Introduction to Machine Learning

Zdravko Markov
Faculty of Mathematics and Informatics,
University of Sofia
E-mail: markov@fmi.uni-sofia.bg
http://www-it.fmi.uni-sofia.bg/markov

May 20, 1998

Course outline

- An example
- ML subfields and related areas
- Inductive learning – languages and posets
  - General setting for induction
  - Propositional (attribute-value) languages
  - Inductive Logic Programming
- Searching the hypothesis space
  - Divide-and-conquer strategies
  - Covering strategies
  - Error estimation, Bayes, Occam

Example

No.	x_1	x_2	x_3	x_4	x_5	x_6	y
1	0	0	0	2	3	1	1
2	2	1	1	1	0	0	0
3	2	1	1	1	3	0	1
4	1	0	0	0	0	1	0
5	2	2	1	2	2	1	1
6	1	2	1	0	2	1	0
7	1	1	0	2	2	1	1
8	0	1	0	1	2	0	0
9	0	0	1	0	0	0	1
10	2	0	1	0	0	0	0
11	1	1	0	2	1	1	?

$X = (x_1, x_2, x_3, x_4, x_5, x_6)$

Instance space $= |\{X\}| = 432$

Concept learning/regression: f?, $y = f(X)$

Prediction: $f(1, 1, 0, 2, 1, 1) = $?

$Error(f) = \frac{|\{X|f(X) \neq y\}|}{432}$

Concept learning

Propositinal Rules

$x_1 = 0 \wedge x_2 = 0 \rightarrow y = 1$

$x_1 = 1 \wedge x_2 = 1 \rightarrow y = 1$

$x_1 = 2 \wedge x_2 = 2 \rightarrow y = 1$

$x_5 = 3 \rightarrow y = 1$

$x_1 = 0 \wedge x_2 = 1 \wedge x_5 = 0 \rightarrow y = 0$

$x_1 = 0 \wedge x_2 = 2 \wedge x_5 = 0 \rightarrow y = 0$

$x_1 = 0 \wedge x_2 = 1 \wedge x_5 = 1 \rightarrow y = 0$

$x_1 = 0 \wedge x_2 = 2 \wedge x_5 = 1 \rightarrow y = 0$

...

Decision table

if $x_1 = 0 \wedge x_2 = 0$ then $y = 1$

else if $x_1 = 1 \wedge x_2 = 1$ then $y = 1$

else if $x_1 = 2 \wedge x_2 = 2$ then $y = 1$

else if $x_5 = 3$ then $y = 1$

else $y = 0$

Relational representation

(variables, relations: $=$, $\neq$)
$x_1 = x_2 \vee x_5 = 3 \rightarrow y = 1$
$x_1 \neq x_2 \wedge x_5 \neq 3 \rightarrow y = 0$

Lazy learning

K-Nearest Neighbour (k-NN)

$$D(X,Y)=\sum_{i=1}^{6} d(X_i,Y_i), \qquad d(x,y)=\begin{cases} 0, & x=y \\ 1, & x\neq y \end{cases}$$

Y	1	2	3	4	5	6	7	8	9	10
$D(X,Y)$	3	5	5	3	4	4	1	4	6	6
y	1	0	1	0	1	0	1	0	1	0

$D(X,Y)$	1	3	3	4	4	4	5	5	6	6
y	1	1	0	1	0	0	0	1	1	0

K	1	2	3	4	5	6	7
1/0	1/0	2/0	2/1	3/1	3/2	3/3	3/4
$y(X)$	1	1	1	1	1	?	0

Similarity of first order terms

$size(A) = \#\ symbols - \#\ different\ variables$

Example: $size(p(X,Y))=1$, $size(p(X,X))=2$

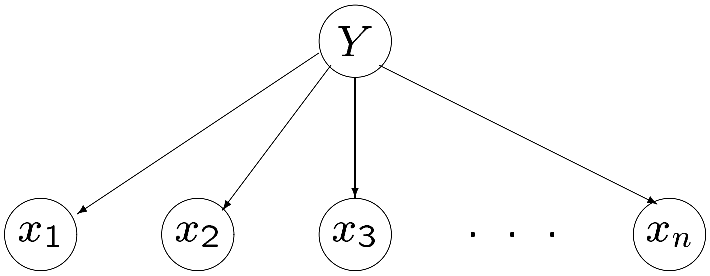
$A \geq B \Leftrightarrow size(A) \leq size(B)$

