WORKSHOP:

KOWLEDGE EXTRACTION BASED ON EVOLUTIONARY ALGORITHMS

15 - 16 May, 2008

Granada, Spain

GP-COACH:

A Novel Genetic Cooperative-Competitive Fuzzy Rule Based Learning Method using Genetic Programming for High-Dimensional Problems

Francisco José Berlanga

Dept. of Computer Science, University of Jaén, Spain

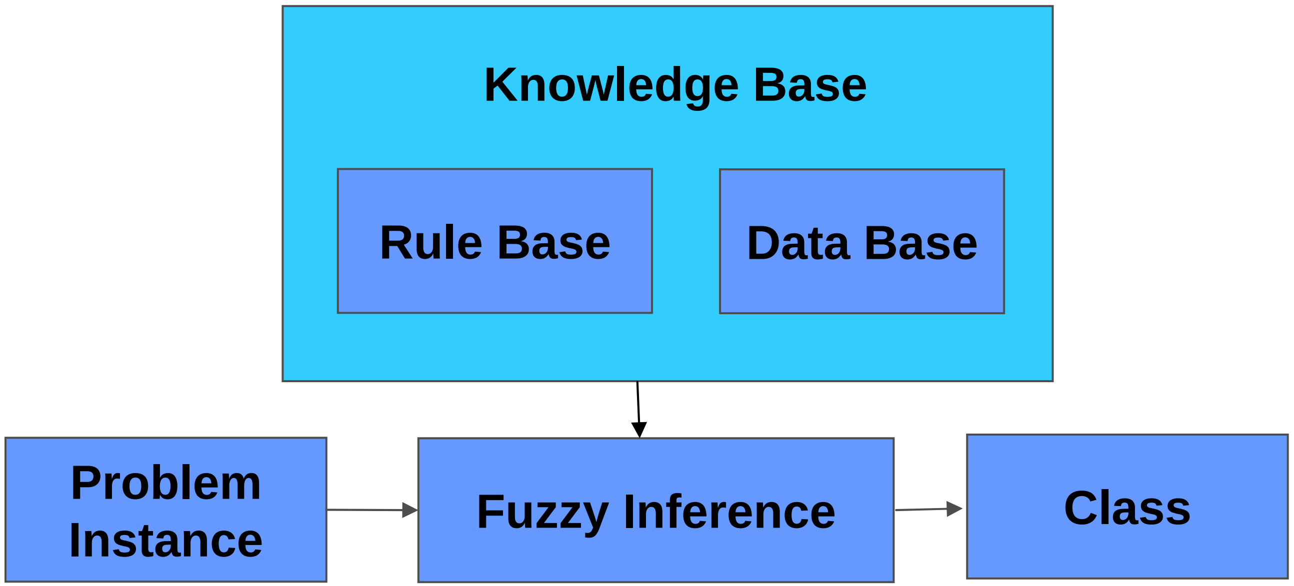
In collaboration with

María José del Jesus and Francisco Herrera

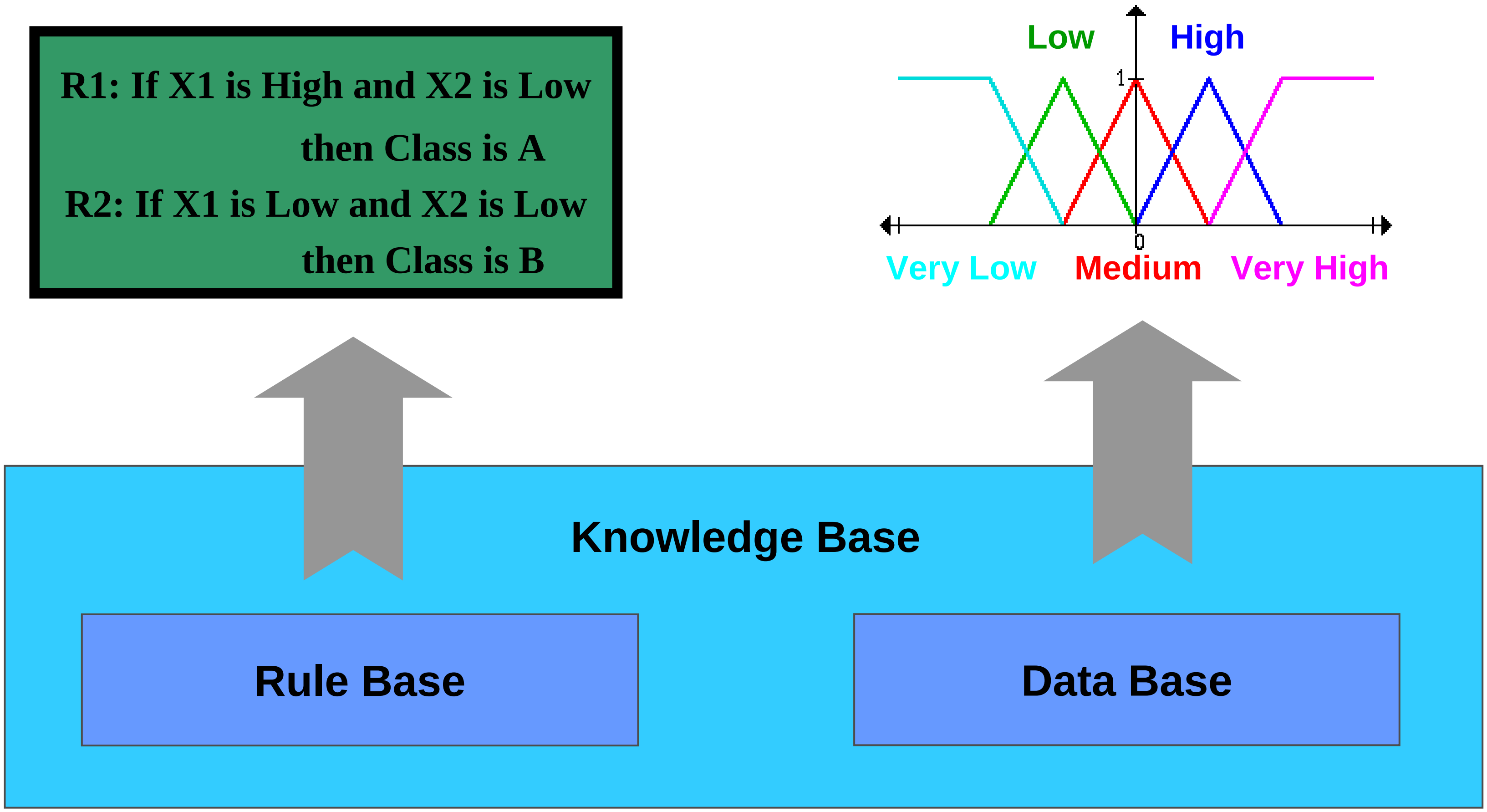
Index

- Introduction
- GP-COACH Algorithm
- Experimental Framework and Results
- Conclusions

Fuzzy Rule-Based Classification System (FRBCS)



Fuzzy Rule-Based Classification System (FRBCS)



Introduction

Learning of FRBCSs for high-dimensional problems

- Number of training examples
- Number of input features

Number of rules

Interpretability

Learning of FRBCSs for high-dimensional problems

1. **Carry out a feature selection (*a priori* or *embedded*) process:**
   - Using information measures
   - Using GFSs
   - Neuro-fuzzy systems
   - Other different techniques

2. **Compact and reduce a previosly learned rule set in a postprocesing stage:**
   - Using GFSs
   - Orthogonal transformation methods
   - Similarity measures

Introduction

FRBSs learning. Carrying out a feature selection process (I):

