Concept Approximation by Rough sets and layered learning

Hung Son Nguyen

Institute of Mathematics, Warsaw University
son@mimuw.edu.pl

February 23, 2006

Outline

Outline

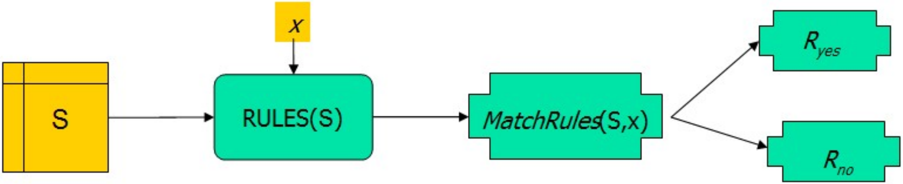
1. Concept Approximation with Layered learning
   - General idea
   - Applications

Hardness of Approximation

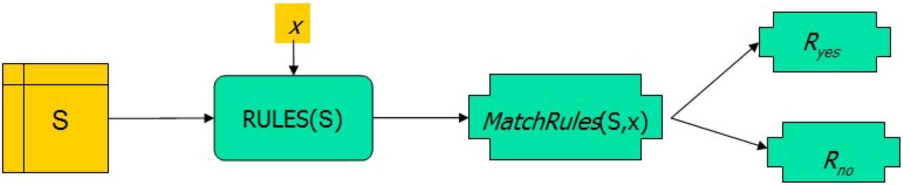
Why the concept approximation problem is hard?

- **Learnability of the target concept:** some concepts are too complex and cannot be approximated directly from feature value vectors.
  - PAC algorithms;
  - Effective learnability of some concept spaces;
  - VC dimension, ...
- **Time and space complexity:** Many problems related to optimal approximation are NP-hard.

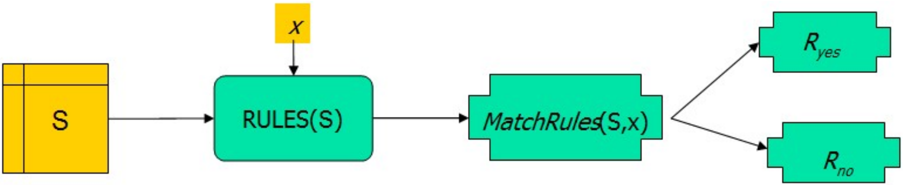
Rough Classifier Defined by Rules



Rough Classifier Defined by Rules



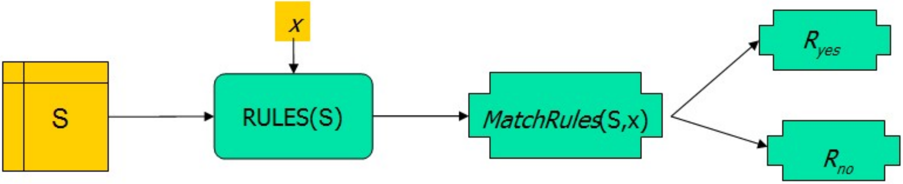
$$w_{yes} = \sum_{\mathbf{r} \in \mathbf{R}_{yes}} strength(\mathbf{r}) \qquad w_{no} = \sum_{\mathbf{r} \in \mathbf{R}_{no}} strength(\mathbf{r})$$



$$w_{yes} = \sum_{\mathbf{r} \in \mathbf{R}_{yes}} strength(\mathbf{r}) \qquad w_{no} = \sum_{\mathbf{r} \in \mathbf{R}_{no}} strength(\mathbf{r})$$

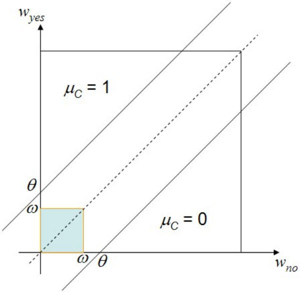
$$\mu_C(x) = \begin{cases} \text{undetermined} & \text{if } \max(w_{yes}, w_{no}) < \omega \\ 0 & \text{if } w_{no} - w_{yes} \geq \theta \text{ and } w_{no} > \omega \\ 1 & \text{if } w_{yes} - w_{no} \geq \theta \text{ and } w_{yes} > \omega \\ \frac{\theta + (w_{yes} - w_{no})}{2\theta} & \text{in other cases} \end{cases}$$

