

# Knowledge-based Systems for Bioinformatics

## Lecture 4: Approximate Reducts, Discretization

Senior Professor **Jan Komorowski**

Computational Biology and Bioinformatics Program,  
Department of Cell and Molecular Biology, Science for Life Laboratory, Uppsala University

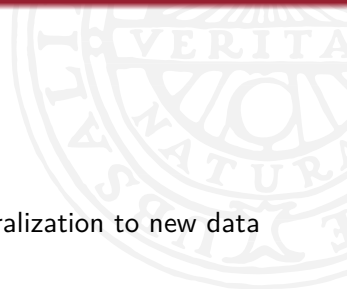
*jan.komorowski@icm.uu.se*

Winter 2026



UPPSALA  
UNIVERSITET

## Plan of the lecture



- ① Computing approximate reducts, as they may improve generalization to new data
- ② Missing values
- ③ How to deal with very big decision tables
- ④ Discretization, the art!

- 1 Computing approximate reducts
- 2 Missing values
- 3 Very big decision tables
- 4 Discretization



# Rough Membership

- Decision system  $\mathcal{A} = (U, A \cup \{d\})$

$$B \subseteq A$$

$$X \subseteq U$$

- The *rough membership function* quantifies the degree of the relative overlap between set  $X$  and the equivalence class to which  $x$  belongs.

$$\mu_x^B(x) : U \rightarrow [0, 1] \text{ and } \mu_x^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|}$$

Interpretation of  $\mu_x^B(x)$ : a frequency-based estimate of  $\Pr(x \in X \mid x, B)$

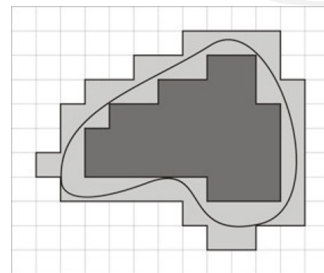
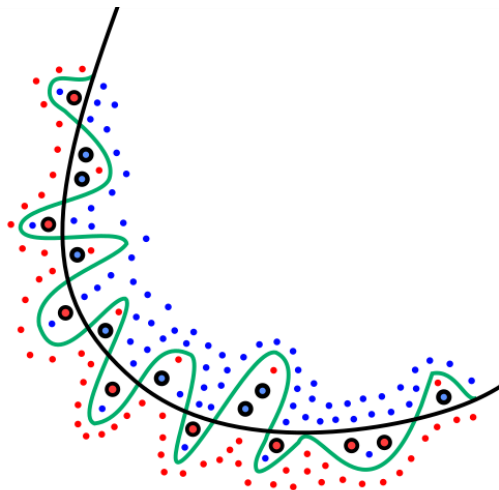


Figure 1:  $B \subseteq A$

# Pitfalls of learning "too well"



The green line represents an overfitted model and the black line represents a regularized model. While the green line best follows the training data, it is too dependant on that data and it is likely to have a higher error rate on new unseen data, compared to the black line.

Chabacano, CC BY-SA 4.0, via Wikimedia Commons, <https://upload.wikimedia.org/wikipedia/commons/1/19/Overfitting.svg>

# Overfitting and other issues with data

- The data we deal with is often:
  - Incomplete
  - Inconsistent
  - Noisy
- The reducts will include many attributes
  - Over-fitting
- Remedy: Approximate reducts
  - “almost reducts”
  - Shorter
  - May capture real patterns in the data, pick up less noise.



## Towards approximate reducts

- Recall: Rough Sets are used to give upper and lower approximations of concepts.

$$\mathcal{A} = (U, A \cup \{d\}), B \subseteq A$$

$B$  - lower approximation of  $X$ :  $\underline{B}X = \{x \mid [x]_B \subseteq X\}$   $B$  - upper approximation of

$X$ :  $\overline{B}X = \{x \mid [x]_B \cap X \neq \emptyset\}$   $B$  - boundary region of  $X$ :  $BN_B(X) = \overline{B}X - \underline{B}X$

## Rough membership function (RMF) - recall

- The *rough membership function* quantifies the degree of the relative overlap between the set  $X$  and the equivalence class to which  $x$  belongs.

$$\mu_X^B(x) : U \rightarrow [0, 1] \text{ and } \mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|}$$

- Examples:

$$\begin{aligned}\mu_{Lung}^A(x_1) &= \frac{|[x_1]_A \cap Lung|}{|[x_1]_A|} = \frac{|\{x_1, x_6\}|}{|\{x_1, x_6\}|} = \frac{2}{2} = 1 \\ \mu_{Lung}^A(x_7) &= \frac{|[x_7]_A \cap Lung|}{|[x_7]_A|} = \frac{|\emptyset|}{|\{x_7, x_8, x_{15}\}|} = \frac{0}{3} = 0 \\ \mu_{Lung}^A(x_3) &= \frac{|[x_3]_A \cap Lung|}{|[x_3]_A|} = \frac{|\{x_{18}\}|}{|\{x_3, x_{13}, x_{18}\}|} = \frac{1}{3} \\ \mu_{Colon}^A(x_3) &= \frac{|[x_3]_A \cap Colon|}{|[x_3]_A|} = \frac{|\{x_3, x_{13}\}|}{|\{x_3, x_{13}, x_{18}\}|} = \frac{2}{3}\end{aligned}$$

