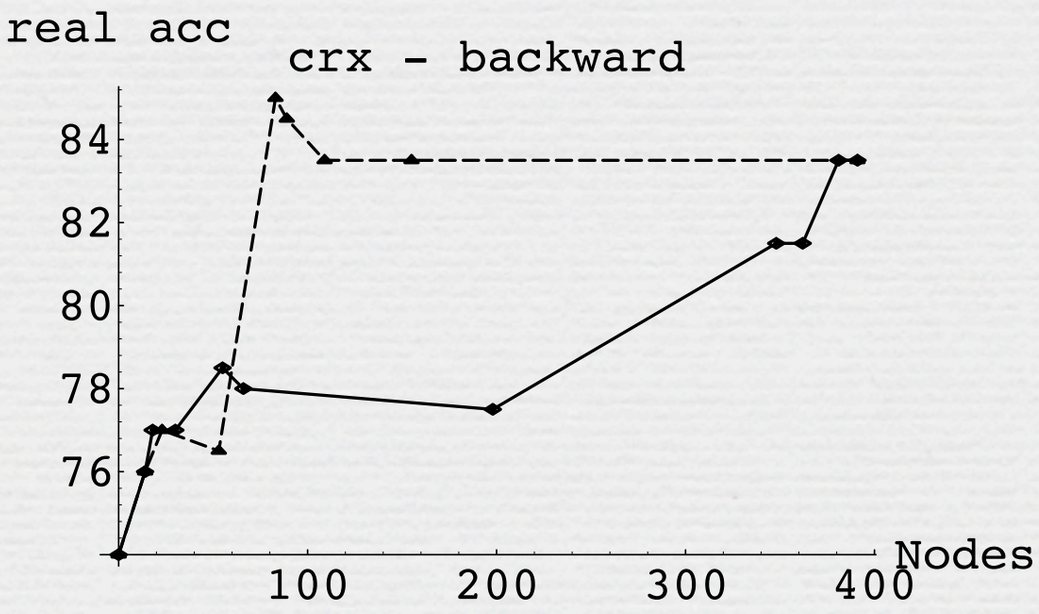


Figure 14: The state space search with dotted arrows indicating compound operators. From the root's children, the nodes (0,1,0,0) and (0,0,1,0) had the highest evaluation values, followed by (0,0,0,1).

Compound operators in state space: Results

- Big improvement in backward search
- Nodes with good accuracy found faster
- Overfitting also faster



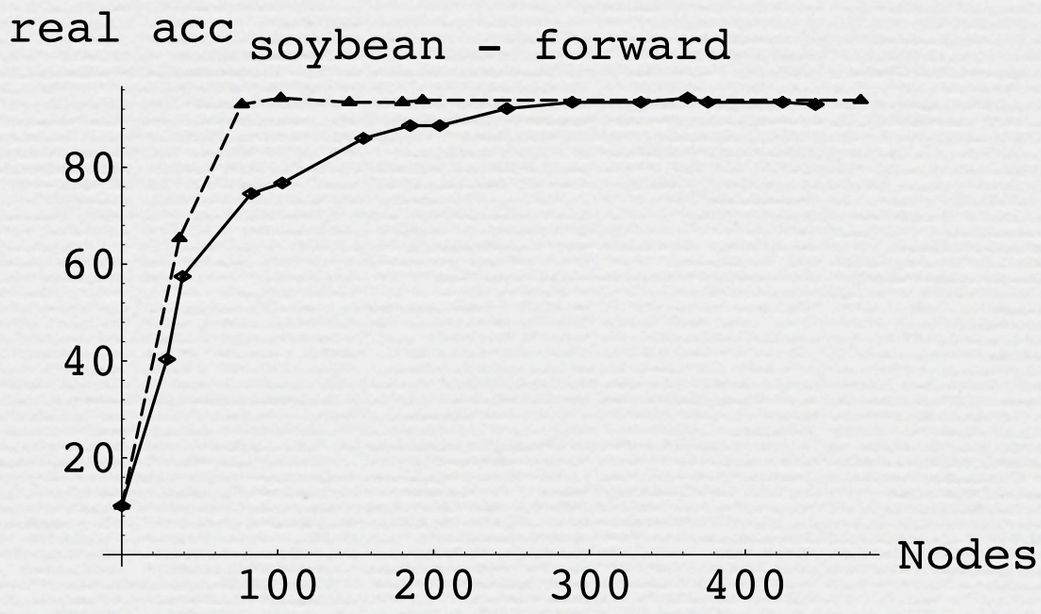


Figure 15: Comparison of compound (dotted line) and non-compound (solid line) searches. The accuracy (y-axis) is that of the best node (as determined by the algorithm) on an independent test set after a given number of node evaluations (x-axis). The running time is proportional to the number of nodes evaluated.

