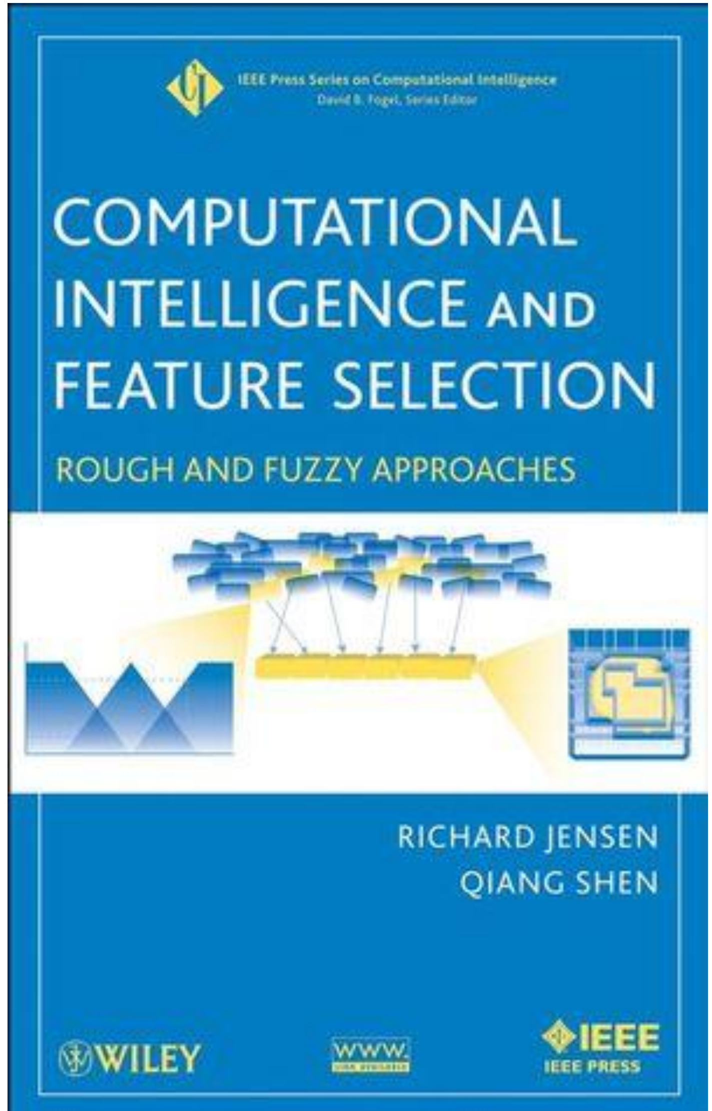
Fuzzy-rough data mining

Richard Jensen
rkj@aber.ac.uk

http://users.aber.ac.uk/rkj

An advert...


IEEE Press Series on Computational Intelligence
David B. Fogel, Series Editor
COMPUTATIONAL INTELLIGENCE AND FEATURE SELECTION
ROUGH AND FUZZY APPROACHES
RICHARD JENSEN
QIANG SHEN
WILEY
WWW.
IEEE
IEEE PRESS

Outline

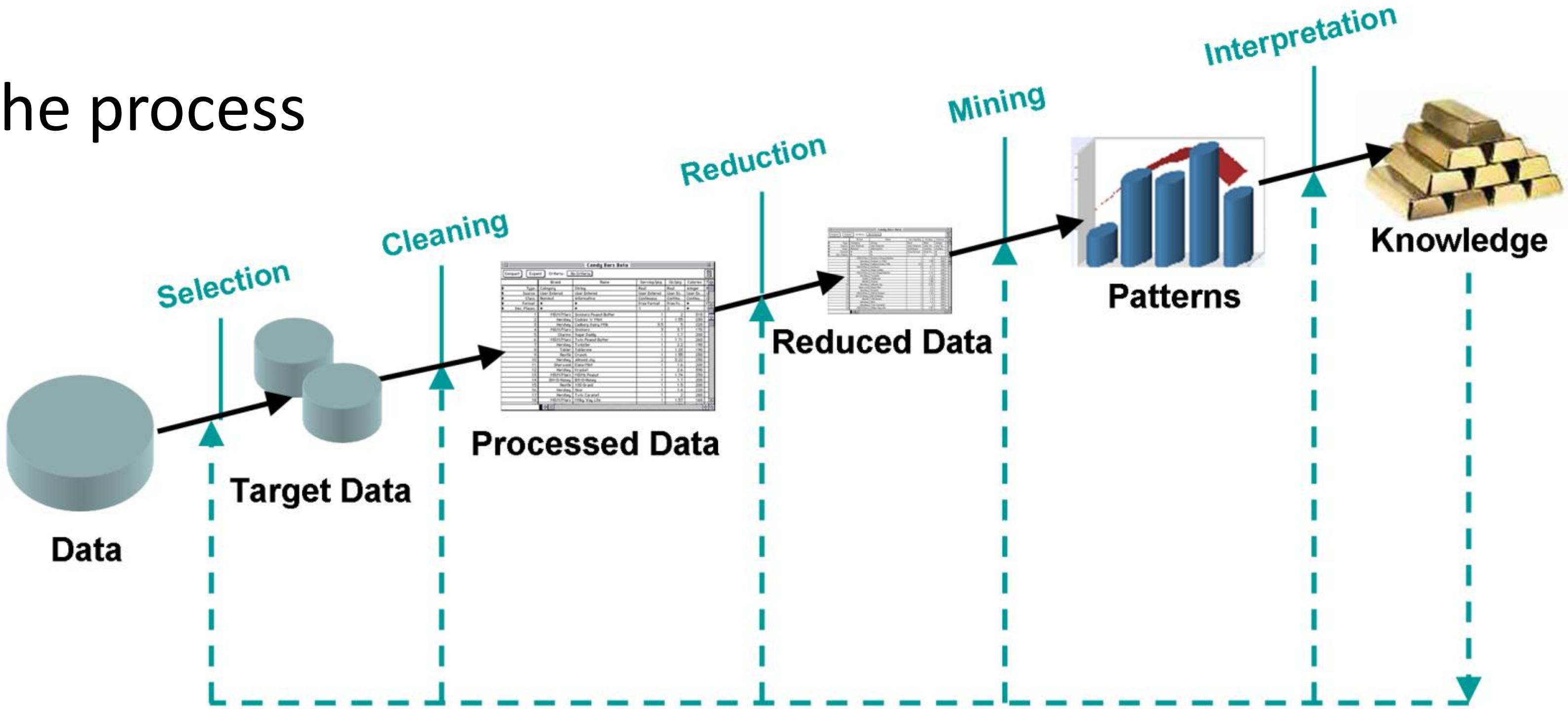
- Introduction to knowledge discovery/data mining
- Feature selection and rough set theory
- Fuzzy-rough feature selection and extensions
- Fuzzy-rough instance selection
- Fuzzy-rough classification/prediction
- Practical session with Weka

Data mining

- Process of semi-automatically analyzing large databases to find patterns (or models) that are:
  - **valid**: hold on new data with some certainty
  - **novel**: non-obvious to the system
  - **useful**: should be possible to act on the item
  - **understandable**: humans should be able to interpret the pattern/model

Knowledge discovery

- The process



- The problem of too much data
  - Requires storage
  - Intractable for data mining algorithms
  - Noisy or irrelevant data is misleading/confounding

Results of Data Mining include:

- *Forecasting* what may happen in the future
- *Classifying* people or things into groups by recognizing patterns
- *Clustering* people or things into groups based on their attributes
- *Associating* what events are likely to occur together
- *Sequencing* what events are likely to lead to later events

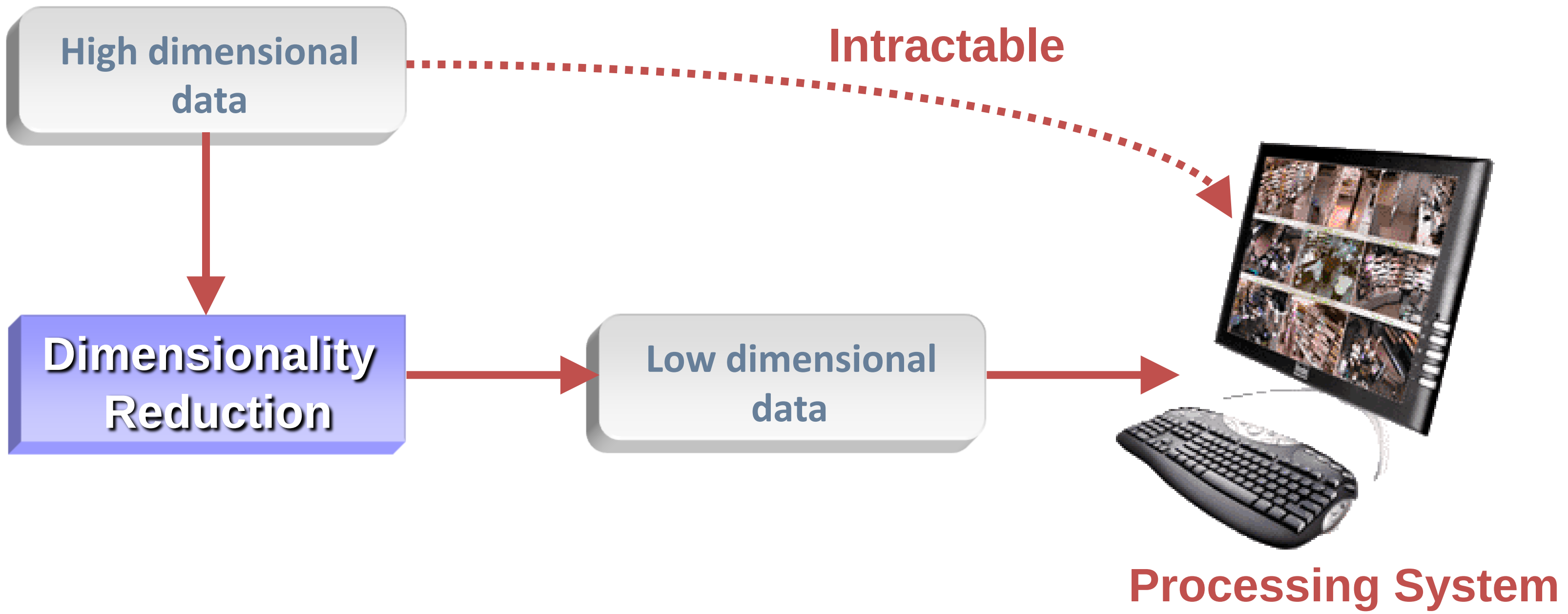
Applications

- Banking: loan/credit card approval
  - predict good customers based on old customers
- Customer relationship management:
  - identify those who are likely to leave for a competitor
- Targeted marketing:
  - identify likely responders to promotions
- Fraud detection: telecommunications, financial transactions
  - from an online stream of events identify fraudulent events
- Medicine: disease outcome, effectiveness of treatments
  - analyze patient disease history: find relationship between diseases

Feature Selection

Feature selection

- Why dimensionality reduction/feature selection?