Eq. class	Gene1	Gene2	Gene3	Sm	Site of origin
$[x_1]$	↓	↓	0	Yes	{L}
$[x_2]$	0	0	0	Yes	{L}
$[x_5]$	0	↓	0	Yes	{L}
$[x_3]$	0	↓	↑	No	{C, L}
$[x_{10}]$	↓	↓	↑	No	{C, L}
$[x_7]$	↓	↑	0	No	{C}
$[x_9]$	0	↑	0	Yes	{C}
$[x_{14}]$	0	↑	↑	No	{C}



# Generalization of the RMF

- The formulae of upper and lower bounds for set approximation can be generalized to a given level of precision  $\pi$ , by using the rough membership function:

$$\underline{B}_\pi X = \{x \mid \mu_X^B(x) \geq \pi\}$$

$$\overline{B}_\pi X = \{x \mid \mu_X^B(x) > 1 - \pi\}$$

- The original upper and lower approximations are then a special case, when  $\pi = 1.0$
- Recall: Positive region

If  $X_{\mathcal{A}}^1, \dots, X_{\mathcal{A}}^{r(d)}$  are decision classes of  $\mathcal{A}$ ,  $B \subseteq A$   
then the set  $\underline{B}X_1 \cup \dots \cup \underline{B}X_{r(d)}$  is called the  $B$  - *positive region* of  $\mathcal{A}$   
and is denoted by  $POS_B(d)$

## Exercise: Positive regions

Compute the positive region of the decision classes using

- ① Gene1 & Gene2
- ② Gene1, Gene2 & Gene3

Recall:

$$POS_C(\{d\}) = \{x \in U \mid |\partial_C(x)| = 1\}$$

$$\underline{AX}^{Colon} = [x_7] \cup [x_9] \cup [x_{14}]$$

$$\underline{AX}^{Lung} = [x_1] \cup [x_2] \cup [x_5]$$

$$\underline{AX}^{Colon} \cup \underline{AX}^{Lung} =$$

$$[x_7] \cup [x_9] \cup [x_{14}] \cup [x_1] \cup [x_2] \cup [x_5] \neq U$$

( $[x_3]$  and  $[x_{10}]$  are not in the positive region)

Pat.	Gene1	Gene2	Gene3	Sm	Site of origin
$x_1$	↓	↓	0	Yes	Lung
$x_2$	0	0	0	Yes	Lung
$x_3$	0	↓	↑	No	Colon
$x_4$	0	0	0	Yes	Lung
$x_5$	0	↓	0	Yes	Lung
$x_6$	↓	↓	0	Yes	Lung
$x_7$	↓	↑	0	No	Colon
$x_8$	↓	↑	0	No	Colon
$x_9$	0	↑	0	Yes	Colon
$x_{10}$	↓	↓	↑	No	Lung
$x_{11}$	0	↓	0	Yes	Lung
$x_{12}$	0	↓	0	Yes	Lung
$x_{13}$	0	↓	↑	No	Colon
$x_{14}$	0	↑	↑	No	Colon
$x_{15}$	↓	↑	0	No	Colon
$x_{16}$	↓	↓	↑	No	Colon
$x_{17}$	0	↓	0	Yes	Lung
$x_{18}$	0	↓	↑	No	Lung

## Exercise: Positive regions (2)

$$\begin{aligned}
 & IND_A(\{Gene1, Gene2\}) \\
 &= \{\{x_6, x_{10}, x_{16}\}, \{x_2, x_4\}, \\
 &\quad \{x_3, x_5, x_{11}, x_{12}, x_{13}, x_{17}, x_{18}\}, \{x_7, x_8, x_{15}\}, \{x_9, x_{14}\}\}
 \end{aligned}$$

$$\begin{aligned}
 & IND_A(\{Gene1, Gene2, Gene3\}) \\
 &= \{\{x_1, x_6\}, \{x_2, x_4\}, \{x_3, x_{13}, x_{18}\}, \\
 &\quad \{x_5, x_{11}, x_{12}, x_{17}\}, \\
 &\quad \{x_7, x_8, x_{15}\}, \{x_9\}, \{x_{10}, x_{16}\}, \{x_{14}\}\}
 \end{aligned}$$