Comparative results

- Filter approach fairly erratic, sometimes degrades classification performance
- Wrapper approach more consistent, usually improves performance
- Best-first search generally better than hill climbing
  - Especially with ID3 induction algorithm
- Backward best-first search with compound operators reduces number of features by 19–40 % on the average, depending on induction algorithm
- More detailed results here, and in the paper

Overfitting

- Definition:
  - Training data modeled too well
  - Predictions poor
- Search engine guided by accuracy estimates
- Estimates can be poor, misleading
- Mainly a problem when number of instances small

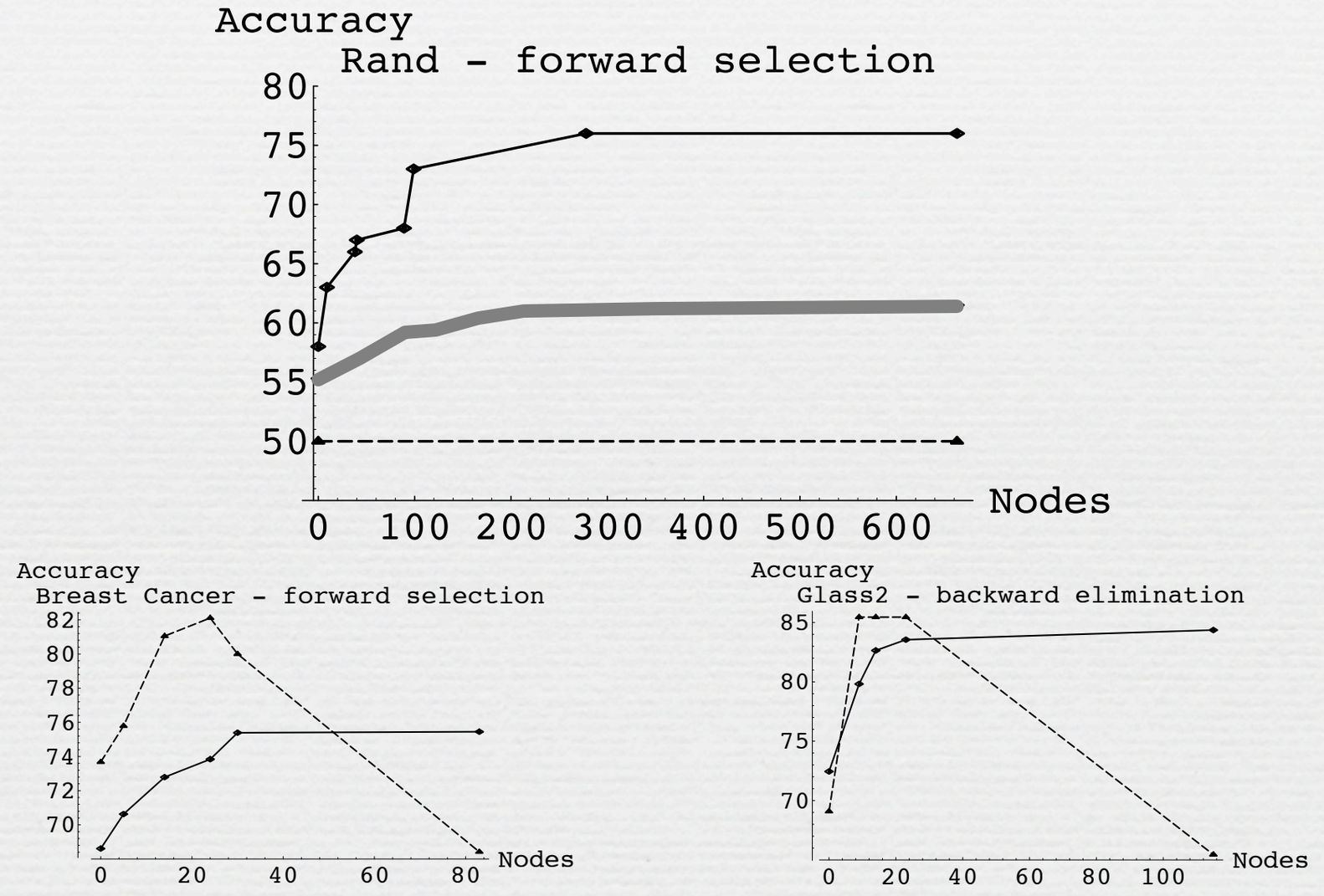


Figure 20: Overfitting in feature subset selection. The top graph shows the estimated and true accuracies for a random dataset and ID3. The solid line represents the estimated accuracy for a training set of 100 instances, the thick grey line for a training set of 500 instances, and the dotted line shows the real accuracy. The bottom graphs graphs show the accuracy for real-world datasets. The solid line is the estimated accuracy, and the dotted line is the accuracy on an independent test set.