$D(A,B) = size(A) + size(B) - 2 * size(lgg(A,B))$

Example:
$D(p(a,a),p(b,b)) = size(p(a,a)) + size(p(b,b)) - 2 * size(p(X,X)) = 3+3-2*2 = 2$

Simple Bayes

Probabilistic Networks



Each node contains a probabilty table

- Node Y: $P(Y = y)$
- Node x_i: $P(x_i = v|Y = y)$

Given a new example X, we compute $P(Y = y|X)$ and choose the class y with the highest probability.

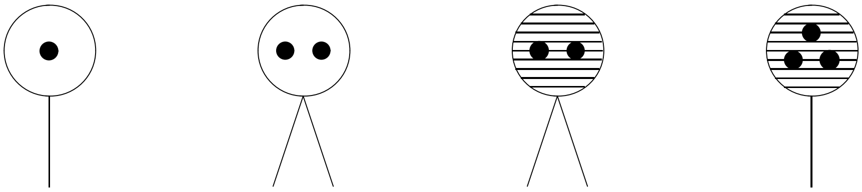
$$P(Y = y|X) = \frac{P(X|Y=y)P(Y=y)}{P(X)}$$

$$P(X|Y = y) = P(x_1 = v_1|Y = y)...P(x_n = v_n|Y = y)$$

Computing the probabilities

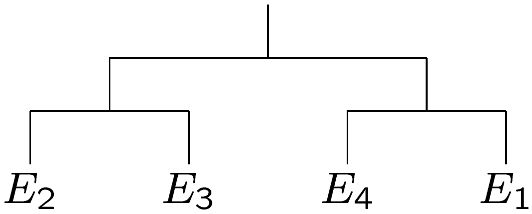
- $P(x_i = v_j|Y = y)$ is the fracion of training examples in class y where $x_i = v_j$
- $P(Y = y)$ is the fraction of training examples belonging to class y

Clustering

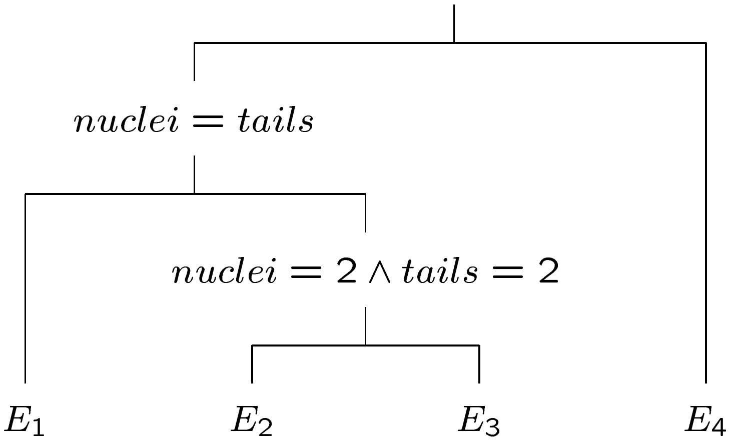


E_1	E_2	E_3	E_4
$color = light$	$color = light$	$color = dark$	$color = dark$
$nuclei = 1$	$nuclei = 2$	$nuclei = 2$	$nuclei = 3$
$tails = 1$	$tails = 2$	$tails = 2$	$tails = 1$

Agglomerative:



Conceptual:



Induction Task

$(L, L_B \subseteq L, L_E \subseteq L, L_H \subseteq L, \rightarrow)$

Given *background knowledge* $B \in L_B$, *positive examples* $E^+ \in L_E$ and *negative examples* $E^- \in L_E$, find a *hypothesis* $H \in L_H$, such that:

1. $B \not\rightarrow E^+$ (*necessity*);
2. $B \not\rightarrow E^-$ (*weak consistency*);
3. $B \cup H \rightarrow E^+$ (*sufficiency*);
4. $B \cup H \not\rightarrow E^-$ (*strong consistency*).