Now, the cardinality of equivalence classes:

$$G1, G2: 2 + 3 + 2 = 7$$

$$G1, G2, G3: 2 + 2 + 4 + 3 + 1 + 1 = 13$$

Pat.	Gene1	Gene2	Gene3	S	Site of origin
$x_1$	↓	↓	0	Yes	Lung
$x_6$	↓	↓	0	Yes	Lung
$x_2$	0	0	0	Yes	Lung
$x_4$	0	0	0	Yes	Lung
$x_5$	0	↓	0	Yes	Lung
$x_{11}$	0	↓	0	Yes	Lung
$x_{12}$	0	↓	0	Yes	Lung
$x_{17}$	0	↓	0	Yes	Lung
$x_{16}$	↓	↓	↑	No	Colon
$x_{10}$	↓	↓	↑	No	Lung
$x_{18}$	0	↓	↑	No	Lung
$x_{13}$	0	↓	↑	No	Colon
$x_3$	0	↓	↑	No	Colon
$x_7$	↓	↑	0	No	Colon
$x_8$	↓	↑	0	No	Colon
$x_{15}$	↓	↑	0	No	Colon
$x_9$	0	↑	0	Yes	Colon
$x_{14}$	0	↑	↑	No	Colon

## Dependency of decision on attributes

- Hint: We may remove some of the attributes. How?
- Dependency of attributes:
  - Consider  $\mathcal{A} = (U, A \cup \{d\})$  and a subset of attributes  $C \subseteq A$
  - The dependency of  $\gamma$  of  $d$  on  $C$  is then defined as

$$\gamma(C, \{d\}) = \frac{|POS_c(\{d\})|}{|U|}$$

where

$$POS_c(\{d\}) = \{x \in U \mid |\partial_c(x)| = 1\}$$

Generalized  
decision class

- This is called the *degree of dependency*, and it is the fraction of elements in  $U$  with a unique value on its decision attribute, when the equivalence classes are constructed by the attributes in  $C$ .

# Example of dependency

- Recall:

$$\gamma(C, \{d\}) = \frac{|POS_c(\{d\})|}{|U|}$$

where

$$POS_c(\{d\}) = \{x \in U \mid |\partial_c(x)| = 1\}$$

- Example:

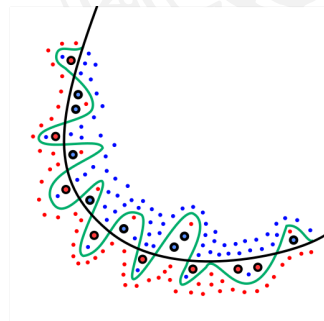
$$\begin{aligned} \gamma(\{Gene1, Gene2\}, \{site\ of\ origin\}) \\ = \frac{2 + 3 + 2}{18} = \frac{7}{18} \end{aligned}$$

$$\begin{aligned} \gamma(\{Gene1, Gene2, Gene3\}, \{site\ of\ origin\}) \\ = \frac{13}{18} \end{aligned}$$

Pat.	Gene1	Gene2	Gene3	Sm	Site of origin
x <sub>1</sub>	↓	↓	0	Yes	Lung
x <sub>2</sub>	0	0	0	Yes	Lung
x <sub>3</sub>	0	↓	↑	No	Colon
x <sub>4</sub>	0	0	0	Yes	Lung
x <sub>5</sub>	0	↓	0	Yes	Lung
x <sub>6</sub>	↓	↓	0	Yes	Lung
x <sub>7</sub>	↓	↑	0	No	Colon
x <sub>8</sub>	↓	↑	0	No	Colon
x <sub>9</sub>	0	↑	0	Yes	Colon
x <sub>10</sub>	↓	↓	↑	No	Lung
x <sub>11</sub>	0	↓	0	Yes	Lung
x <sub>12</sub>	0	↓	0	Yes	Lung
x <sub>13</sub>	0	↓	↑	No	Colon
x <sub>14</sub>	0	↑	↑	No	Colon
x <sub>15</sub>	↓	↑	0	No	Colon
x <sub>16</sub>	↓	↓	↑	No	Colon
x <sub>17</sub>	0	↓	0	Yes	Lung
x <sub>18</sub>	0	↓	↑	No	Lung

# Overfitting

- What is the corresponding rule-based phenomenon to curve overfitting?
  - The more detail, the longer Boolean expression in the cells of the discernibility matrix
  - Length of the rules increases, more and more descriptors
  - IF  $C_1 \& C_2 \& C_3 \& C_4 \& \dots \& C_n$  THEN Decision
  - Such reducts (and consequently the rules) may be too specific and will not generalize well to new, unseen data
- How can we control it?
- Idea:
  - Let us try “almost” reducts, that is, expressions with fewer descriptors