- Growth of information - need to manage this effectively
- Curse of dimensionality - a problem for machine learning and data mining
- Data visualisation - graphing data

Why do it?

- Case 1: We're interested in *features*
  - We want to know which are relevant
  - If we fit a model, it should be *interpretable*

- Case 2: We're interested in *prediction*
  - Features are not interesting in themselves
  - We just want to build a good classifier (or other kind of predictor)

Case 1

- We want to know which features are relevant; we don’t necessarily want to do prediction

- *E.g. what causes lung cancer?*
  - Features are aspects of a patient’s medical history
  - Decision feature: did the patient develop lung cancer?
  - Which features best predict whether lung cancer will develop?

- *E.g. what stabilizes protein structure?*
  - Features are structural aspects of a protein
  - Real-valued decision feature—protein energy
  - Features that give rise to low energy are stabilizing

Case 2

- We want to build a good predictor

- *E.g. text classification*
  - Features for all English words, and maybe all word pairs
  - Common practice: throw in every feature you can think of, let feature selection get rid of useless ones
  - Training too expensive with all features

- *E.g. disease diagnosis*
  - Features are outcomes of expensive medical tests
  - Which tests should we perform on the patient?

Aspects of features

- Correlation
  - The extent to which one subset of features depends on another

- So ideally we want:
  - **High relevancy**: high correlation with the decision feature
  - **Low redundancy**: very little correlation between features within a subset

Problems

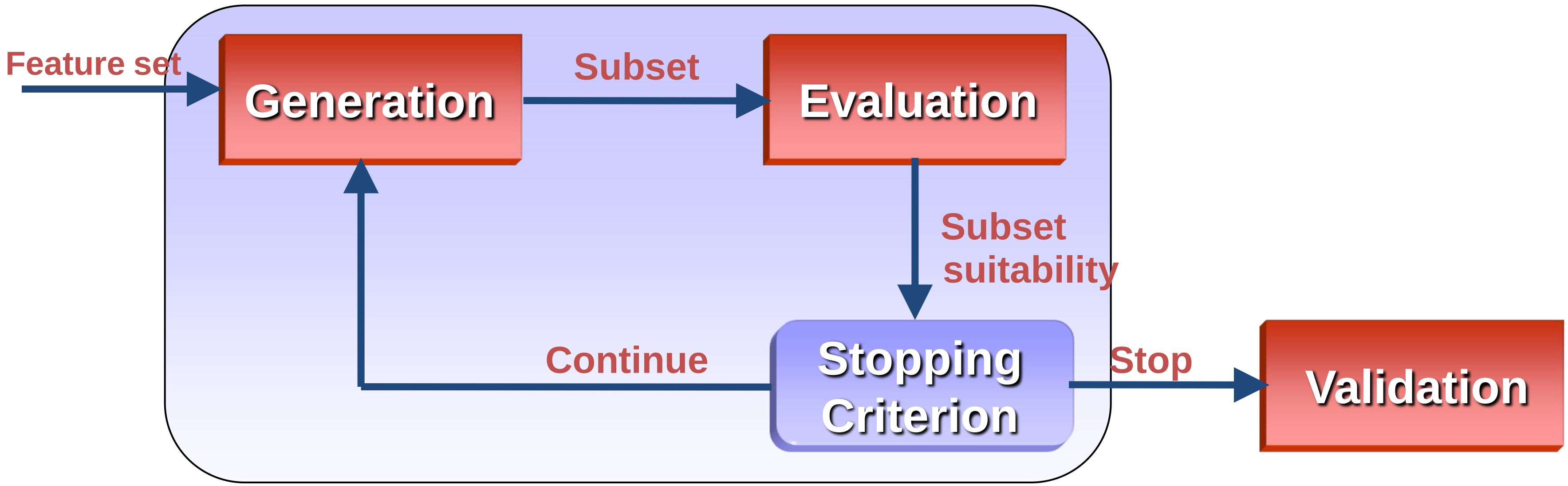
- Noisy data
  - Due to measurement inaccuracies
  - Can also affect decision feature values

- Inconsistent data

	f_1	f_2	class
instance 1	a	b	c1
instance 2	a	b	c2

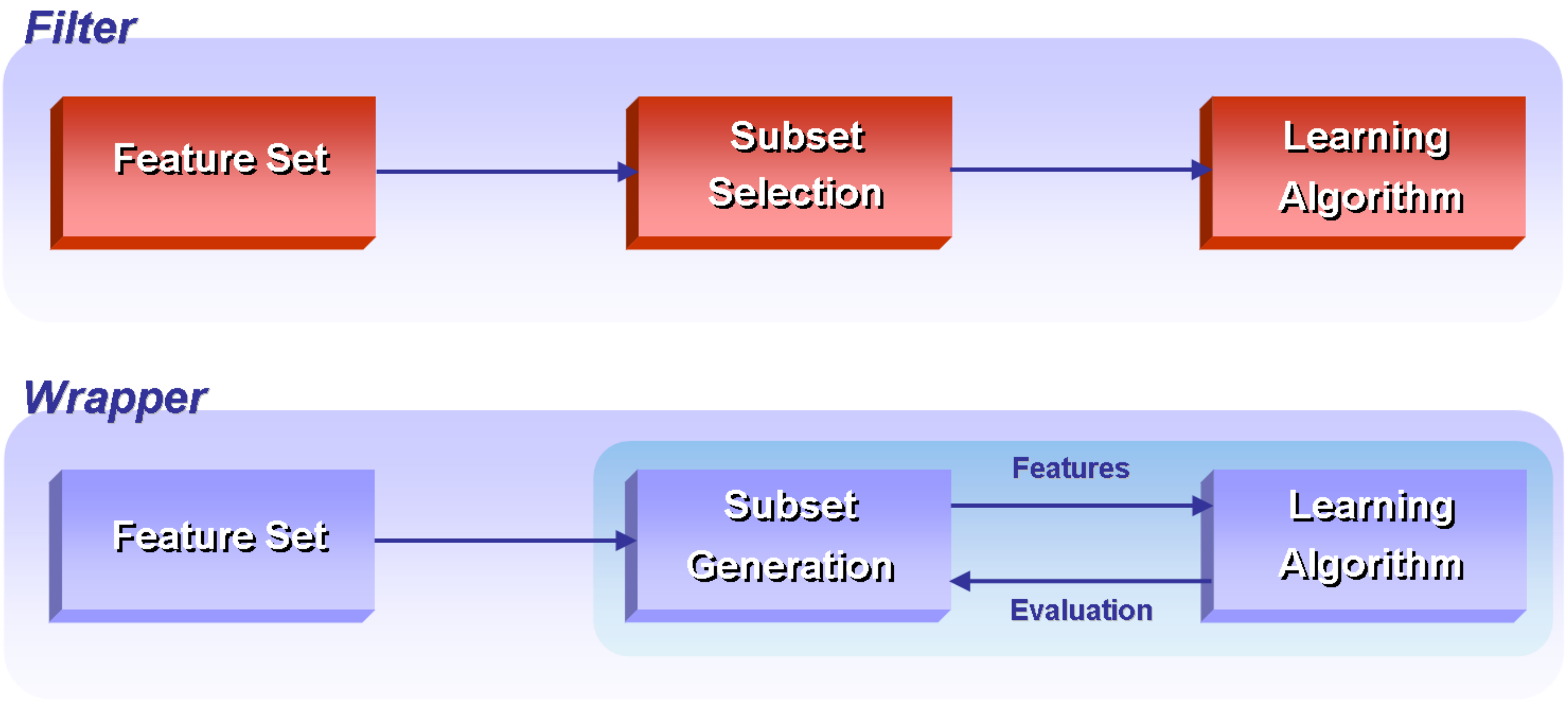
Feature selection process

- Feature selection (FS) preserves data semantics by **selecting** rather than **transforming**



- **Subset generation**: forwards, backwards, random…
- **Evaluation function**: determines 'goodness' of subsets
- **Stopping criterion**: decide when to stop subset search

Types of FS



Filter
Feature Set
Subset Selection
Learning Algorithm
Wrapper
Feature Set
Subset Generation
Features
Learning Algorithm
Evaluation

Search in FS

- We have an evaluation function Eval
  - Task: find a subset of features, F, that maximises Eval(F)
  - Often want to minimise |F|

  - Filter, if Eval = subset evaluation measure
  - Wrapper, if Eval = evaluation via classifier

- Brute-force approach impractical, so search is required
  - Often greedy hill-climbing is used
  - Other search techniques can be used...