- Using information measures
  - R. Bhatt and M. Gopal, On fuzzy-rough sets approach to feature selection, Pattern Recognition Letters, vol. 26, no. 7, pp. 965–975, 2005
  - Q. Shen and R. Jensen, Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring, Pattern Recognition, vol. 37, no. 7, pp. 1351–1363, 2004
  - R. Silipo and M. Berthold, Input features' impact on fuzzy decision processes, IEEE Trans. Syst., Man, Cybern. B, vol. 30, no. 6, pp. 821–834, 2000
- Using GFSs
  - J. Casillas, O. Cordón, M. del Jesus, and F. Herrera, Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems, Information Sciences, vol. 136, no. 1-4, pp. 135–157, 2001
  - A. González and R. Pérez, Selection of relevant features in a fuzzy genetic learning algorithm, IEEE Trans. Syst., Man, Cybern. B, vol. 31, no. 3, pp. 417–425, 2001
  - S. Yu, S. D. Backer, and P. Scheunders, Genetic feature selection combined with composite fuzzy nearest neighbor classifiers for hyperspectral satellite imagery, Pattern Recognition Letters, vol. 23, no. 1-3, pp. 183–190, 2002

Introduction

FRBSs learning. Carrying out a feature selection process (II):

- Neuro-fuzzy systems
  - **D. Chakraborty and N. Pal, A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification, IEEE Trans. Neural Netw., vol. 15, no. 1, pp. 110–123, 2004**
  - **R. Li, M. Mukaidono, and I. Burhan, A fuzzy neural network for pattern classification and feature selection, Fuzzy Sets and Systems, vol. 130, no. 1, pp. 101–108, 2002**
  - **V. Ravi and H. Zimmermann, A neural network and fuzzy rule base hybrid for pattern classification, Soft Computing, vol. 5, no. 2, pp. 152–159, 2001**
- Other different techniques
  - **V. Ravi, P. Reddy, and H. Zimmermann, Pattern classification with principal component analysis and fuzzy rule bases, European Journal of Operational Research, vol. 126, no. 3, pp. 526–533, 2000**
  - **V. Ravi and H. Zimmermann, Fuzzy rule based classification with FeatureSelector and modified threshold accepting, European Journal of Operational Research, vol. 123, no. 1, pp. 16–28, 2000**
  - **J. Roubos, M. Setnes, and J. Abonyi, Learning fuzzy classification rules from labeled data, Information Sciences, vol. 150, no. 1-2, pp. 77–93, 2003**

Introduction

FRBSs learning. Compacting and reducing a previous rule set (I):

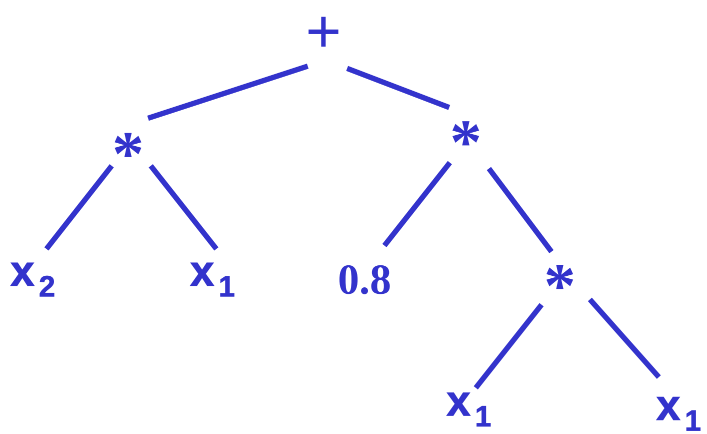
- Using GFSs
  - J. Casillas, O. Cordón, M. del Jesus, and F. Herrera, Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction, IEEE Trans. Fuzzy Syst., vol. 13, no. 1, pp. 13–29, 2005
  - H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, Selecting fuzzy if-then rules for classification problems using genetic algorithms, IEEE Trans. Fuzzy Syst., vol. 3, no. 3, pp. 260–270, 1995
  - H. Ishibuchi and T. Yamamoto, Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, Fuzzy Sets and Systems, vol. 141, no. 1, pp. 59–88, 2004
- Orthogonal transformation methods
  - M. Setnes and R. Babuˇska, Rule base reduction: Some comments on the use of orthogonal transforms, IEEE Trans. Syst., Man, Cybern. B, vol. 31, pp. 199–206, 2001
  - Y. Yam, P. Baranyi, and C. Yang, Reduction of fuzzy rule base via singular value decomposition, IEEE Trans. Fuzzy Syst., vol. 7, no. 2, pp. 120–132, 1999
  - J. Yen and L. Wang, Simplifying fuzzy rule-based models using orthogonal transformation methods, IEEE Trans. Syst.,Man, Cybern. B, vol. 29, pp. 13–24, 1999