## Approximate reducts - definition and example

- An *approximate reduct* is a set of attributes that discern objects in  $U$  to some degree.
- *Error of reduct approximation* of a reduct  $C$  with  $B \subseteq C$  is defined as:

$$\epsilon_{(C, \{d\})}(B) = \frac{\gamma(C, \{d\}) - \gamma(B, \{d\})}{\gamma(C, \{d\})} = 1 - \frac{\gamma(B, \{d\})}{\gamma(C, \{d\})}$$

and measures how well  $B$  approximates  $C$  (relative to the decision attribute)

- Example: Let  $B = \{Gene1, Gene2\}$  and  $C = \{Gene1, Gene2, Gene3\}$ :

$$\frac{7}{18}$$

$$\frac{13}{18}$$

$$\epsilon_{(C, \{d\})}(B) = \frac{\gamma(C, \{d\}) - \gamma(B, \{d\})}{\gamma(C, \{d\})} = 1 - \frac{\gamma(B, \{d\})}{\gamma(C, \{d\})} = 1 - \frac{7}{13} = \frac{6}{13}$$

$$\gamma(C, \{d\}) = \frac{|POS_C(\{d\})|}{|U|}$$

where  $POS_C(\{d\}) = \{x \in U \mid |\partial_C(x)| = 1\}$

# Computing approximate reducts

- Approximate reducts can be computed using a *genetic algorithm*.
- Population: Many possible reducts represented as bit-strings of length  $|A|$ .
- Fitness function: Weighted sum of *reduct length* and  $\epsilon_{(C, \{d\})}(B)$ .
- Selection: Relative to fitness.
- Crossing over and mutations.



- 1 Computing approximate reducts
- 2 Missing values
- 3 Very big decision tables
- 4 Discretization



## Missing values

- What do we do if an attribute lacks one value?
- Complete by:
  - Averaging the other values
  - Selecting the smallest
  - Selecting the largest
  - Checking distribution and choosing the mean
  - Adjusting the algorithm to handle N/A
  - Remove the object!



- 1 Computing approximate reducts
- 2 Missing values
- 3 Very big decision tables
- 4 Discretization



## How to deal with big decision tables

- Very large numbers of attributes
- Very large numbers of objects



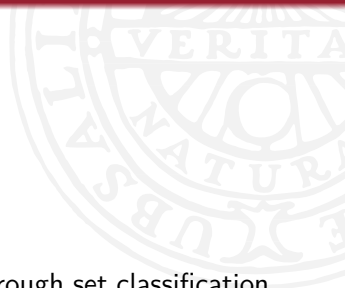
# Very large numbers of objects - Dynamic reducts



- A decision table with  $m \gg k$
- Idea: Sample the universe  $U$  several times and compute reducts (cf. Monte Carlo!!!)
- Keep reducts that occur frequently (e.g. in at least 0.7 of the runs).
- Gives fewer and more general reducts.
- Takes time.
- An algorithm for computing reducts must be selected. This is just a wrapper around it.
- Similar idea to Random Forests, Bagging and Boosting..

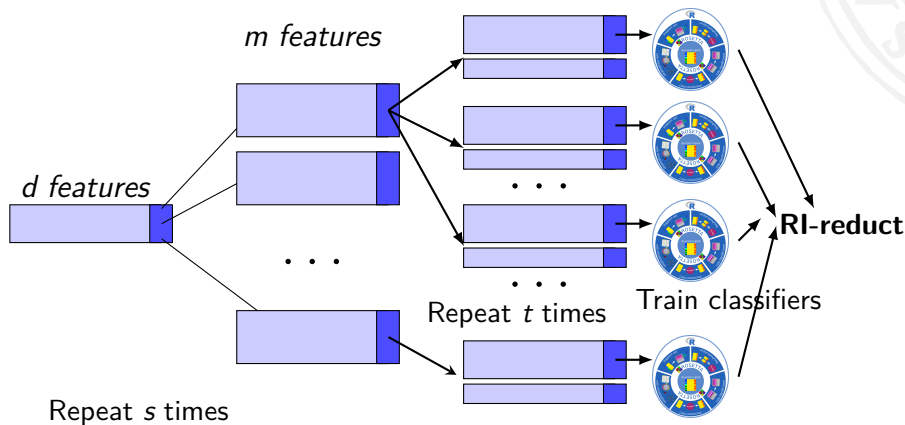
## Random Reducts

- Recall the construction by MCFS
- Replace the computation of decision tree classification with rough set classification
- Modify accordingly the computation of Relative Importance (of attributes)



# Monte Carlo feature selection with rough sets - Random reducts\*

- Replace the computation of decision tree classification with rough set classification
- Modify accordingly the computation of Relative Importance (of attributes)



\* *Random Reducts: A Monte Carlo Rough Set-based Method for Feature Selection in Large Datasets*, Kruczyk, Marcin, Baltzer, Nicholas, Mieczkowski, Jakub, Damiński, Michał, Koronacki, Jacek, Komorowski, Jan; DOI: 10.3233/FI-2013-909