Hypothesis *generality/specificity*:
$H \geq H' \Leftrightarrow \{e|e \in L_E, H \rightarrow e\} \supseteq \{e|e \in L_E, H' \rightarrow e\}$.

Most general hypothesis $\top$: $e \in L_E \Rightarrow \top \rightarrow e$.

Most specific hypothesis $\perp$: $e \in E^+ \Leftrightarrow \perp \rightarrow e$.

Hypothesis space: $\{H|H \geq \perp, H \leq \top\}$

Generalization (specialization) *operators*:
$H' = \rho(H)$, $H' \geq H$ $(H' \leq H)$

Examples: gropping condition $(H_1 \geq H_2 \Leftrightarrow H_1 \subseteq H_2)$, term instantiation $(t_1 \geq t_2 \Leftrightarrow \exists\theta, t_1\theta = t_2)$.

Least general generalization (lgg): $H = lgg(H_1, H_2)$ iff $H \geq H_1$, $H \geq H_2$ and $\forall H'$, $H' \geq H_1$, $H' \geq H_2 \Rightarrow H' \geq H$.

Examples: $lgg(H_1, H_2) = H_1 \cap H_2$, most specific *antiunifier* ($lgg(p(a,a), p(b,b)) = p(X,X)$).

Searching the hypothesis space – covering strategy

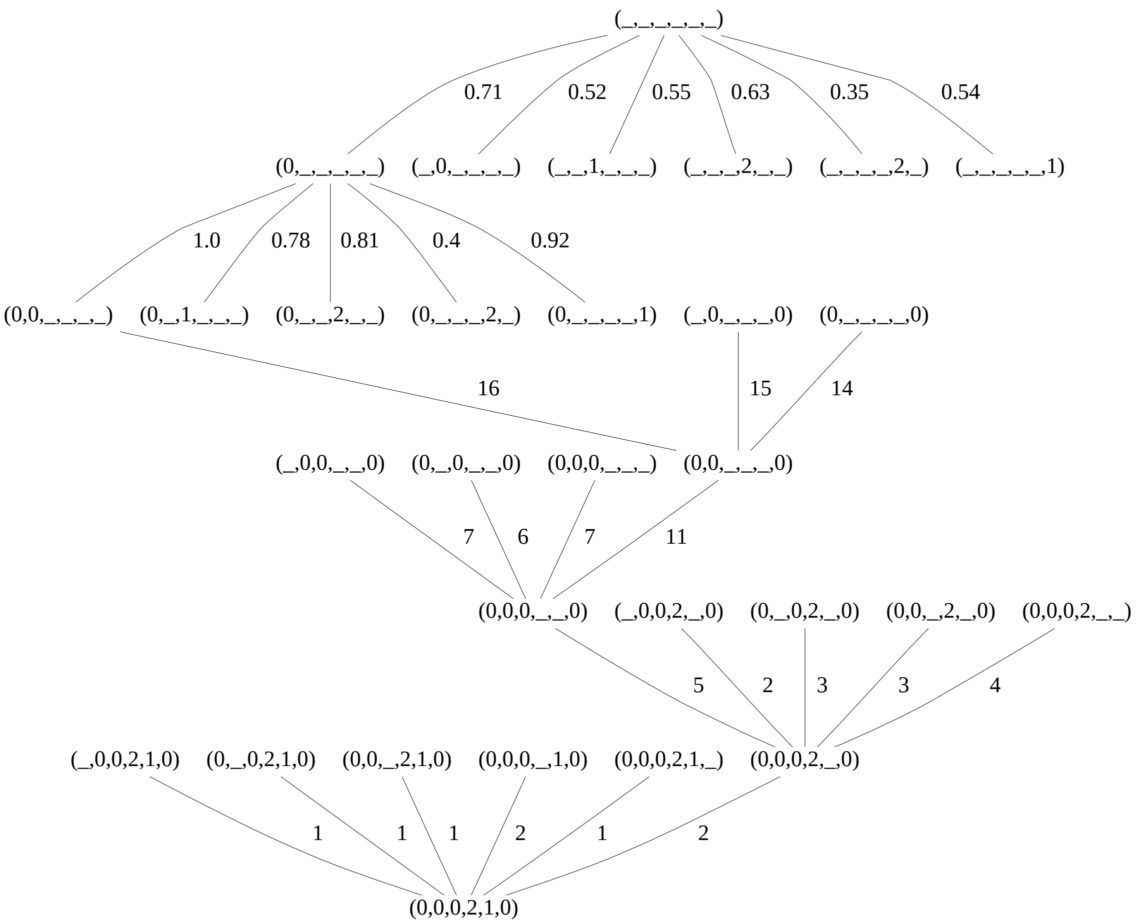
$H = H_1 \vee H_2 \vee ... \vee H_n$

1. find H_i, $\forall e \in E^- H_i \not\rightarrow e$;

2. $E^+ = E^+ \setminus \{e | H_i \rightarrow e\}$;

3. if $E^+ \neq \emptyset$ then go to 1 else stop.