Introduction

FRBSs learning. Compacting and reducing a previous rule set (II):

- Similarity measures
  - **J. Abonyi, J. A. Roubos, and F. Szeifert, Data-driven generation of compact, accurate, and linguistically sound fuzzy classifiers based on a decision-tree initialization, International Journal of Approximate Reasoning, vol. 32, no. 1, pp. 1–21, 2003**
  - **A. Bouchachia and R. Mittermeir, Towards incremental fuzzy classifiers, Soft Computing, vol. 11, no. 2, pp. 193–207, 2007**
  - **S. Papadakis and J. Theocharis, A genetic method for designing TSK models based on objective weighting: application to classification problems, Soft Computing, vol. 10, no. 9, pp. 805–824, 2006**

Genetic Programming (GP)



$$S \rightarrow x_1 \mid x_2 \mid N \mid + \; S \; S \mid * \; S \; S$$
$$N \rightarrow \Re$$

Chromosome: $+ * x_2 \, x_1 * 0.8 * x_1 \, x_1$

Expression: $x_2 * x_1 + 0.8 * x_1^2$

Introduction

FRBSs learning. GP-based approaches (I):

- M. Akbarzadeh-T, K. Kumbla, E. Tunstel, and M. Jamshidi, Soft computing for autonomous robotic systems, Computers and Electrical Engineering, vol. 26, no. 1, pp. 5–32, 2000
- A. Homaifar, D. Battle, E. Tunstel, and G. Dozier, Genetic programming design of fuzzy controllers for mobile robot path tracking, International Journal of Knowledge-Based Intelligent Engineering Systems, vol. 4, no. 1, pp. 33–52, 2000
- E. Alba, C. Cotta, and J. Troya, Evolutionary design of fuzzy logic controllers using strongly-typed GP, Mathware & Soft Computing, vol. 6, pp. 109–124, 1999
- A. Geyer-Schulz, Fuzzy rule-based expert systems and genetic machine learning. Heidelberg: Physica-Verlag, 1995
- L. Sánchez and J. Corrales, Niching scheme for steady state GA-P and its application to fuzzy rule based classifiers induction, Mathware & Soft Computing, vol. 7, no. 2-3, pp. 337–350, 2000
- L. Sánchez, I. Couso, and J. Corrales, Combining GP operators with SA search to evolve fuzzy rule based classifiers, Information Sciences, vol. 136, no. 1-4, pp. 175–191, 2001
- R. Mendes, F. de B. Voznika, A. Freitas, and J. Nievola, A genetic-programming-based approach for the learning of compact fuzzy rule-based classification systems, in Principles of Data Mining and Knowledge Discovery: 5th European Conference (PKDD’01), ser. LNCS vol. 2168. Springer-Verlag, 2001, pp. 314–325
- A. Tsakonas, A comparison of classification accuracy of four genetic programming-evolved intelligent structures, Information Sciences, vol. 176, no. 6, pp. 691–724, 2006

Introduction

FRBSs learning. GP-based approaches (II):

- B.-C. Chien, J. Lin, and T.-P. Hong, Learning discriminant functions with fuzzy attributes for classification using genetic programming, Expert Systems with Applications, vol. 23, no. 1, pp. 31–37, 2002
- F. Berlanga, del M.J. Jesus, and F. Herrera, Learning fuzzy rules using genetic programming: Context-free grammar definition for high-dimensionality problems, in Proc. I Workshop on Genetic Fuzzy Systems (GFS'05), Granada, Spain, 2005, pp. 136–141
- F. Berlanga, del M.J. Jesus, M. Gacto, and F. Herrera, A genetic-programming-based approach for the learning of compact fuzzy rule-based classification systems, in 8th International Conference on Artificial Intelligence and Soft Computing (ICAISC'06), ser. LNCS vol. 4029. Springer-Verlag, 2006, pp. 182–191