Summary

- Feature subset selection reviewed
- Relevance of a feature, definitions
  - Optimality for given task more important
- Wrapper approach
  - Search space
  - Operators
  - Search engine
  - Evaluation function
- On average, classification performance improved with feature subset selection
- Problems: overfitting, CPU time

* Relevance and optimality: Examples

Example 2 (Relevance does not imply optimality) Let the universe of possible instances be $\{0,1\}^3$, that is, three Boolean features, say X_1, X_2, X_3. Let the distribution of instances be uniform, and assume the target concept is $f(X_1, X_2, X_3) = (X_1 \wedge X_2) \vee X_3$. Under any reasonable definition of relevance, all features are relevant to this target function.

If the hypothesis space is the space of monomials, *i.e.*, conjunctions of literals, the only optimal feature subset is $\{X_3\}$. The accuracy of the monomial X_3 is 87.5%, the highest accuracy achievable within this hypothesis space. Adding another feature to the monomial will decrease the accuracy. ∎

Example 3 (Optimality does not imply relevance) Assume there exists a feature that always takes the value one. Under all the definitions of relevance described above, this feature is irrelevant. Now consider a limited Perceptron classifier (Rosenblatt 1958, Minsky & Papert 1988) that has an associated weight with each feature and then classifies instances based upon whether the linear combination is greater than zero (the threshold is fixed at zero). (Contrast this with a regular Perceptron that classifies instances depending on whether the linear combination is greater than some threshold, not necessarily zero.) Given this extra feature that is always set to one, the limited Perceptron is equivalent in representation power to the regular Perceptron. However, removal of all irrelevant features would remove that crucial feature.

❋ Classification results

Table 16: A comparison of C4.5 with ID3-FSS, C4.5-FSS, and Naive-Bayes-FSS. The p-val columns indicates the probability that the column before it is improving over C4.5

Dataset	C4.5 original	ID3-FSS Frwd-BFS	p-val	C4.5-FSS Back-BFS	p-val	NB-FSS Back-BFS	p-val
breast cancer	95.42± 0.7	94.57± 0.7	0.11	95.28± 0.6	0.41	96.00± 0.6	0.81
cleve	72.30± 2.2	79.52± 2.3	**1.00**	77.88± 3.2	**0.98**	82.56± 2.5	**1.00**
crx	85.94± 1.4	85.22± 1.6	0.31	85.80± 1.3	0.46	84.78± 0.8	0.15
DNA	92.66± 0.8	94.27± 0.7	**0.99**	94.44± 0.7	**0.99**	96.12± 0.6	**1.00**
horse-colic	85.05± 1.2	82.07± 1.5	**0.01**	84.77± 1.3	0.41	82.33± 1.3	**0.01**
Pima	71.60± 1.9	68.73± 2.2	0.08	70.18± 1.3	0.20	76.03± 1.6	**0.99**
sick-euthyroid	97.73± 0.5	97.06± 0.5	0.09	97.91± 0.4	0.66	97.35± 0.5	0.21
soybean-large	91.35± 1.6	91.65± 1.0	0.59	91.93± 1.3	0.65	94.29± 0.9	**0.99**
Corral	81.25± 3.5	100.00± 0.0	**1.00**	81.25± 3.5	0.50	90.62± 2.6	**1.00**
m-of-*n*-3-7-10	85.55± 1.1	77.34± 1.3	**0.00**	85.16± 1.1	0.36	87.50± 1.0	**0.97**
Monk1	75.69± 2.1	97.22± 0.8	**1.00**	88.89± 1.5	**1.00**	72.22± 2.2	**0.05**
Monk2-local	70.37± 2.2	95.60± 1.0	**1.00**	88.43± 1.5	**1.00**	67.13± 2.3	0.07
Monk2	65.05± 2.3	63.89± 2.3	0.31	67.13± 2.3	0.82	67.13± 2.3	0.82
Monk3	97.22± 0.8	97.22± 0.8	0.50	97.22± 0.8	0.50	97.22± 0.8	0.50
Average real:	86.51	86.64		87.27		88.68	
Average artif.	79.19	88.55		84.68		80.30	