Rough Classifier Defined by Rules

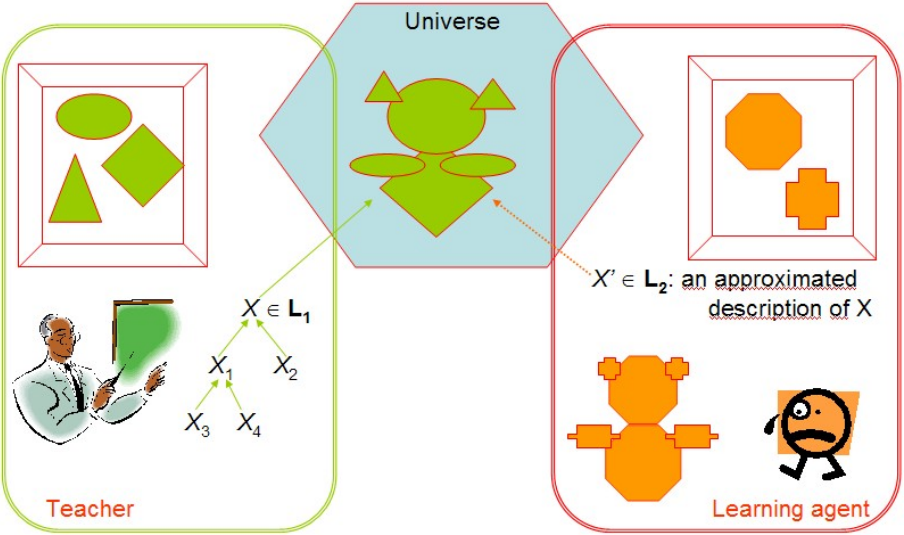


$$w_{yes} = \sum_{\mathbf{r} \in \mathbf{R}_{yes}} strength(\mathbf{r}) \qquad w_{no} = \sum_{\mathbf{r} \in \mathbf{R}_{no}} strength(\mathbf{r})$$

$$\mu_C(x) = \begin{cases} \text{undetermined} & \text{if } \max(w_{yes}, w_{no}) < \omega \\ 0 & \text{if } w_{no} - w_{yes} \geq \theta \text{ and } w_{r} \\ 1 & \text{if } w_{yes} - w_{no} \geq \theta \text{ and } w_{y} \\ \frac{\theta + (w_{yes} - w_{no})}{2\theta} & \text{in other cases} \end{cases}$$



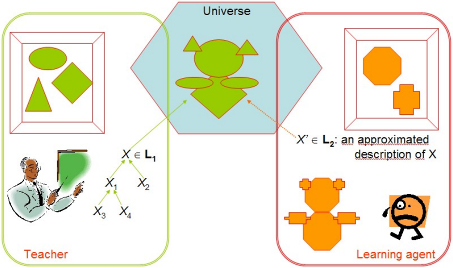
Reasoning via Layered Learning



Reasoning via Layered Learning

Given:

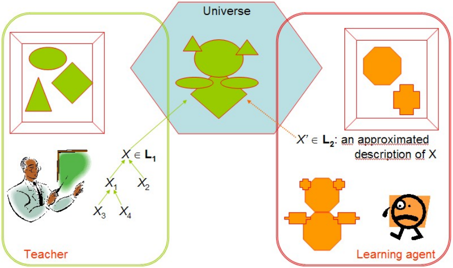
- U: the set of examples;
- A: the set of attributes;
- H: concept decomposition diagram;
- $D = dec_{C_1}, dec_{C_2}, ...dec_C$



Reasoning via Layered Learning

Given:

- U: the set of examples;
- A: the set of attributes;
- H: concept decomposition diagram;
- $D = dec_{C_1}, dec_{C_2}, ...dec_C$



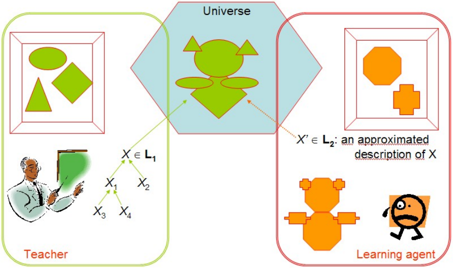
Goal: For each concept C in the hierarchy:

- construct a decision system $\mathbb{S}_C$;
- induce a rough approximation of C, i.e., a rough membership functions for C: $[\mu_{C^+}(x), \mu_{C^-}(x)]$

Reasoning via Layered Learning

Given:

- U: the set of examples;
- A: the set of attributes;
- H: concept decomposition diagram;
- $D = dec_{C_1}, dec_{C_2}, ...dec_C$



Goal: For each concept C in the hierarchy:

- construct a decision system $\mathbb{S}_C$;
- induce a rough approximation of C, i.e., a rough membership functions for C: $[\mu_{C^+}(x), \mu_{C^-}(x)]$

System control: The system can be tuned by

- uncertainty parameters: θ;
- learning parameters for each level.