- 1 Computing approximate reducts
- 2 Missing values
- 3 Very big decision tables
- 4 Discretization**





# Discretization

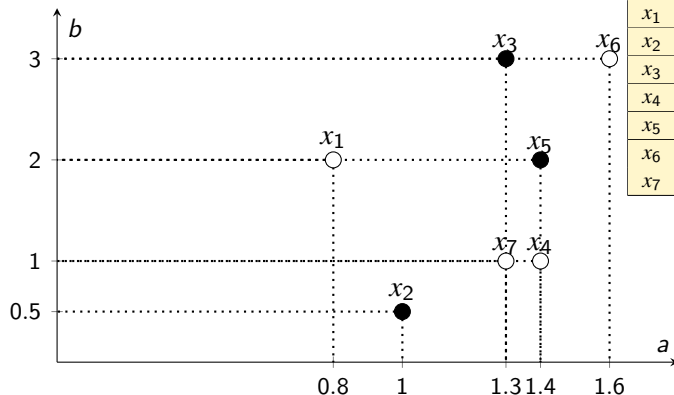


- Human reasoning with many factors is often qualitative.
- Experimental data is mostly quantitative.
- From quantitative to qualitative data. Examples:
  - Number of heart beats: Low, Normal or High Pulse.
  - Degrees Celsius: Very Cold, Cold, Normal, Warm or Hot.
- Discretization is the process achieving this categorization.

# Discretization - basic issues

- Definition of cuts
- How many cuts?

- How to decide where to cut?
- Do we need all the cuts?



U	a	b	d
$x_1$	0.8	2	1
$x_2$	1	0.5	0
$x_3$	1.3	3	0
$x_4$	1.4	1	1
$x_5$	1.4	2	0
$x_6$	1.6	3	1
$x_7$	1.3	1	1



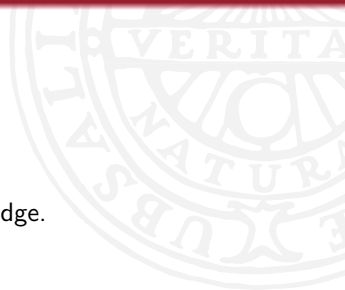
U	a	b	d
$x_1$	0	2	1
$x_2$	1	0	0
$x_3$	1	2	0
$x_4$	1	1	1
$x_5$	1	2	0
$x_6$	2	2	1
$x_7$	1	1	1

# Notation

- Given a decision system  $\mathcal{A} = (U, A \cup \{d\})$ 
  - Cuts: A sequence of values  $v_1 < v_2 < \dots < v_k$  where  $v_i \in V_a$
  - Partition of  $\mathcal{A}$  : A partition of every  $V_a$  into non-overlapping intervals defined by a sequence of cuts.
  - Size of a partition: The number of cuts used in the definition of the partition.
- After the partition we obtain  $\mathcal{A}^P = (U, A^P \cup \{d\})$ 
  - This is the discretized decision system  $\mathcal{A} = (U, A \cup \{d\})$  where the attribute values have been replaced according to partition  $P$ .

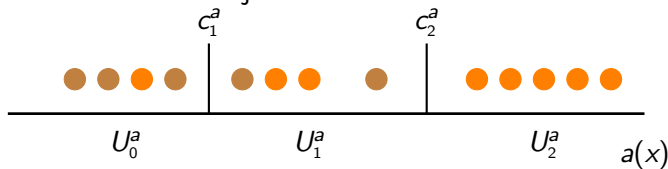
# Different types of discretization

- There are many ways to come up with a partition  $P$ :
  - User defined: May be a good idea if we have domain knowledge.
  - Infer cuts from the data:
    - Equal frequency binning
    - Naive (and semi-naive) method
    - Entropy based
    - Linear discriminant discretization
    - Boolean reasoning algorithm
    - and many more.



# Equal frequency binning

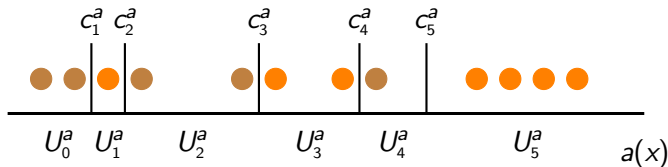
- Partitions the range of  $a(x)$  into  $(n + 1)$  intervals so that each interval has the same number of objects.



- Not a very smart algorithm (but sometimes used anyway because it is simple).

# Naive discretization algorithm

- Makes a cut between two adjacent objects if they have different classes.



- More formally: Let  $v_1^a < \dots < v_i^a < \dots < v_m^a$  be the sorted values of  $a$ , where  $\partial_j^a = \{d(x) | a(x) = v_j^a\}$ . Then

$$c_a = \left\{ \frac{v_j^a + v_{j+1}^a}{2} \mid \partial_j^a \neq \partial_{j+1}^a \text{ and } j = 1, \dots, m-1 \right\}$$

- This often produces a lot of cuts.