Index

- Introduction
- GP-COACH Algorithm
- Experimental Framework and Results
- Conclusions

GP-COACH Algorithm

Main Features

A. Uses a **context-free grammar** that allow the learning of **DNF fuzzy rules** and the absence of some of the input features

B. Follows the **genetic cooperative-competitive learning (GCCL) approach** which encodes a single rule per individual in the population and the complete RB is formed by the whole population

- **Local fitness function** → evaluates individuals (rules)
- **Global fitness function** → evaluates one population (rule set)

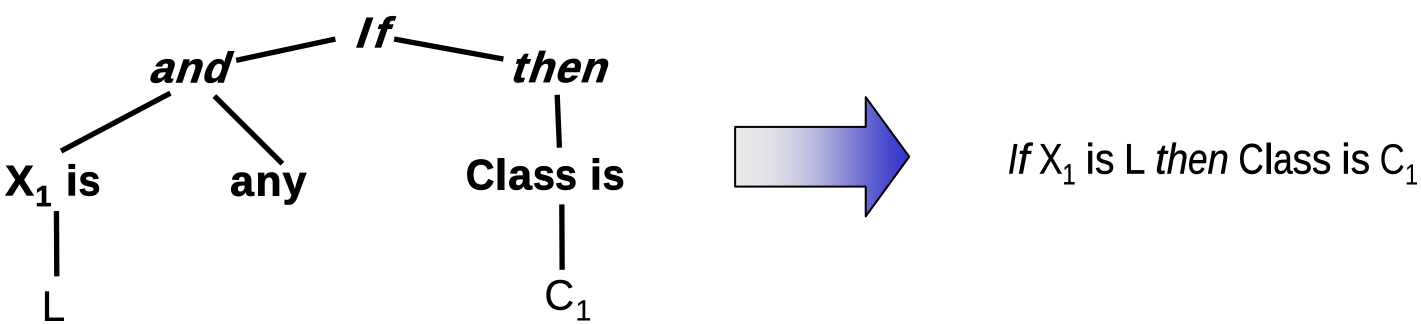
C. Includes a **mechanism to maintain the diversity** in the population that eliminates irrelevant rules: **Token Competition**

D. Uses a **two level hierarchical inference process**

GP-COACH Algorithm

Context-free Grammar

- start → [If], antec, [then], conseq, [.]
- antec → descriptor1, [and], descriptor2.
- descriptor1 → [any].
- descriptor1 → [X_1 is] label.
- descriptor2 → [any].
- descriptor2 → [X_2 is] label.
- label → {member(?a, [L, M, H, L or M, L or H, M or H, L or M or H])}, [?a].
- conseq → [Class is] descriptorClass
- descriptorClass → {member(?a, [C_1, C_2, C_3])}, [?a].



Local Fitness Function

$$\boxed{\text{raw\_fitness} = (\alpha \times \text{Confidence}) + ((1-\alpha) \times \text{Support})}$$

$$\text{Confidence} = \frac{\mu_{\text{tp}}}{(\mu_{\text{tp}} + \mu_{\text{fp}})}$$

$$\text{Support} = \frac{\mu_{\text{tp}}}{N_{C^k}}$$

μ_{tp} ➔ is the sum of the matching degree for true positives

μ_{fp} ➔ is the sum of the matching degree for false positives

N_{C^k} ➔ is the number of examples belonging to the same class that the one indicated in the consequent of the rule