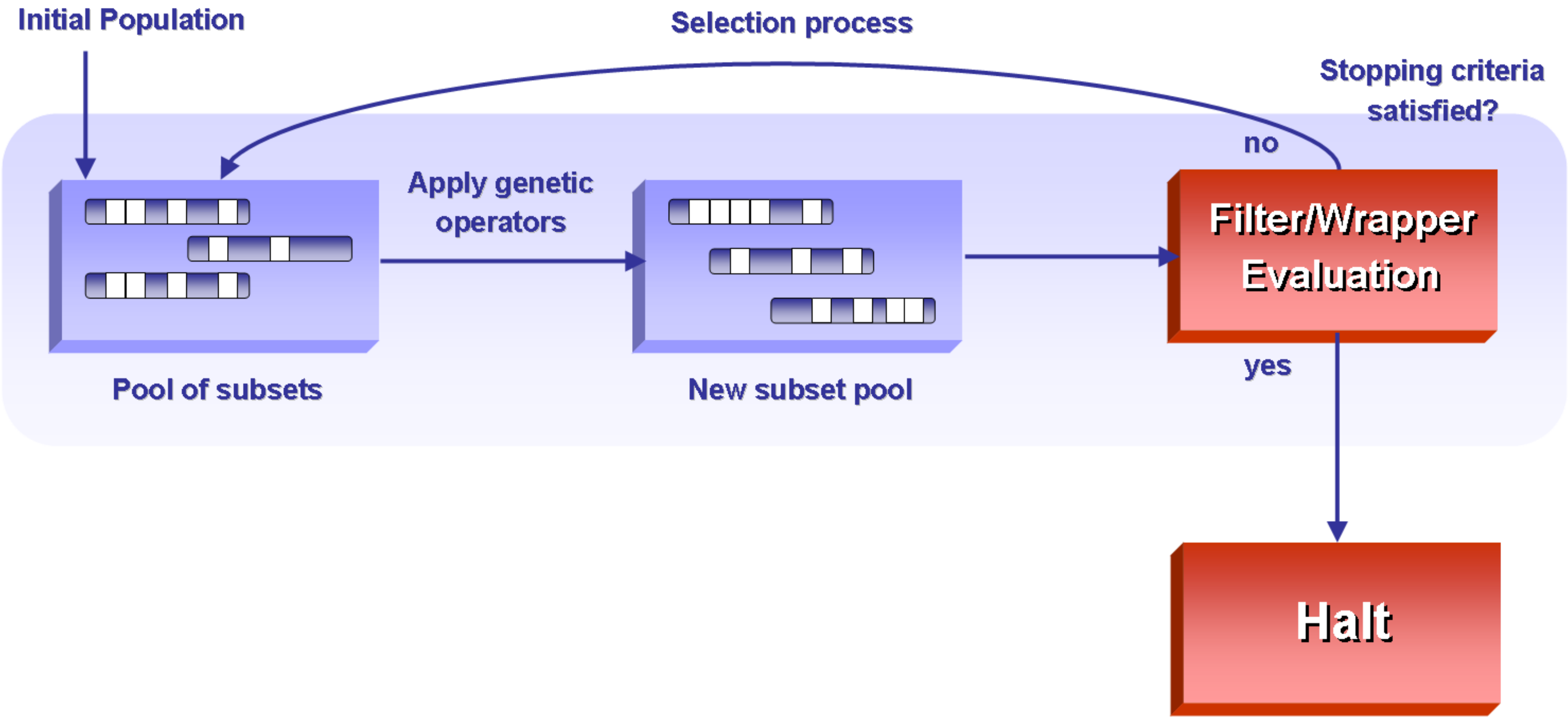
Sequential forward selection

- Algorithm:
  - Begin with zero attributes
  - Evaluate all feature subsets w/ exactly 1 feature
  - Select the one with the best performance
  - Add to this subset the feature that yields the best performance for subsets of next larger size
  - Repeat this until stopping criterion is met

Sequential backward selection

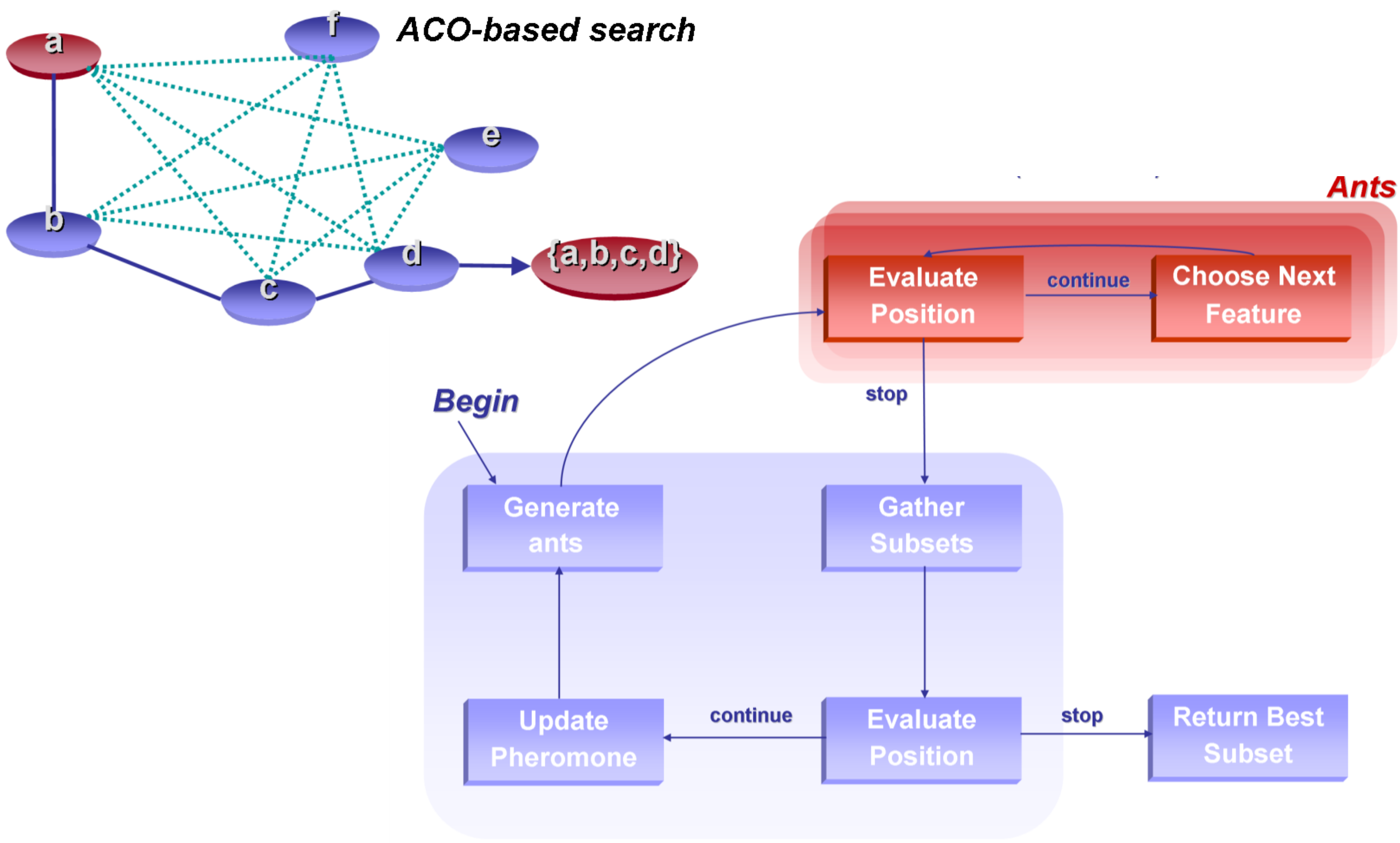
- Begins with all features
- Repeatedly removes a feature whose removal yields the maximal performance improvement
  - Can be filter or wrapper

GA-based FS



Initial Population
Selection process
Stopping criteria
satisfied?
no
Apply genetic
operators
Filter/Wrapper
Evaluation
yes
Pool of subsets
New subset pool
Halt

Ant-based FS



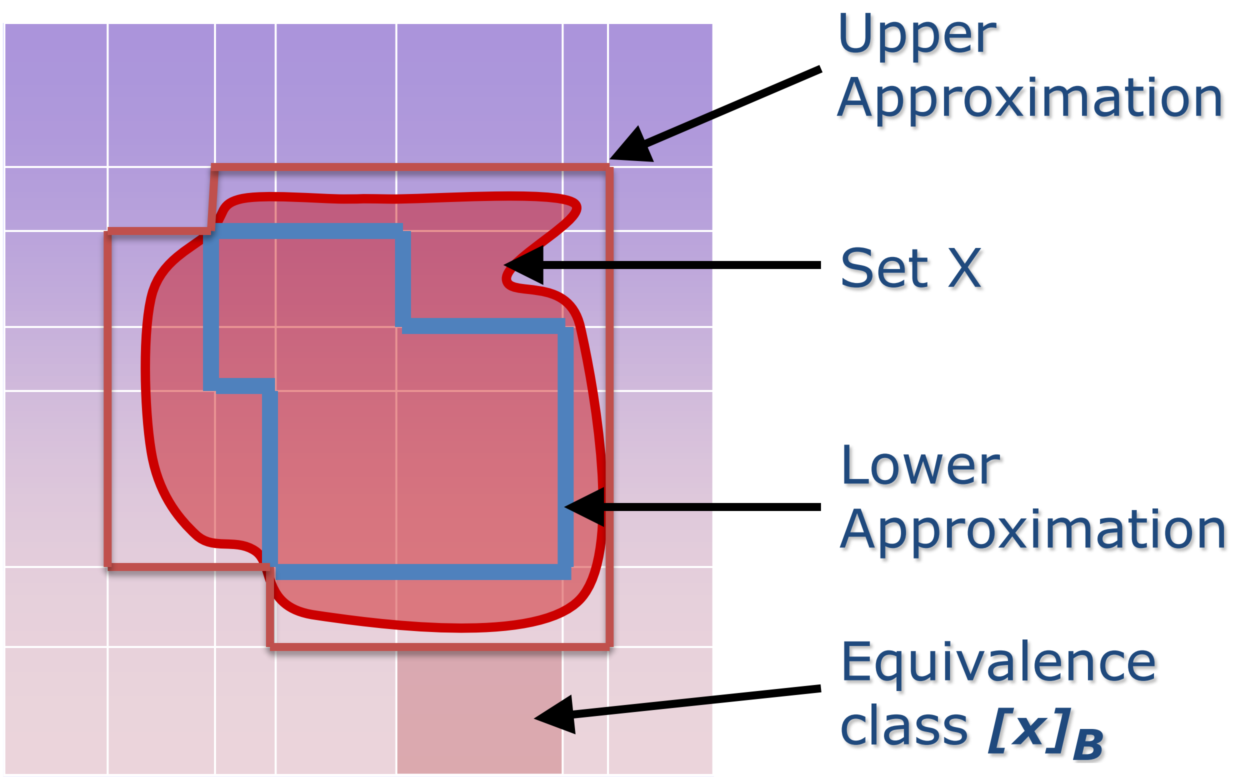
ACO-based search
a
f
e
b
d
c
{a,b,c,d}
Ants
Evaluate Position
continue
Choose Next Feature
stop
Begin
Generate ants
Gather Subsets
Update Pheromone
continue
Evaluate Position
stop
Return Best Subset