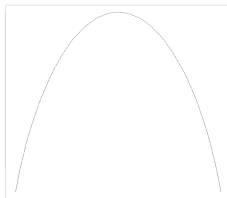
# Entropy based discretization I\*

- Entropy measures the average information content (number of bits) for an outcome of a random variable  $X$

$$\text{Entropy}(X) = - \sum_{i=1}^I P(X = c_i) \log_2 P(X = c_i)$$

- If  $X$  can be assigned to one of two classes ( $c_1$  or  $c_2$ )

$\text{Entropy}(X)$



$P(X=C_0)$

\*Compare to decision tree construction

## Entropy based discretization II

- The range of  $a(x)$  is divided recursively into intervals by
  - selecting the cut with highest entropy-gain
  - testing if this gain is significant
- The gain in entropy for cut  $c$  is defined as:

$$U_0 = \{x \in U | a(x) < c\}$$
$$U_1 = \{x \in U | c \leq a(x)\}$$

$$\text{Ent}(U) = - \sum_{i=1}^I P(d_i | U) \log_2 P(d_i | U)$$

$$\text{Gain}(a, c; U) = \text{Ent}(U) - \left( \frac{|U_0|}{|U|} \text{Ent}(U_0) + \frac{|U_1|}{|U|} \text{Ent}(U_1) \right)$$

- A cut  $c$  is accepted if:

$$\text{Gain}(a, c; U) > \frac{\log_2(|U| - 1)}{|U|} + \frac{\log_2(3^I - 2) - (I * \text{Ent}(U) - I_0 * \text{Ent}(U_0) - I_1 * \text{Ent}(U_1))}{|U|}$$

- Where  $I_i$  is the number of decision classes in  $U$ .



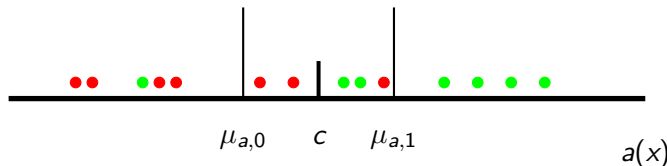
# Linear discriminant discretization I\*

Discriminant analysis is used for discretization:

- One bin  $U_i^a$  is created for each class  $d_i$  and the value of  $a(x)$  for object  $x$  is assigned to  $U_1^a$  if:

$$p(a(x)|d_1) * p(d_1) \geq p(a(x)|d_0) * p(d_0)$$

- We assume that the values of  $a$  are normally distributed with the same variance:



$$U_0^a = \{x \in U | a(x) < c\}$$

$$U_1^a = \{x \in U | c \leq a(x)\}$$

## Linear discriminant discretization II\*

- Assuming the values of  $a$  are normally distributed with the same variance, we may assign  $a(x)$  to  $U_1^a$  if

$$\frac{\mu_{a,1} - \mu_{a,0}}{\sigma_a^2} a(x) - \frac{\mu_{a,1}^2 - \mu_{a,0}^2}{2\sigma_a^2} \geq v \text{ where } v = \ln \frac{p(d_0)}{p(d_1)}$$

- This is similar to making a cut  $c$

$$c = \frac{\mu_{a,1} + \mu_{a,0}}{2} + \frac{\mu_a^2}{\mu_{a,1} - \mu_{a,0}} v$$

# Boolean Reasoning Discretization Algorithm

- Given a consistent decision system

$$\mathcal{A} = (U, A \cup \{d\}) \text{ where } V_a = R \text{ for all } a \in A$$