GP-COACH Algorithm

Global Fitness Function

$$\text{Global\_fitness} = (w_1 \times \%Tra) + \left(w_2 \times \overline{\#V}\right) + \left(w_3 \times \overline{\#C}\right) + \left(w_4 \times \overline{\#R}\right)$$

$$\overline{\#A} = 1 - \#A \text{ and } \#A = \{\#V, \#C, \#R\}$$

$\%Tra$ ➔ the normalized value of the correct percentage on training examples

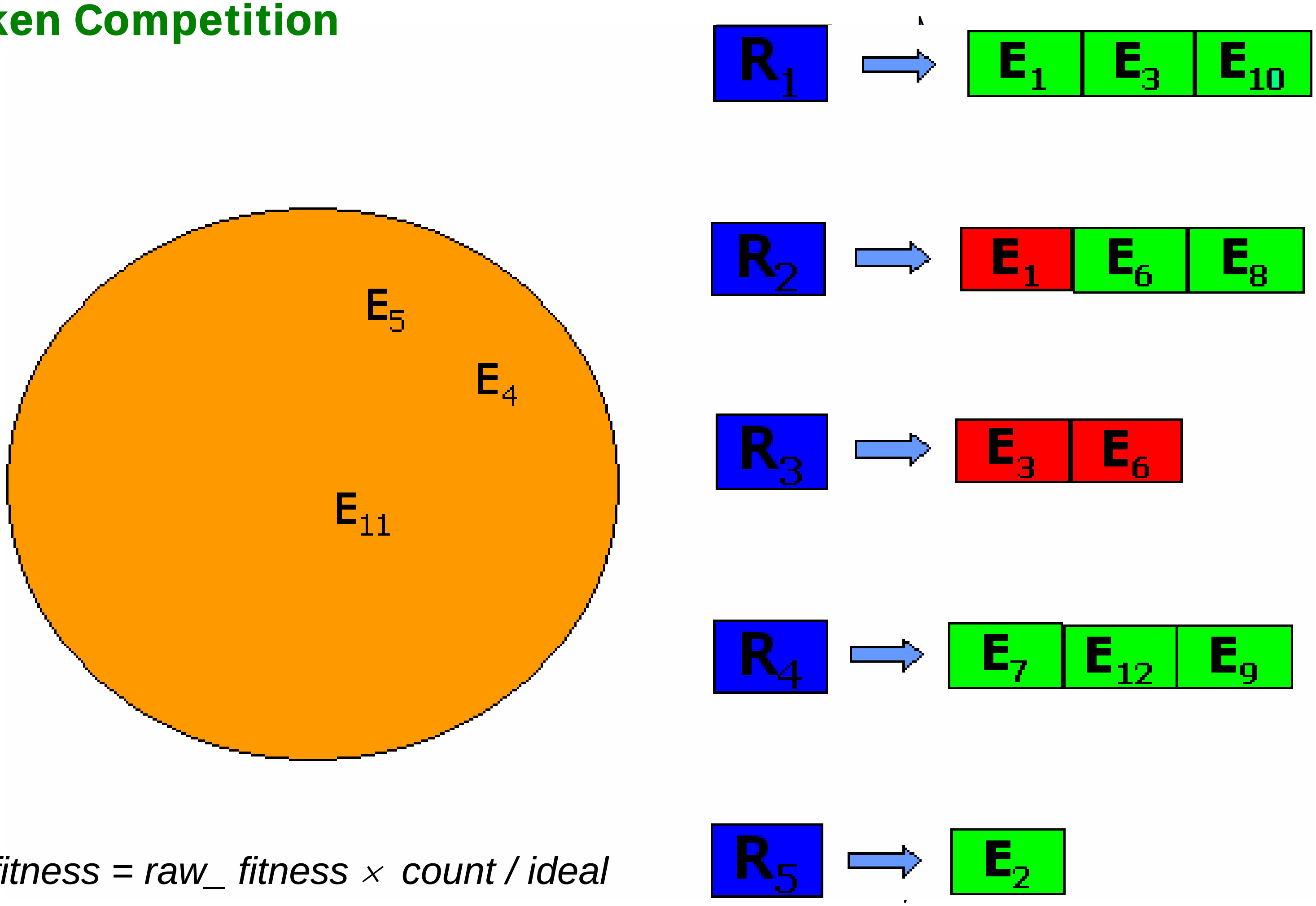
$\#V$ ➔ the normalized value of the number of variables per individual (rule) in the population

$\#C$ ➔ the normalized value of the number of labels (or conditions) per individual