Rough set theory

Rough set theory

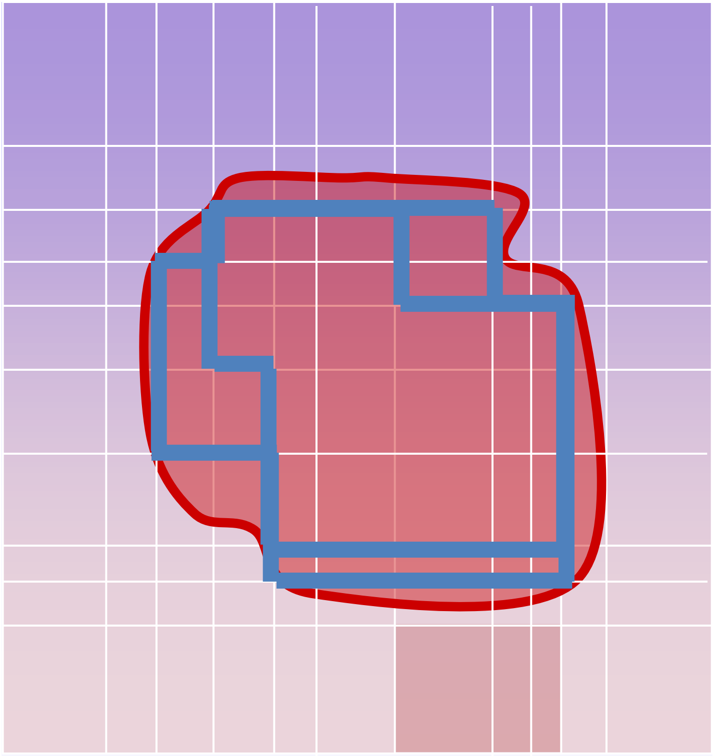
- Invented in 1982, but took off in the 1990s
- Based on set theory
- Approximates (estimates) a concept via two sets:
  - Lower approximation (containing elements that definitely belong to the concept)
  - Upper approximation (containing elements that possibly belong)

Rough set theory



Rough set feature selection

- By considering more features, concepts become easier to define…



Recap: datasets

	Headache	*Muscle pain*	*Temp.*	*Flu*
U1	Yes	Yes	Normal	No
U2	Yes	Yes	High	Yes
U3	Yes	Yes	Very-high	Yes
U4	No	Yes	Normal	No
U5	No	No	High	No
U6	No	Yes	Very-high	Yes

Objects, instances

Each column is a feature/symptom/measurement

The final column is the decision feature

Example dataset

	Headache	*Muscle pain*	*Favourite TV program*	*Temp.*	*Flu*
U1	Yes	Yes	Lost	Normal	No
U2	Yes	Yes	24	High	Yes
U3	Yes	Yes	24	Very-high	Yes
U4	No	Yes	A-Team	Normal	No
U5	No	No	24	High	No
U6	No	Yes	A-Team	Very-high	Yes

Equivalence classes

	Headache	*Muscle pain*	*Favourite TV program*	*Temp.*	*Flu*
U1	**Yes**	**Yes**	**Lost**	**Normal**	**No**
U2	**Yes**	**Yes**	**24**	**High**	**Yes**
U3	**Yes**	**Yes**	**24**	**Very-high**	**Yes**
U4	**No**	**Yes**	**A-Team**	**Normal**	**No**
U5	**No**	**No**	**24**	**High**	**No**
U6	**No**	**Yes**	**A-Team**	**Very-high**	**Yes**

$[U1]_{R_H} = R_{\{H\}}\, U1 = \{U1, U2, U3\}$

$[U5]_{R_H} = R_{\{H\}}\, U5 = \{U4, U5, U6\}$

$R_{\{H\}} = \{\{U1, U2, U3\},\{U4, U5, U6\}\}$

$R_{\{H,M\}} = \{\{U1,U2,U3\}, \{U4,U6\},\{U5\}\}$

$X = \{U1, U2, U3, U4, U5, U6\}$

Rough set theory

- Approximating a concept A using knowledge in feature subset B
  - Lower approximation: contains objects that definitely belong to A

$$R_B{\downarrow}A \quad = \quad \{x \in X \mid [x]_{R_B} \subseteq A\}$$

  - Upper approximation: contains objects that possibly belong to A

$$R_B{\uparrow}A \quad = \quad \{x \in X \mid [x]_{R_B} \cap A \neq \emptyset\}$$

Rough set theory

	Headache	*Muscle pain*	*Temp.*	*Flu*
U1	**Yes**	**Yes**	**Normal**	**No**
U2	**Yes**	**Yes**	**High**	**Yes**
U3	**Yes**	**Yes**	**Very-high**	**Yes**
U4	**No**	**Yes**	**Normal**	**No**
U5	**No**	**No**	**High**	**No**
U6	**No**	**Yes**	**Very-high**	**Yes**

Set B={H,M}
R_B = {{U1,U2,U3}, {U4,U6}, {U5}}

$$R_B{\downarrow}A \quad = \quad \{x \in X | [x]_{R_B} \subseteq A\}$$

For concept Flu=No
A={U1,U4,U5}

$$R_B{\downarrow}A \quad =$$

Rough set feature selection

- Positive region
  - Find the lower approximation for each decision concept
  - Take the union of these
  - Summarises the information contained in a subset for the full dataset

$$POS_B = \bigcup_{x \in X} R_B{\downarrow}[x]_{R_d}$$

- Dependency function
  - Take the cardinality of the positive region (number of elements) and divide by the number of objects in the dataset
  - This is the evaluation measure
  - When this reaches 1, search can stop

$$\gamma_B = \frac{|POS_B|}{|X|}$$

Reducts

Given (X,A∪{d}), B $\subseteq$ A

B is a decision reduct if B satisfies

$$\gamma_B = \gamma_{\mathcal{A}}$$

and no proper subset of B satisfies it

- Core = set of all features that appear in *every* reduct

Rough set feature selection

- One approach: QuickReduct
- Attempts to remove unnecessary or redundant features
  - **Evaluation**: function based on rough set concept of lower approximation
  - **Generation**: greedy hill-climbing algorithm employed
  - **Stopping criterion**: when maximum evaluation value is reached (= reduct or superreduct)

Rough set feature selection

	Headache	*Muscle pain*	*Favourite TV program*	*Temp.*	*Flu*
U1	Yes	Yes	Lost	Normal	No
U2	Yes	Yes	24	High	Yes
U3	Yes	Yes	24	Very-high	Yes
U4	No	Yes	A-Team	Normal	No
U5	No	No	24	High	No
U6	No	Yes	A-Team	Very-high	Yes

consider each feature individually at first

Rough set feature selection

	Headache	*Flu*
U1	Yes	No
U2	Yes	Yes
U3	Yes	Yes
U4	No	No
U5	No	No
U6	No	Yes

Set B={H}
Lower approximation for Flu=No is {}
Lower approximation for Flu=Yes is {}
Positive region is {}

Rough set feature selection

	Temp.	*Flu*
U1	Normal	No
U2	High	Yes
U3	Very-high	Yes
U4	Normal	No
U5	High	No
U6	Very-high	Yes

Set B={T}
Lower approximation for Flu=No is {U1,U4}
Lower approximation for Flu=Yes is {U3,U6}
Positive region is {U1,U3,U4,U6}

Rough set feature selection

	Favourite TV program	*Temp.*	*Flu*
U1	**Lost**	**Normal**	**No**
U2	**24**	**High**	**Yes**
U3	**24**	**Very-high**	**Yes**
U4	**A-Team**	**Normal**	**No**
U5	**24**	**High**	**No**
U6	**A-Team**	**Very-high**	**Yes**

Set B={TV, T}
Lower approximation for Flu=No is {U1,U4}
Lower approximation for Flu=Yes is {U3,U6}
Positive region is {U1,U3,U4,U6} (unchanged)

Rough set feature selection

	Headache	*Temp.*	*Flu*
U1	**Yes**	**Normal**	**No**
U2	**Yes**	**High**	**Yes**
U3	**Yes**	**Very-high**	**Yes**
U4	**No**	**Normal**	**No**
U5	**No**	**High**	**No**
U6	**No**	**Very-high**	**Yes**

Set B={H,T}
Lower approximation for Flu=No is {U1,U4,U5}
Lower approximation for Flu=Yes is {U2,U3,U6}
Positive region is {U1,U2,U3,U4,U5,U6} = X

Example 2

	Diploma	*Experience*	*French*	*Reference*	*Decision*
x_1	MBA	Medium	Yes	Excellent	Accept
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_4	MSc	High	Yes	Neutral	Accept
x_5	MSc	Medium	Yes	Neutral	Reject
x_6	MSc	High	Yes	Excellent	Accept
x_7	MBA	High	No	Good	Accept
x_8	MCE	Low	No	Excellent	Reject

$$R_B{\downarrow}A \quad = \quad \{x \in X | [x]_{R_B} \subseteq A\}$$

Fuzzy-rough feature selection

Fuzzy-rough set theory

- Problems:
  - Rough set methods (usually) require data discretization beforehand
  - Also no flexibility in approximations