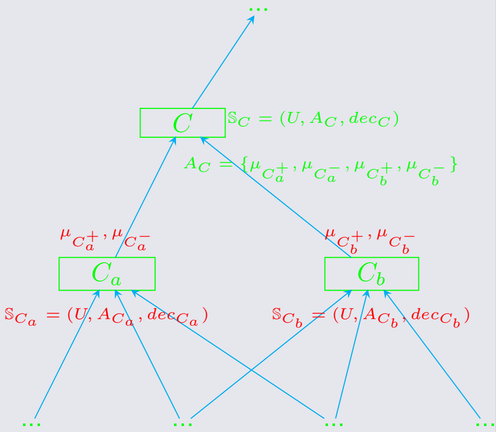
$\mathbb{S}_C = (U, A_C, dec_C)$, where

$$A_C = \{a_{C_1}, ..., a_{C_n}\}$$

is a collection of rough approximations of subconcepts $C_1, ..., C_n$:

- either $a_{C_j} = [\mu_{j^+}, \mu_{j^-}]$;
- or $a_{C_j} = [w_{yes}^{C_j}, w_{no}^{C_j}]$;

Schema



Layered learning algorithm

for $l := 0$ to max_level **do**
  for (any concept C_k at the level l in H) **do**
    if $l = 0$ **then**
      $\mathbb{S}_{C_k} := (U, A_k, dec_{C_k});$
    else
      $A_k := \bigcup O_{k_i};$
      $\mathbb{S}_{C_k} := (U, A_k, dec_{C_k});$
    end if
    generate the rule set $RULES(\mathbb{S}_{C_k})$ for decision table $\mathbb{S}_{C_k}$;
    generate the output vector $O_k = \{w_{yes}^{C_k}, w_{no}^{C_k}\}$,
  end for
end for

Example: Nursery data set

- Creator: Vladislav Rajkovic et al. (13 experts)
- Donors: Marko Bohanec (marko.bohanec@ijs.si)
  Blaz Zupan (blaz.zupan@ijs.si)
- Date: June, 1997
- Number of Instances: 12960 (instances completely cover the attribute space)
- Number of Attributes: 8

Attributes

NURSERY	not_recom, recommend, very_recom, priority, spec_prior
. EMPLOY	*Employment of parents and child's nursery*
. . parents	usual, pretentious, great_pret
. . has_nurs	proper, less_proper, improper, critical, very_crit
. STRUCT_FINAN	*Family structure and financial standings*
. . STRUCTURE	*Family structure*
. . . form	complete, completed, incomplete, foster
. . . children	1, 2, 3, more
. . housing	convenient, less_conv, critical
. . finance	convenient, inconv
. SOC_HEALTH	*Social and health picture of the family*
. . social	non-prob, slightly_prob, problematic
. . health	recommended, priority, not_recom

Method:

1. Use clustering algorithm to approximate intermediate concepts;
2. Use rule based algorithm (RSES system) to approximate the target concept;

Method:

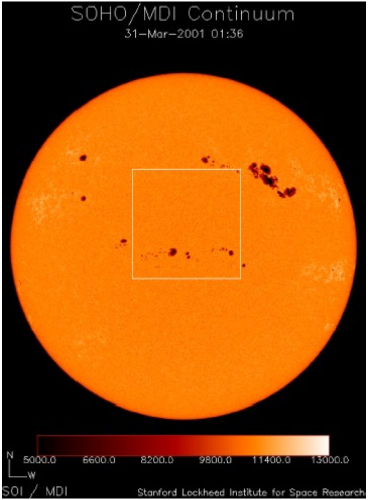
1. Use clustering algorithm to approximate intermediate concepts;
2. Use rule based algorithm (RSES system) to approximate the target concept;