$\#R$ ➔ the normalized value of the number of rules in the population

GP-COACH Algorithm

Token Competition



$$Modified\_fitness = raw\_fitness \times count / ideal$$

GP-COACH Algorithm

Two level hierarchical inference process

1. **Primary rules:** Strong and general rules created by the genetic operators. They cover most of the examples

2. **Secondary rules**: Weaker and more specific rules, generated after the Token Competition procedure in order to increase the diversity in the population

   - *Secondary rules are only taken into account if there not exist any primary rule matching with some of the examples*

   - *It is also possible that GP-COACH learns rule sets having no secondary rules, because their primary rules are strong enough to cover all the given examples*

GP-COACH Algorithm

Genetic Operators

1. **Crossover**: Produces one child from two parents. A part in the first parent is randomly selected and exchanged by another part, randomly selected, in the second one.

2. **Mutation**: Operates on label sets level

    - Add a label to the set
    - Remove a label from the set
    - Exchange a label in the set by another one not included in it

3. **Insertion**: Adds a variable to a rule

4. **Dropping Condition**: Remove a variable from a rule

GP-COACH Algorithm

Pseudocode

```
Initialize(Initial_pop)
For each Initial_pop[i] do
   Initial_pop[i].fitness ← Evaluate(Initial_pop[i])
End for
Copy Initial_pop to Best_pop
Best_pop.global_fitness ← Global_score(Best_pop)
Copy Initial_pop to Current_pop
While(not termination-condition) do
   Offspring_pop = Ø
   While (Size(Offspring_pop) ≠ Size(Current_pop)) do
      Parent ← Binary_tournament(Current_pop)
      Child ← Genetic_operator(Parent)
      Evaluate(Child)
      Add Child to Offspring_pop
   End while
   Joint_pop ← Current_pop U Offspring_pop
   New_current_pop ← Token_competition(Joint_pop)
   New_current_pop.global_fitness ← Global_score(New_current_pop)
   If New_current_pop.global_fitness > Best_pop.global_fitness then
      Best_pop.global_fitness ← New_current_pop.global_fitness
      Copy New_current_pop to Best_pop
   End if
   Copy New_current_pop to Current_pop
End while
Return(Best_pop)
```

Index

- Introduction
- GP-COACH Algorithm
- Experimental Framework and Results
- Conclusions

Experimental Framework

Datasets

Name	N. Instances	N. Features	N. Classes
Ionosphere	351	34	2
Pen-based	10992	16	10
Spambase	4597	57	2
Thyroid	7200	21	3
Vehicle	846	18	4
Wdbc	569	30	2

Experimental Framework

Comparative Methods

1. **2SLAVE**:
   - GA-based method (IRL approach) for learning DNF fuzzy rules
   - Feature selection process during the learning phase

2. **Tsakonas**:
   - GP-based method for learning FRBCSs
   - Context-free grammar to generate complete rule sets per individual in the population (Pittsburgh approach)

3. **FRBCS_GP:**
   - Does not uses a two level hierarchical inference process
   - Population size does not change during the evolutionary process
   - Does not uses any kind of global fitness score to keep the best evolved population
   - Uses a crisp fitness function, which uses the number of positive and negative examples
   - Uses a ranking selection scheme to select parents from the population, and different genetic operators to generate new children

Results (I)

	Ionosphere					Pen-based				
Method	**#R**	**#V**	**#C**	**%Tra**	**%Test**	**#R**	**#V**	**#C**	**%Tra**	**%Test**
$2SLAVE_1$	6	12.73	25.16	82.73	81.88	**39.97**	11.76	26.55	75.98	75.85
$2SLAVE_2$	6	12.73	25.16	85.46	84.14	**39.97**	11.76	26.55	81.32	81.16
$Tsakonas_1$	35.4	2.44	**2.92**	40.37	39.69	65	**1.17**	**1.2**	15.56	15.58
$Tsakonas_2$	35.4	2.44	**2.92**	51.05	52.42	65	**1.17**	**1.2**	15.19	15.38
$FRBCS\_GP_1$	14.03	2.41	6.63	83.65	82.81	87.07	3.25	7.67	73.43	73.44
$FRBCS\_GP_2$	14.03	2.41	6.63	82.86	81.48	87.07	3.25	7.67	75.74	75.55
$GP\text{-}COACH_1$	**5.9**	2.14	5.20	93.61	92.14	85.9	4.22	9.08	78.97	78.85
$GP\text{-}COACH_2$	5.97	**2.06**	5.07	**94.26**	**92.31**	85.13	4.26	9.13	**82.51**	**82.38**