- Example
  - Objects either belong fully to the lower (or upper) approximation, or not at all

Hybridizing rough and fuzzy sets

- Two lines of thought in hybridization
  - **Axiomatic** approach – investigate mathematical properties
  - **Constructive** approach – generalize lower and upper approximations

- The fuzzy-rough tools described in this tutorial are built on the definition in:
  - *A.M. Radzikowska, E.E. Kerre, A comparative study of fuzzy rough sets, Fuzzy Sets and Systems, vol. 126, no. 2, pp. 137-155, 2002.*

Fuzzy rough sets

Rough set

$$R_B\uparrow A = \{x \in X \mid \boxed{[x]_{R_B} \cap A \neq \emptyset}\}$$

$$R_B\downarrow A = \{x \in X \mid \boxed{[x]_{R_B} \subseteq A}\}$$

Fuzzy-rough set

t-norm

$$R\uparrow A(y) = \sup_{x \in X} \mathcal{T}(R(x,y), A(x))$$

$$R\downarrow A(y) = \inf_{x \in X} \mathcal{I}(R(x,y), A(x))$$

implicator

FRFS (old)

- Based on Dubois and Prade's definitions
  - Fuzzy lower approximation:

$$\mu_{\underline{P}X}(x) = \sup_{F \in U/P} \min(\mu_F(x), \inf_{y \in U} I(\mu_F(y), \mu_X(y)))$$

  - Fuzzy positive region:

$$\mu_{POS_P(Q)}(x) = \sup_{X \in U/Q} \mu_{\underline{P}X}(x)$$

  - Evaluation function:

$$\gamma'_P(Q) = \frac{|\mu_{POS_P(Q)}(x)|}{|U|} = \frac{\sum_{x \in U} \mu_{POS_P(Q)}(x)}{|U|}$$

FRFS (new)

- Based on fuzzy similarity

$$R_a(x,y) = 1 - \frac{|a(x) - a(y)|}{|a_{\max} - a_{\min}|}$$

$$R_P(x,y) = \bigcap_{a \in P} \{R_a(x,y)\}$$

- Lower/upper approximations

$$\begin{aligned} R{\uparrow}A(y) &= \sup_{x \in X} \mathcal{T}(R(x,y), A(x)) \\ R{\downarrow}A(y) &= \inf_{x \in X} \mathcal{I}(R(x,y), A(x)) \end{aligned}$$

FRFS: evaluation function

- Fuzzy positive region #1

$$POS_B(y) = \left(\bigcup_{x \in X} R_B \downarrow R_d x \right)(y)$$

- Fuzzy positive region #2 (weak)

$$POS_B(y) = (R_B \downarrow R_d y)(y)$$

- Dependency function

$$\gamma_B = \frac{|POS_B|}{|POS_{\mathcal{A}}|}$$

FRFS: evaluation function

- Alternative measure: delta function

$$\delta_B = \frac{\min_{x \in X} POS_B(x)}{\min_{x \in X} POS_{\mathcal{A}}(x)}$$

- Properties

Proposition 2: For subsets B_1, B_2 of $\mathcal{A}$,

$$B_1 \subseteq B_2 \Rightarrow \begin{cases} \gamma_{B_1} \leq \gamma_{B_2} \\ \delta_{B_1} \leq \delta_{B_2} \end{cases}$$

Proposition 3: $\gamma_{\mathcal{A}} = \delta_{\mathcal{A}} = 1$

FRFS: finding reducts

- Fuzzy-rough QuickReduct
  - **Evaluation**: use the dependency function (or other fuzzy-rough measure)
  - **Generation**: greedy hill-climbing
  - **Stopping criterion**: when maximal evaluation function is reached (or to degree α)

FRFS example

Object	a	b	c	q
1	-0.4	-0.3	-0.5	no
2	-0.4	0.2	-0.1	yes
3	-0.3	-0.4	-0.3	no
4	0.3	-0.3	0	yes
5	0.2	-0.3	0	yes
6	0.2	0	0	no

$$R_a(x,y) = \max\left(\min\left(\frac{a(y)-a(x)+\sigma_a}{\sigma_a}, \frac{a(x)-a(y)+\sigma_a}{\sigma_a}\right), 0\right)$$

FRFS example

- If A={1,3,6}, calculate $(R_a{\downarrow}A)(3)$

$$
\begin{array}{cc}
 & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
R_a(x,y) = & \begin{pmatrix}
1.0 & 1.0 & 0.699 & 0.0 & 0.0 & 0.0 \\
1.0 & 1.0 & 0.699 & 0.0 & 0.0 & 0.0 \\
0.699 & 0.699 & 1.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 1.0 & 0.699 & 0.699 \\
0.0 & 0.0 & 0.0 & 0.699 & 1.0 & 1.0 \\
0.0 & 0.0 & 0.0 & 0.699 & 1.0 & 1.0
\end{pmatrix}
\end{array}
$$

$$A(x) = (1,0,1,0,0,1)$$

$$\mathcal{I}_{\mathcal{S}_{\mathrm{W}}}(x,y) = \min(1-x+y,1)$$

$$(R{\downarrow}A)(y) = \inf_{x \in X} \mathcal{I}(R(x,y), A(x))$$

FRFS example

$$\begin{aligned}(R{\downarrow}A)(y) &= \inf_{x\in X} \mathcal{I}(R(x,y), A(x)) \\ &= \inf\{I(0.699,1), I(0.699,0), I(1,1), \\ &\qquad I(0,0), I(0,0), I(0,1)\} \\ &= 0.301\end{aligned}$$

$$\begin{aligned}\gamma_{\{a\}} &= \frac{0.602}{6} \\ &= 0.1003\end{aligned}$$

FRFS example

- The search continues...

$$\gamma_{\{b\}} = 0.3597 \qquad \gamma_{\{c\}} = 0.4078$$

- Feature *c* looks more promising, so choose this and continue searching

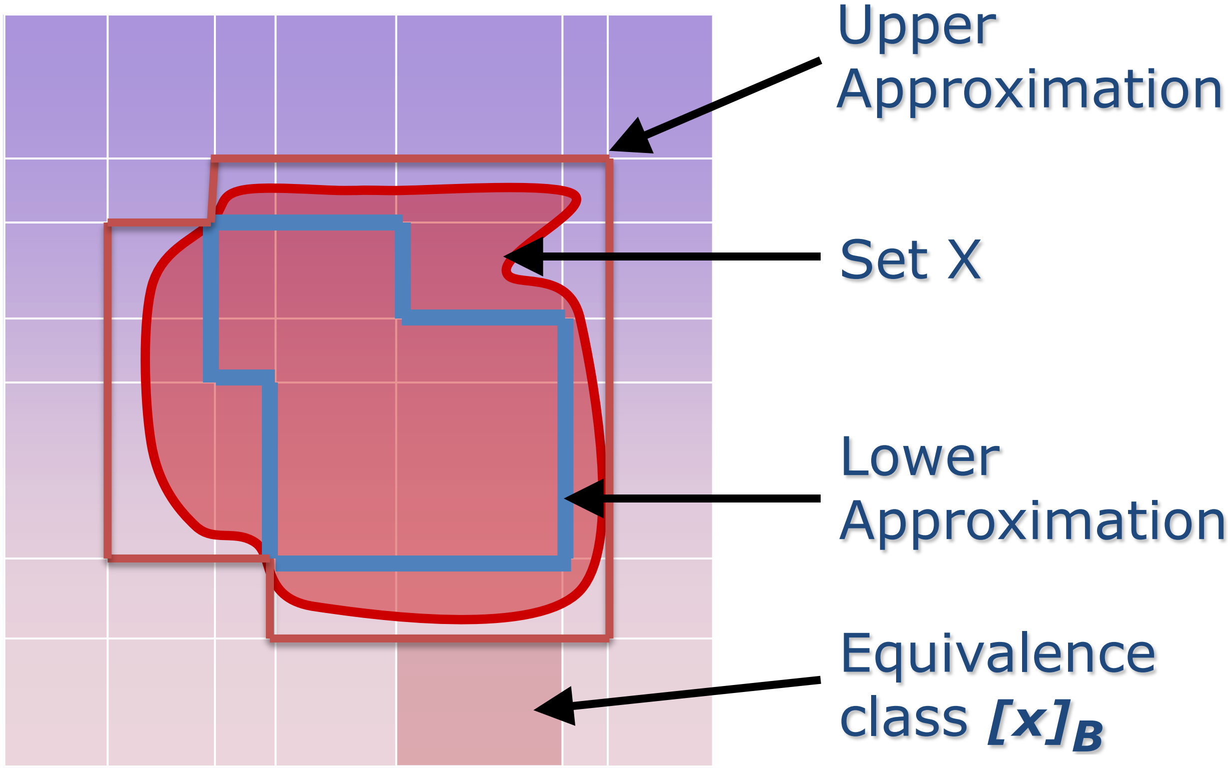
$$\gamma_{\{a,c\}} = 0.5501 \qquad \gamma_{\{b,c\}} = 1.0$$

FRFS

- Other subset generation methods
  - GAs
  - ACO
  - Backward elimination

- Other subset evaluations
  - Fuzzy boundary region
  - Fuzzy entropy
  - Fuzzy discernibility functions

Boundary region



FRFS: boundary region

- Fuzzy lower and upper approximation define fuzzy boundary region

$$R{\uparrow}A(y) - R{\downarrow}A(y)$$

- For each concept, minimise the boundary region
  - (also applicable to crisp RSFS)
- Results seem to show this is a more informed heuristic (but slower to calculate)

FRFS: issues

- Problem – noise tolerance!