Generalization/specialization graph



Lgg-based bottom-up search

Propositional language:
$E = \cup_{i=1}^{k} E_i$, $B = \emptyset$, $H = \{H_1, H_2, ..., H_k\}$, H_i satisfies the induction task for $E^+ = E_i$, $E^- = E \backslash E_i$, $B = \emptyset$.

1. choose $e_i, e_j \in E_k$ for some k;
2. $H_{ij}^k = lgg(e_i, e_j)$;
3. if $h_{ij}^k \not\rightarrow e, e \in E^l, l \neq k$ then
   $E^k = (E^K \setminus \{e | h_{ij} \rightarrow e\}) \cup \{h_{ij}\}$;
4. if step 1 or 3 are impossible then stop else go to 1.

Relational language:

θ-Subsumption: $C_1 \geq_\theta C_2 \Leftrightarrow \exists\theta\ C_1\theta \subseteq C_2$.

$lgg_\theta(C_1, C_2) = \{L | L = lgg(L_1, L_2), L_i \in C_i, i = 1, 2\}$

Relative lgg:
$C = rlgg(C_1, C_2, B) \Leftrightarrow B \cup C \models C_1, B \cup C \models C_2$

$E = E^+ \cup E^-$, B − ground atoms, H − Horn clauses.

$rlgg(e_1, e_2, B) = lgg_\theta((e_1 : -B), (e_2 : -B))$

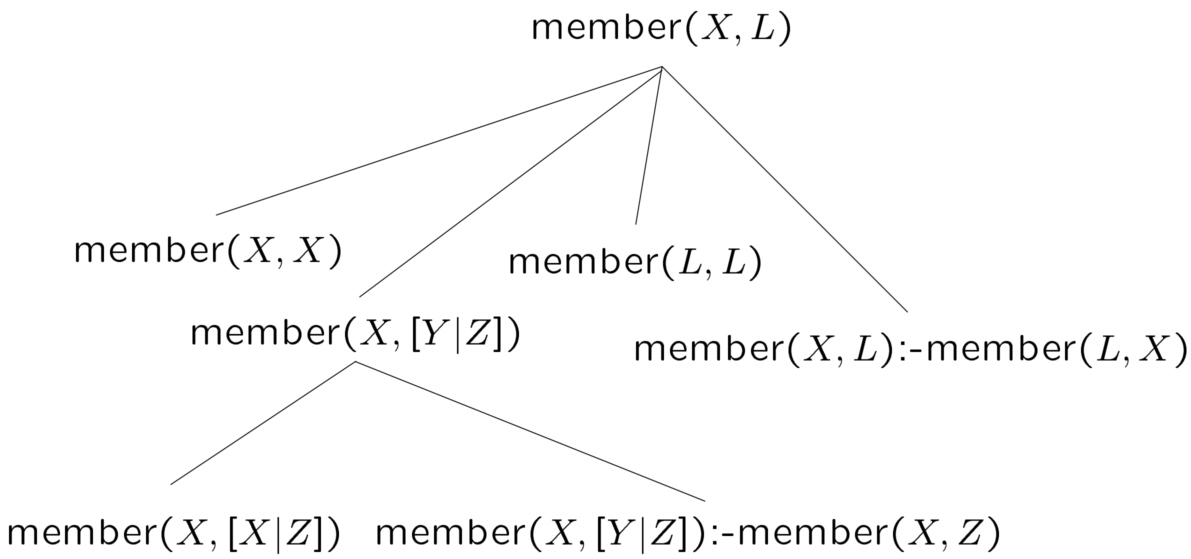
1. choose $e_i, e_j \in E^+$;
2. $C_{ij} = rlgg(e_i, e_j, B)$;
3. Reduce C_{ij}
4. $E^+ = E^+ \setminus \{e | B \cup C_{ij} \models e\}$;
5. if $E^+ = \emptyset$ then stop else go to 1.