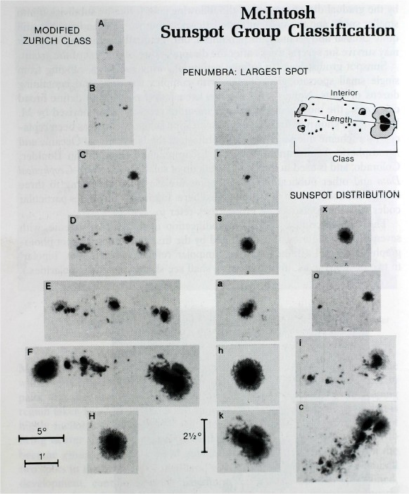
Results: (60% – training, 40% – testing)

	original attributes only	using intermediate concepts
Accuracy	83.4	99.9%
Coverage	85.3%	100%
Nr of rules	634	42 (for the target concept) 92 (for intermediate concepts)

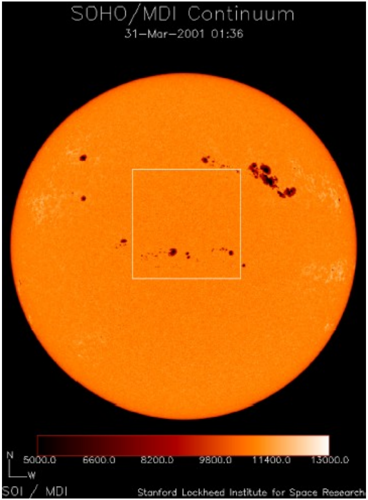
Outline

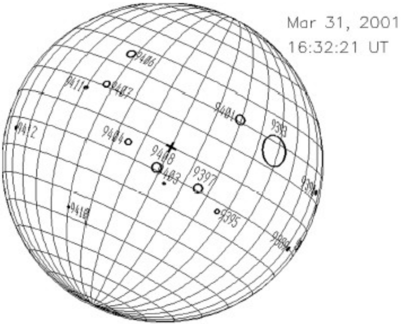
Sunspots Recognition and Classification





Sunspots Recognition and Classification

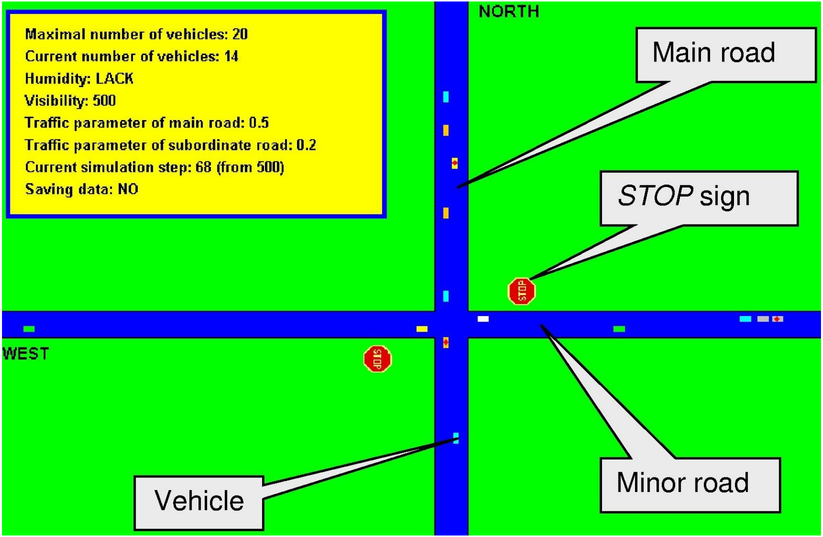




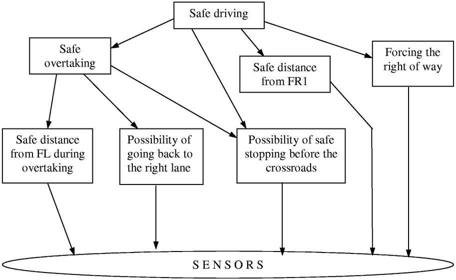
Joint USAF/NOAA Solar Region Summary (MAR 30, 2001 24:00:00 UT)