$$R{\uparrow}A(y) = \boxed{\sup_{x \in X}} \mathcal{T}(R(x,y), A(x))$$

$$R{\downarrow}A(y) = \boxed{\inf_{x \in X}} \mathcal{I}(R(x,y), A(x))$$

Vaguely quantified rough sets

Pawlak rough set

y belongs to the lower approximation of A iff
all elements of Ry belong to A

y belongs to the upper approximation of A iff
at least one element of Ry belongs to A

VQRS

y belongs to the lower approximation of A iff
most elements of Ry belong to A

y belongs to the upper approximation of A iff
at least some elements of Ry belong to A

VQRS

$$R{\uparrow}_{Q_l}A(y) \;=\; Q_l\left(\frac{|Ry \cap A|}{|Ry|}\right)$$

$$R{\downarrow}_{Q_u}A(y) \;=\; Q_u\left(\frac{|Ry \cap A|}{|Ry|}\right)$$

R, A: crisp *or* fuzzy
R↓A, R↑A: fuzzy

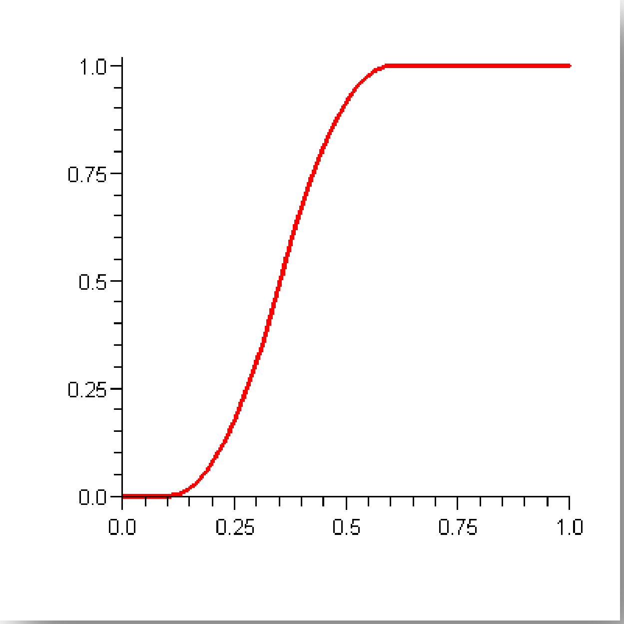
y belongs to the lower approximation *to the extent that* **most elements** of Ry belong to A

y belongs to the upper approximation *to the extent that* **some elements of** Ry belongs to A

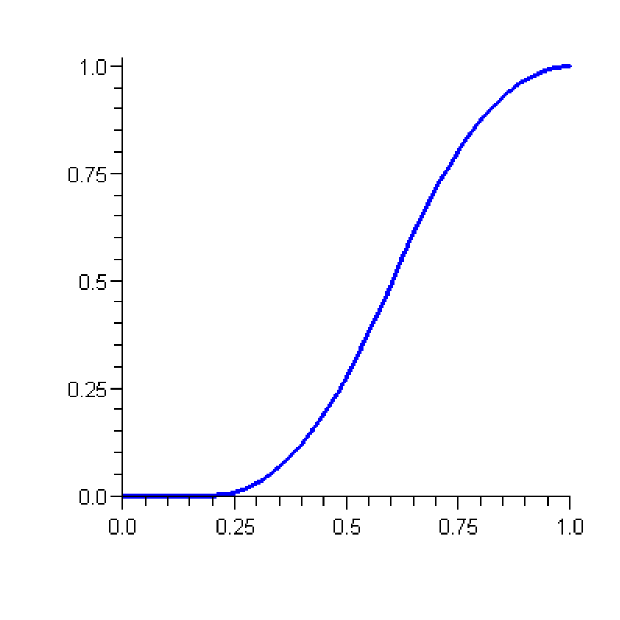
Fuzzy quantifiers: examples

Fuzzy quantifier (Zadeh):
[0,1]→[0,1] mapping Q

Q_l: some



Q_u: most

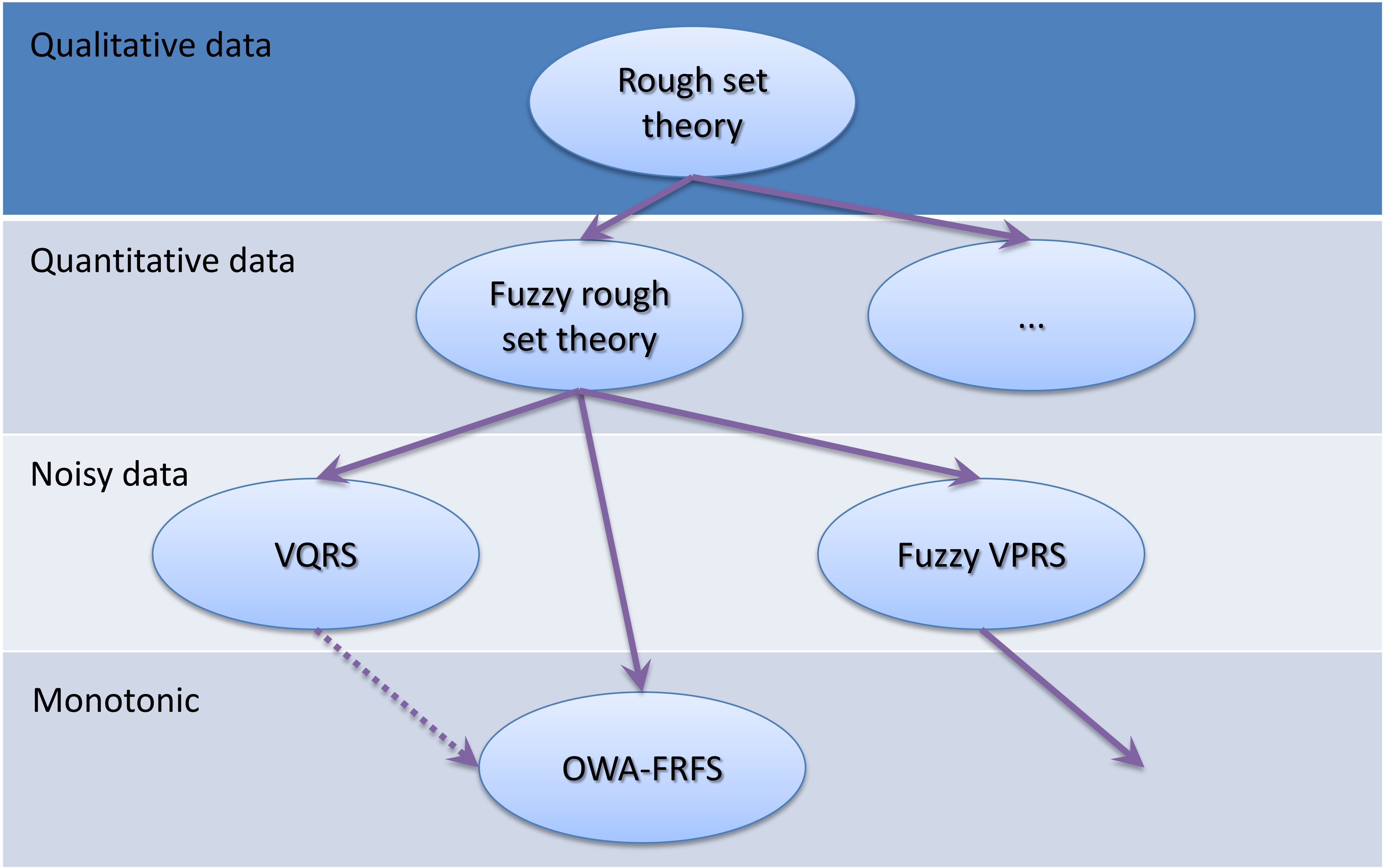


VQRS-based feature selection

- Use the quantified lower approximation, positive region and dependency degree
  - **Evaluation**: the quantified dependency (can be crisp or fuzzy)
  - **Generation**: greedy hill-climbing
  - **Stopping criterion**: when the quantified positive region is maximal (or to degree α)

- Should be more noise-tolerant, but is non-monotonic

Progress


Qualitative data
Rough set theory
Quantitative data
Fuzzy rough set theory
...
Noisy data
VQRS
Fuzzy VPRS
Monotonic
OWA-FRFS

OWA-FRFS

- E.g. values {3, 5, 2, 7, 4}
  - (Ordered values = 7 5 4 3 2)
  - OWA modelling of sup and inf:
    - sup-weights 1 0 0 0 0 = 7
    - inf-weights 0 0 0 0 1 = 2
  - OWA relaxation of sup and inf:
    - sup-relax-weights 0.7 0.2 0.1 0.0 0.0 = 6.3
    - inf-relax-weights 0.0 0.0 0.1 0.2 0.7 = 2.4

OWA-FRFS

- New lower and upper approximations

$$(R{\downarrow}_{W_l}A)(y) = OWA_{W_l}\langle \mathcal{I}(R(x_i,y),A(x_i))\rangle$$
$$(R{\uparrow}_{W_u}A)(y) = OWA_{W_u}\langle \mathcal{T}(R(x_i,y),A(x_i))\rangle$$

- Feature selectors can be built on this

- Vague quantifiers can be modelled with this

More issues...

- Problem #1: how to choose fuzzy similarity?
- Problem #2: how to handle missing values?