- Find a minimal partition  $P_a$  of  $V_a$  for all  $a \in A$  such that

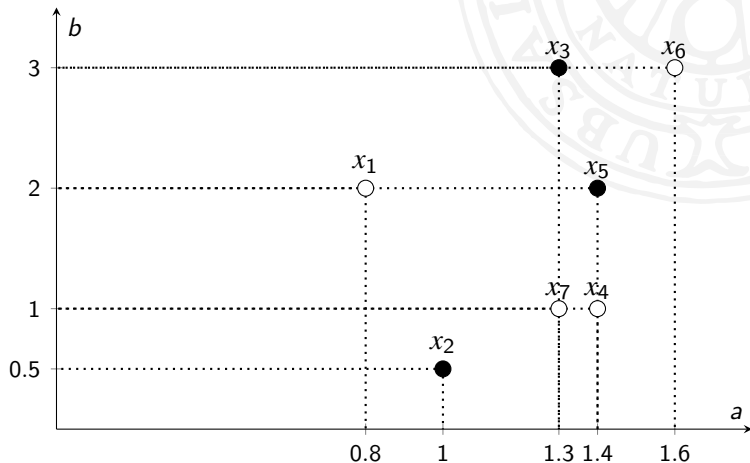
$$\mathcal{A}^P = (U, A^P \cup \{d\})$$

is still a consistent decision system



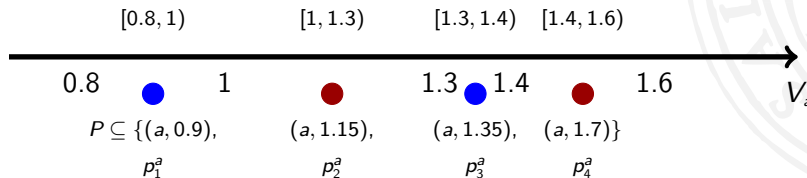
# Geometrical interpretation

U	a	b	d
$x_1$	0.8	2	1
$x_2$	1	0.5	0
$x_3$	1.3	3	0
$x_4$	1.4	1	1
$x_5$	1.4	2	0
$x_6$	1.6	3	1
$x_7$	1.3	1	1



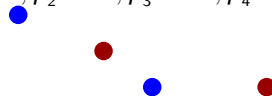
- What is an optimal partition of this?

# Boolean interpretation



One-to-one correspondence:  
Partition  $\longleftrightarrow$  Valuation of Boolean variables

$$P = \{(a, 0.9), (a, 1.35)\} \Leftrightarrow p_1^a = T, p_2^a = F, p_3^a = T, p_4^a = F$$



# Boolean reasoning

- $\Phi^A$ : Given a partition (a valuation), are all objects discernible (is the formula satisfied)?
  - $(p_1 \wedge p_2) \vee (p_3 \wedge p_4)$  is satisfied for the valuation
  - $p_1^a = T, p_2^a = T, p_3^a = F, p_4^a = F$  but not for  $p_1^a = F, p_2^a = T, p_3^a = T, p_4^a = F$
- $\psi(i, j)$ : Given a partition (a valuation), are objects  $u_i$  and  $u_j$  discernible (the formula satisfied)?
- Problem reformulation:
  - Find a minimal partition  $P \equiv$  Find prime implicants of  $\Phi^A$ .
- Remember that objects from the same decision class need not be discerned!

## Discernibility formula construction

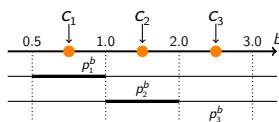
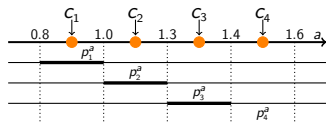
U	a	b	d
$x_1$	0.8	2	1
$x_2$	1	0.5	0
$x_3$	1.3	3	0
$x_4$	1.4	1	1
$x_5$	1.4	2	0
$x_6$	1.6	3	1
$x_7$	1.3	1	1

 $\Phi^A =$  $\psi(1, 2) \wedge \psi(1, 3) \wedge \dots \wedge \psi(5, 7)$  $\psi(1, 2) = p_1^a \vee p_1^b \vee p_2^b$  $\psi(1, 3) = p_1^a \vee p_2^a \vee p_3^b$  $\dots$  $\psi(1, 2)$  is the discernibility of objects  $x_1$  and  $x_2$ : $x_1$  and  $x_2$  are discernible by cut  $p_1^a = 0.9$  because  $x_1^a = 0.8 < 0.9 < x_2^a = 1.0$ 

or by cut

 $p_1^b = 0.75$  because $x_1^b = 0.5 < 0.75 < x_1^b = 2.0$ 

or by cut

 $p_2^b = 1.5$  because $x_2^b = 0.5 < 1.5 < x_1^b = 2.0$ 

## Discernibility formula construction II

$$\psi(x_1, x_2) = p_1^a \vee p_1^b \vee p_2^b$$

$$\psi(x_1, x_3) = p_1^a \vee p_2^a \vee p_3^b$$

$$\psi(x_1, x_5) = p_1^a \vee p_2^a \vee p_3^a$$

$$\psi(x_4, x_2) = p_2^a \vee p_3^a \vee p_1^b$$

$$\psi(x_4, x_3) = p_2^a \vee p_2^b \vee p_3^b$$

$$\psi(x_4, x_5) = p_2^b$$

$$\psi(x_6, x_2) = p_2^a \vee p_3^a \vee p_4^a \vee p_1^b \vee p_2^b \vee p_3^b$$

$$\psi(x_6, x_3) = p_3^a \vee p_4^a$$

$$\psi(x_6, x_5) = p_4^a \vee p_3^b$$

$$\psi(x_7, x_2) = p_2^a \vee p_1^b$$

$$\psi(x_7, x_3) = p_2^b \vee p_3^b$$

$$\psi(x_7, x_5) = p_3^a \vee p_2^b$$

- These correspond to the entries of a discernibility matrix.