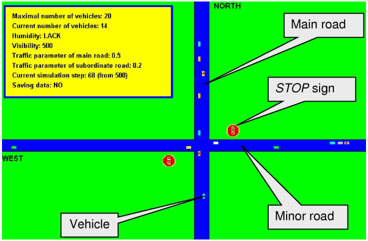
NMBR	LOCATI	LO	AREA	Mcl	LL	NN	MAG TYPE
9387	N08W92	216	0060	Hsx	02	01	Alpha
9389	S10W63	187	0050	Bxo	08	08	Beta
9390	N13W65	189	0050	Hsx	02	01	Alpha
9393	N17W30	154	2240	Fkc	19	63	Beta-Gamma-Delta
9395	S13W17	141	0050	Hsx	02	02	Alpha
9396	S06W85	209	0140	Dao	10	05	Beta
9397	S09W06	130	0180	Eao	15	23	Beta-Gamma
9401	N21W11	135	0230	Eki	13	37	Beta-Gamma
9403	S13E06	118	0010	Bxo	05	02	Beta
9404	S05E23	101	0080	Cao	04	07	Beta
9406	N26E41	083	0170	Hax	03	01	Alpha

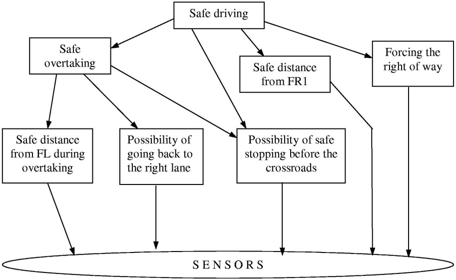
Road Situation Simulator

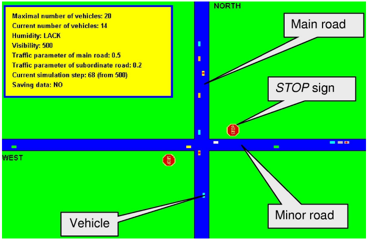


Road Situation Simulator

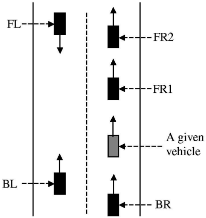








- **Universe** = set of vectors $s(c,t)$, where
  - c is a car;
  - t is a time instant;
- **Concept** = "Dangerous situation on the road";
- **Evaluation measures:**
  - True positive rate;
  - Coverage rate;
  - Computation time;
  - Rule sizes;



Differential Calculus to Function Approximation

- **ill-defined data:** limited number of objects and large number of attributes;
- prediction of a **real decision variable** based on nominal attributes;
- the need for the knowledge about the **real mechanisms behind the data**;

No.	Combination	B-1	1-4	4-6	6-E	PB	PE	Binding affinity
1	A2B2C2D2a2b2	1	1	1	1	1	1	4.52526247
2	A1B2C1D1a2b2	-1	1	-1	-1	1	1	4.818066119
3	A1B2C2D1a2b2	-1	1	1	-1	1	1	5.036009902
...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	
39	A1B1C1D1a1b1	-1	-1	-1	-1	-1	-1	8.963821581
40	A1B1C1D1a2b1	-1	-1	-1	-1	1	-1	8.998482244

Our proposition

Input

1. A decision table

$\mathbb{S}$	a_1	a_2	...	dec
u_1	1	-1	...	4.23
u_2	1	1	...	4.31
...	...	...	...	...
u_n	-1	1	...	8.92

2. Domain knowledge

Our proposition

Input

1. A decision table

$\mathbb{S}$	a_1	a_2	...	dec
u_1	1	-1	...	4.23
u_2	1	1	...	4.31
...	...	...	...	...
u_n	-1	1	...	8.92

2. Domain knowledge

First level

- Create comparing table

	a_1	a_2	...	change
u_1, u_2	$1 \to 1$	$-1 \to 1$	...	$\nearrow$
u_1, u_3	...	...	...	$\searrow$
...	...	...	...	...

- Learn the preference relation, i.e., decision rules of form

$$a_2 : -1 \to 1 \wedge a_6 = 1 ... \implies change = \searrow$$

Our proposition

Input

1. A decision table

$\mathbb{S}$	a_1	a_2	...	dec
u_1	1	-1	...	4.23
u_2	1	1	...	4.31
...	...	...	...	...
u_n	-1	1	...	8.92

2. Domain knowledge

First level

- Create comparing table

	a_1	a_2	...	change
u_1, u_2	$1 \rightarrow 1$	$-1 \rightarrow 1$	...	$\nearrow$
u_1, u_3	...	...	...	$\searrow$
...	...	...	...	...

- Learn the preference relation, i.e., decision rules of form

$$a_2 : -1 \rightarrow 1 \wedge a_6 = 1 ... \implies change = \searrow$$

Second level

- Ranking prediction;
- Decision value prediction;
- Experiment design,