Interval-valued FRFS

- Answer #1: Model uncertainty in fuzzy similarity by interval-valued similarity

IV fuzzy rough set

$$\mu_{\underline{\widetilde{R_P X}}}(x) = \inf_{y \in \mathbb{U}} \mathcal{I}(\mu_{\widetilde{R_P}}(x,y), \mu_{\widetilde{X}}(y))$$

$$\mu_{\overline{\widetilde{R_P X}}}(x) = \sup_{y \in \mathbb{U}} \mathcal{T}(\mu_{\widetilde{R_P}}(x,y), \mu_{\widetilde{X}}(y))$$

IV fuzzy similarity

$$\mu_{R_{a*}}(x,y) = 1 - \left(\frac{|a(x) - a(y)|}{|a_{\max} - a_{\min}|}\right)^m$$

$$\mu_{R_a^*}(x,y) = 1 - \frac{|a(x) - a(y)|}{|a_{\max} - a_{\min}|}$$

Interval-valued FRFS

- When comparing two object values for a given attribute – what to do if at least one is missing?
- Answer #2: Model missing values via the unit interval

$$\mu_{\widetilde{R_a}}(x,y) = \begin{cases} \mu_{\widetilde{R_a}}(x,y) & \text{if } a(x), a(y) \neq *, \\ [0,1] & \text{otherwise} \end{cases}$$

Other measures

- Boundary region

$$\mu_{\widetilde{BND_P(X)}}(x) \quad = \quad \mu_{\widetilde{\overline{R_P}X}}(x) - \mu_{\widetilde{\underline{R_P}X}}(x)$$

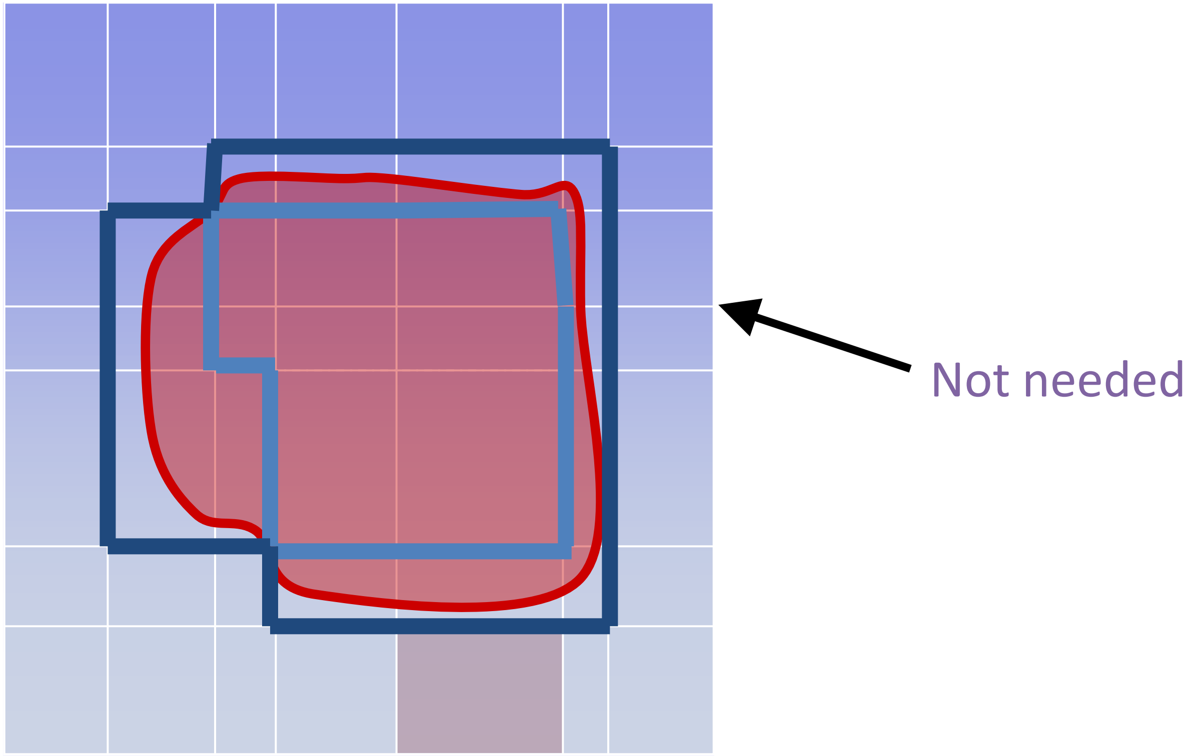
- Discernibility function

$$\widetilde{c_{ij}}(P) = \mathcal{I}(\underbrace{\mathcal{T}(\mu_{\widetilde{R_a}}(x_i, x_j))}_{a \in P}, \mu_{\widetilde{R_{\mathbb{D}}}}(x_i, x_j))$$

$$\widetilde{g}(P) = \frac{2. \sum\limits_{1 \leq i < j \leq n} \widetilde{c_{ij}}(P)}{n(n-1)}$$

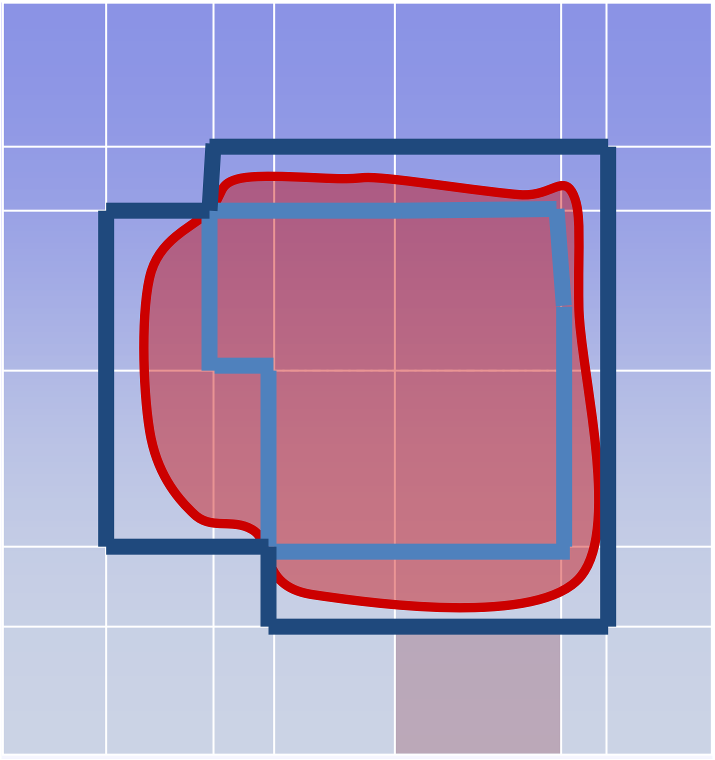
Instance Selection

Instance selection: basic idea



Remove objects to keep the underlying approximations unchanged, or to improve them

Instance selection: basic idea



Remove objects to keep the underlying approximations unchanged, or to improve them

Fuzzy-rough sets

- Parameterized relation

$$R_a^\alpha(x,y) = \max\left(0, 1 - \alpha\frac{|a(x)-a(y)|}{l(a)}\right)$$

$$R_B^\alpha(x,y) = \mathcal{T}(\underbrace{R_a^\alpha(x,y)}_{a\in B})$$

- Fuzzy-rough definitions:

$$(R_B^\alpha{\downarrow}^S A)(y) \quad = \quad \inf_{x\in S} \mathcal{I}(R_B^\alpha(x,y), A(x))$$

$$POS_B^{\alpha,S}(y) = (R_B^\alpha{\downarrow}^S R_d^\alpha y)(y)$$

$$\gamma_B^{\alpha,S} = \frac{\sum_{y\in S} POS_B^{\alpha,S}(y)}{|S|}$$

FRIS-I

FRIS-I(S,α,τ).
S, the set of objects to be reduced;
α, the granularity parameter;
τ, a selection threshold.

(1) $Y \leftarrow S$
(2) **foreach** $x \in S$
(3) **if** $(POS_{\mathcal{A}}^{\alpha,S}(x) < \tau)$
(4) $Y \leftarrow Y - \{x\}$
(5) **return** Y

FRIS-II

FRIS-II(S,α).
S, the set of objects to be reduced;
α, the granularity parameter.

while (`true`)
    $z \leftarrow \emptyset$, $\rho_z \leftarrow 1$
    foreach $x \in S$
        if ($POS_{\mathcal{A}}^{\alpha,S}(x) < \rho_z$)
            $z \leftarrow x$
            $\rho_z \leftarrow POS_{\mathcal{A}}^{\alpha,S}(x)$
    if ($z \neq \emptyset$)
        $S \leftarrow S - \{z\}$
    else return S

FRIS-III

FRIS-III(S,α).
S, the set of objects to be reduced;
α, the granularity parameter.

(1) $\rho \leftarrow \gamma_{\mathcal{A}}^{\alpha,S}$
(2) **while** $(\rho \neq 1)$
(3) $z \leftarrow \emptyset,\ \rho_z \leftarrow 0$
(4) **foreach** $x \in S$
(5) **if** $(\gamma_{\mathcal{A}}^{\alpha,S-\{x\}} > \rho_z)$
(6) $z \leftarrow x,\ \rho_z \leftarrow \gamma_{\mathcal{A}}^{\alpha,S-\{x\}}$
(7) $S \leftarrow S - z$
(8) $\rho \leftarrow \rho_z$
(9) **return** S

Fuzzy-rough classification and prediction

Nearest neighbour algorithm

- 1-NN:

  Given a test instance x_m,

  - First locate the nearest training example x_n
  - Then $f(x_m) := f(x_n)$

- *k*-NN:

  Given a test instance x_m,

  - First locate the *k* nearest training examples
  - **If** target function = discrete **then** take vote among its *k* nearest neighbours
    else take the mean of the *f* values of the *k* nearest neighbours (prediction)

Fuzzy NN

Input: X, the training data; $\mathcal{C}$, the set of decision classes; y, the object to be classified; K, the number of nearest neighbours

Output: Classification for y

begin

  $N \leftarrow$ getNearestNeighbours(y,K)

  foreach $C \in \mathcal{C}$ **do**

    $C'(y) = \sum_{x \in N} R(x, y)C(x)$

  end

  output $\underset{C \in \mathcal{C}}{\arg\max}\,(C'(y))$

end

Fuzzy-rough NN

Input: X, the training data; $\mathcal{C}$, the set of decision classes; y, the object to be classified