Top-down relational learning

$E^+ = \{$member(a,[a,b]), member(b,[b]), member(b,[a,b])$\}$,
$E^- = \{$member(x,[a,b])$\}$

Specialization operatior (θ-subsumption):

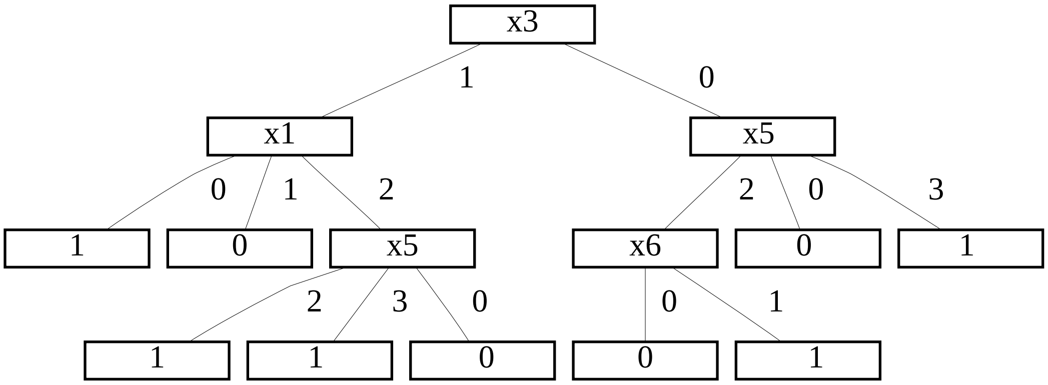
- $\rho(C) = C\theta$, $|\theta| = 1$
- $\rho(C) = C \cup \neg L$



Leaves:

- successful: $C \models e \Rightarrow e \in E^+$
- unssuccesul: $\not\exists e \in E^+, C \models e$.

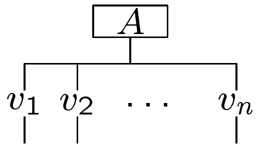
Divide-and-conquer strategy – decision trees



$x_3 = 1 \wedge x_1 = 0 \rightarrow y = 1$
$x_3 = 1 \wedge x_1 = 1 \rightarrow y = 0$
$x_3 = 1 \wedge x_1 = 2 \wedge x_5 = 2 \rightarrow y = 1$
$x_3 = 1 \wedge x_1 = 2 \wedge x_5 = 3 \rightarrow y = 1$
$x_3 = 1 \wedge x_1 = 2 \wedge x_5 = 0 \rightarrow y = 0$
$x_3 = 0 \wedge x_5 = 0 \rightarrow y = 0$
$x_3 = 0 \wedge x_5 = 3 \rightarrow y = 1$
$x_3 = 0 \wedge x_5 = 2 \wedge x_6 = 0 \rightarrow y = 0$
$x_3 = 0 \wedge x_5 = 2 \wedge x_6 = 1 \rightarrow y = 1$

TDIDT algorithm:

1. If all examples in E are of a single class then generate a leaf node labeled with this class. Otherwise *choose* an attribute A with values $v_1, v_2, ..., v_n$ and build a decision node as follows:



2. $E = \cup_{i=1}^{n} E_i$, where $E_i = \{e | e \in E, A = v_i \in e\}$
3. Call the algorithm recursively for each E_i, $i = 1, ..., n$.