## Discernibility formula construction III

- The discernibility formula can be constructed by:

$$\Phi^A = \bigwedge \{ \psi(i, j) : d(x_i) \neq d(x_j) \}$$

- This gives the following formula for the example:

$$\begin{aligned} \Phi^A = & (p_1^a \vee p_1^b \vee p_2^b) \wedge (p_1^a \vee p_2^a \vee p_3^b) \\ & \wedge (p_1^a \vee p_2^a \vee p_3^a) \\ & \wedge (p_2^a \vee p_3^a \vee p_1^b) \wedge (p_2^a \vee p_2^b \vee p_3^b) \\ & \wedge (p_2^a \vee p_3^a \vee p_4^a \vee p_1^b \vee p_2^b \vee p_3^b) \\ & \wedge (p_3^a \vee p_4^a) \wedge (p_4^a \vee p_3^b) \wedge (p_2^a \vee p_1^b) \\ & \wedge (p_2^b \vee p_3^b) \wedge (p_3^a \vee p_2^b) \wedge p_2^b \end{aligned}$$



U	a	b	d
$x_1$	0.8	2	1
$x_2$	1	0.5	0
$x_3$	1.3	3	0
$x_4$	1.4	1	1
$x_5$	1.4	2	0
$x_6$	1.6	3	1
$x_7$	1.3	1	1

# Discernibility formula construction IV

- Discernibility formula

$$\Phi^A = (p_1^a \vee p_1^b \vee p_2^b) \wedge (p_1^a \vee p_2^a \vee p_3^b)$$

$$\wedge (p_1^a \vee p_2^a \vee p_3^a)$$

$$\wedge (p_2^a \vee p_3^a \vee p_1^b) \wedge (p_2^a \vee p_2^b \vee p_3^b)$$

$$\wedge (p_2^a \vee p_3^a \vee p_4^a \vee p_1^b \vee p_2^b \vee p_3^b)$$

$$\wedge (p_3^a \vee p_4^a) \wedge (p_4^a \vee p_3^b) \wedge (p_2^a \vee p_1^b)$$

$$\wedge (p_2^b \vee p_3^b) \wedge (p_3^a \vee p_2^b) \wedge p_2^b$$

- Discernibility formula in DNF form

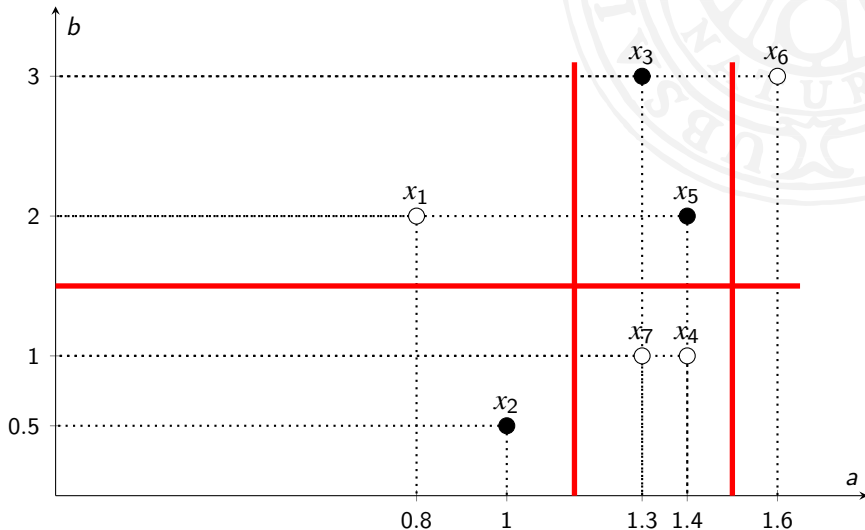
$$\Phi^A = (p_2^a \wedge p_4^a \wedge p_2^b) \vee (p_2^a \wedge p_3^a \wedge p_2^b \wedge p_3^b) \vee (p_3^a \wedge p_1^b \wedge p_2^b \wedge p_3^b) \vee (p_1^a \wedge p_4^a \wedge p_1^b \wedge p_2^b)$$

$$S = \{p_2^a, p_4^a, p_2^b\}$$

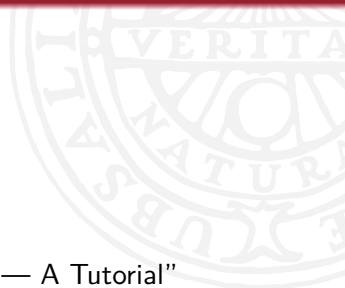
$$P(S) = \{(a, 1.15), (a, 1.5), (b, 1.5)\}$$

# Discretized decision system

U	a	b	d
$x_1$	0.8	2	1
$x_2$	1	0.5	0
$x_3$	1.3	3	0
$x_4$	1.4	1	1
$x_5$	1.4	2	0
$x_6$	1.6	3	1
$x_7$	1.3	1	1



## Suggestion



- Read Section 7.2 on Boolean Discretization in “Rough Sets — A Tutorial”