Output: Classification for y

begin

$N \leftarrow \text{getNearestNeighbours}(y, K)$

$\tau \leftarrow 0$, $Class \leftarrow \emptyset$

foreach $C \in \mathcal{C}$ **do**

if $((R{\downarrow}C)(y) + (R{\uparrow}C)(y))/2 \geq \tau$ **then**

$Class \leftarrow C$

$\tau \leftarrow ((R{\downarrow}C)(y) + (R{\uparrow}C)(y))/2$

end

end

output $Class$

end

Fuzzy-rough NN

Input: X, the training data; d, the decision feature; y, the object for which to find a prediction

Output: Classification for y

begin

$N \leftarrow$ getNearestNeighbours(y,K)

$\tau_1 \leftarrow 0$, $\tau_2 \leftarrow 0$

foreach $z \in N$ **do**

$M \leftarrow ((R{\downarrow}R_d z)(y) + (R{\uparrow}R_d z)(y))/2$

$\tau_1 \leftarrow \tau_1 + M * d(z)$

$\tau_2 \leftarrow \tau_2 + M$

end

if $\tau_2 > 0$ **then**

output τ_1/τ_2

else

output $\sum_{z \in N} d(z)/|N|$

end

end

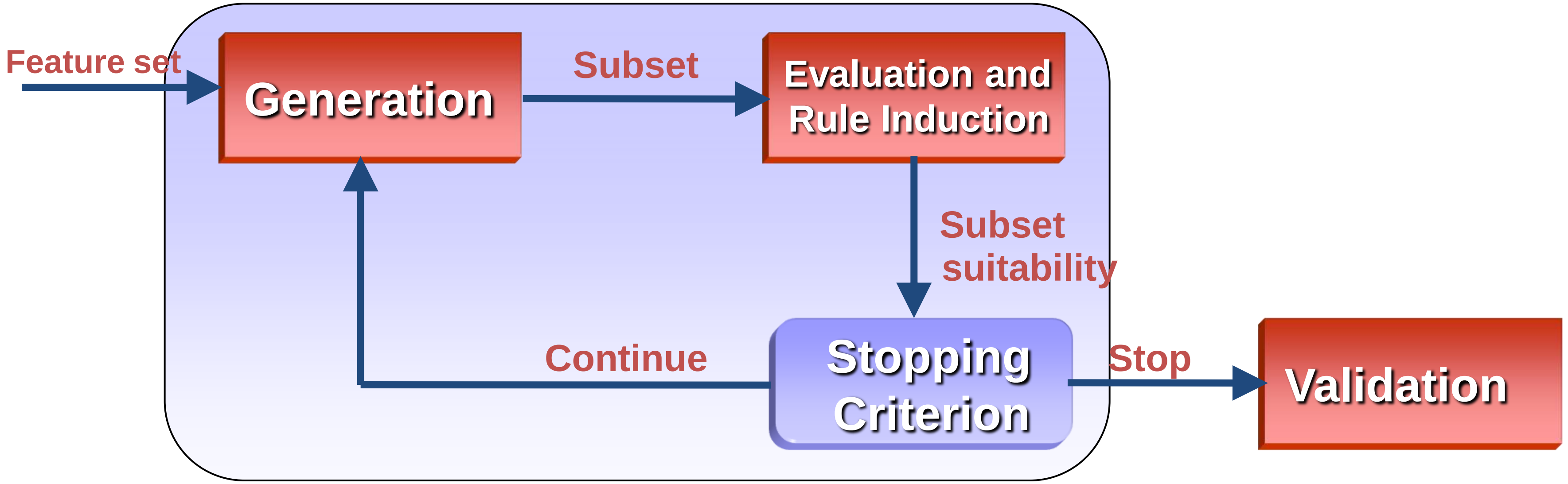
Discovering rules via RST

- Equivalence classes
  - Form the antecedent part of a rule
  - The lower approximation tells us if this is predictive of a given concept (*certain* rules)

- Typically done in one of two ways:
  - Overlaying reducts
  - Building rules by considering individual equivalence classes (e.g. LEM2)

Framework

- The fuzzy tolerance classes used during this process can be used to create fuzzy rules



QuickRules

$B := \{\}, Rules := \{\}, Cov := \{\}$
do
    $T := B$
    foreach $a \in (\mathcal{A} \setminus B)$
        foreach $y \in X \setminus covered(Cov)$
            if $POS_{B \cup \{a\}}(y) = POS_{\mathcal{A}}(y)$
                $\textsc{Check}(B \cup \{a\}, R_{B \cup \{a\}}y, R_d y)$
        if $\gamma_{B \cup \{a\}} > \gamma_T$
            $T := B \cup \{a\}$
    $B := T$
until $\gamma_B = \gamma_{\mathcal{A}}$
return $B, Rules$

Check

CHECK(B, C, D).

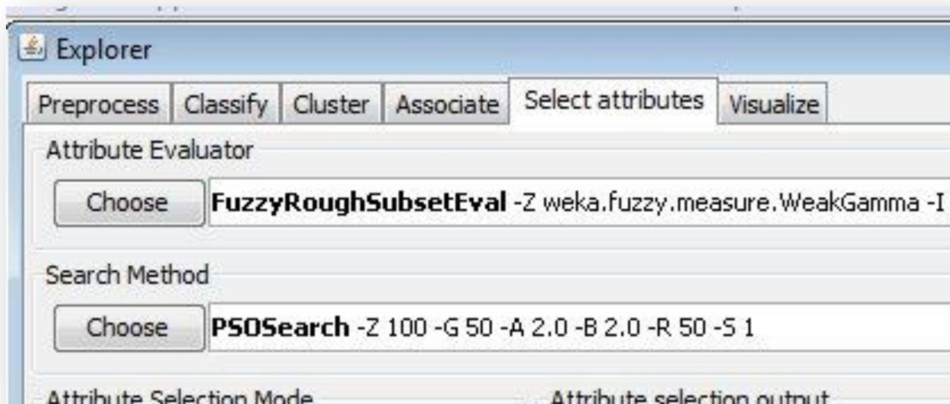
$Add := true$
foreach $Rule \in Rules$
    if $C \subseteq Rule.C$
        Add := false; **break**
    elseif $Rule.C \subset C$
        $Rules := Rules \setminus Rule$
if Add = true
    $Rules := Rules \cup (B, C, D)$
    $Cov := Cov \cup C$
return

Weka

Try the algorithms!

FR methods in Weka

- Weka implementations of all fuzzy-rough methods can be downloaded from:



http://users.aber.ac.uk/rkj/book/wekafull.jar

Other developments

- Fuzzy Discernibility Matrices for FRFS
  - Extends the DMs for crisp rough set feature selection
  - Also employs similar simplification schemes

- Fuzzy-rough semi-supervised learning
  - For mixtures of labelled and unlabelled data

Papers

- Fuzzy-rough feature selection
  - R. Jensen and Q. Shen. **New Approaches to Fuzzy-Rough Feature Selection**. IEEE Transactions on Fuzzy Systems, vol. 17, no. 4, pp. 824-838, 2009.
  - R. Jensen and Q. Shen. **Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches.** IEEE Press/Wiley & Sons, 2008.
  - C. Cornelis, R. Jensen, G. Hurtado Martin, D. Slezak. **Attribute Selection with Fuzzy Decision Reducts.** Information Sciences, vol. 180, no. 2, pp. 209-224, 2010.
  - G.C.Y. Tsang, D. Chen, E.C.C. Tsang, J.W.T. Lee, and D.S. Yeung. **On attributes reduction with fuzzy rough sets.** Proc. 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2775–2780, 2005.
  - X.Z. Wang, Y. Ha, and D. Chen. **On the reduction of fuzzy rough sets.** Proc. 2005 International Conference on Machine Learning and Cybernetics, vol. 5, pp. 3174–3178, 2005.
  - Q. Hu, D. Yu, and Z. Xie. **Information-preserving hybrid data reduction based on fuzzy-rough techniques**. Pattern Recognition Letters, vol. 27, no. 5, pp. 414–423, 2006.
  - Q. Hu, P. Zhu, J. Liu, Y. Yang, D. Yu. **Feature Selection via Maximizing Fuzzy Dependency.** Fundamenta Informaticae, vol. 98 (2-3): 167-181, 2010.
  - E.C.C. Tsang, D. Chen, D.S. Yeung, X. Wang, J. Lee. **Attributes Reduction Using Fuzzy Rough Sets.** IEEE T. Fuzzy Systems, vol. 16, no. 5, pp. 1130-1141, 2008.
  - ...

Papers

- FRFS extensions
  - R. Jensen and Q. Shen. **Interval-valued Fuzzy-Rough Feature Selection in Datasets with Missing Values.** Proceedings of the 18th International Conference on Fuzzy Systems (FUZZ-IEEE'09), pp. 610-615, 2009.
  - C. Cornelis, N. Verbiest and R. Jensen. **Ordered Weighted Average Based Fuzzy Rough Sets.** Proceedings of the 5th International Conference on Rough Sets and Knowledge Technology (RSKT2010), pp. 78-85, 2010.
  - C. Cornelis and R. Jensen. **A Noise-tolerant Approach to Fuzzy-Rough Feature Selection.** Proceedings of the 17th International Conference on Fuzzy Systems (FUZZ-IEEE'08), pp. 1598-1605, 2008.

Papers

- FR instance selection
  - R. Jensen and C. Cornelis. **Fuzzy-rough instance selection.** Proceedings of the 19th International Conference on Fuzzy Systems (FUZZ-IEEE'10), pp. 1776-1782, 2010.

- FR classification/prediction
  - R. Jensen and C. Cornelis. **A New Approach to Fuzzy-Rough Nearest Neighbour Classification.** Transactions on Rough Sets XIII, LNCS 6499, pp. 56-72, 2011.
  - R. Jensen, C. Cornelis and Q. Shen. **Hybrid Fuzzy-Rough Rule Induction and Feature Selection.** Proceedings of the 18th International Conference on Fuzzy Systems (FUZZ-IEEE'09), pp. 1151-1156, 2009.
  - R. Jensen and Q. Shen**. Fuzzy-Rough Feature Significance for Fuzzy Decision Trees.** Proceedings of the 2005 UK Workshop on Computational Intelligence, pp. 89-96. 2005.