- **#R**: Average rule number
- **#V**: Average antecedent variables per rule
- **#C**: Average conditions number per rule
- **%Tra**: Correct percentage with training
- **%Test**: Correct percentage with test

Subscrips: Related to the fuzzy reasoning method (FRM) used:

- **1**: Classical FRM (max-min)
- **2**: Normalized sum FRM

Results (II)

	Spambase					Thyroid				
Method	**#R**	**#V**	**#C**	**%Tra**	**%Test**	**#R**	**#V**	**#C**	**%Tra**	**%Test**
$2SLAVE_1$	**7.9**	22.25	49.19	73.66	73.74	**3.7**	9.23	20.52	92.81	92.87
$2SLAVE_2$	**7.9**	22.25	49.19	69.87	70.14	**3.7**	9.23	20.52	92.85	92.88
$Tsakonas_1$	27.43	**3.03**	**3.68**	39.63	39.93	37.27	2.02	**2.56**	80.31	80.38
$Tsakonas_2$	27.43	**3.03**	**3.68**	49.63	49.49	37.27	2.02	**2.56**	43.23	43.77
$FRBCS\_GP_1$	43.93	13.46	31.57	73.72	73.50	29.33	12.01	34.16	92.90	92.89
$FRBCS\_GP_2$	43.93	13.46	31.57	75.03	74.55	29.33	12.01	34.16	92.82	92.82
$GP\text{-}COACH_1$	14	5.27	9.63	77.29	76.62	6.13	**1.59**	2.73	**93.27**	**93.26**
$GP\text{-}COACH_2$	10.77	3.93	7.96	**83.70**	**83.59**	6.3	1.60	2.80	93.26	93.25

- **#R**: Average rule number
- **#V**: Average antecedent variables per rule
- **#C**: Average conditions number per rule
- **%Tra**: Correct percentage with training
- **%Test**: Correct percentage with test

Subscrips: Related to the fuzzy reasoning method (FRM) used:
- **1**: Classical FRM (max-min)
- **2**: Normalized sum FRM

Results (III)

	Vehicle					Wdbc				
Method	**#R**	**#V**	**#C**	**%Tra**	**%Test**	**#R**	**#V**	**#C**	**%Tra**	**%Test**
$2SLAVE_1$	**11.73**	11.07	24.96	49.11	47.30	5.2	10.31	20.22	92.59	92.74
$2SLAVE_2$	**11.73**	11.07	24.96	49.21	47.38	5.2	10.31	20.22	92.43	92.33
$Tsakonas_1$	37.73	**1.89**	**2.48**	26.15	26.09	27.8	2.76	3.69	51.89	53.28
$Tsakonas_2$	37.73	**1.89**	**2.48**	26.32	27.00	27.8	2.76	3.69	52.39	52.71
$FRBCS\_GP_1$	55.23	6.10	15.84	53.13	52.60	14.9	3.59	8.85	94.48	93.49
$FRBCS\_GP_2$	55.23	6.10	15.84	56.47	55.05	14.9	3.59	8.85	**95.27**	**94.37**
$GP\text{-}COACH_1$	34.03	4.08	9.17	54.81	53.55	4.97	1.11	**2.61**	94.19	92.32
$GP\text{-}COACH_2$	33.53	4.22	9.57	**58.33**	**55.09**	**4.8**	**1.09**	2.67	95.03	93.38

- **#R**: Average rule number
- **#V**: Average antecedent variables per rule
- **#C**: Average conditions number per rule
- **%Tra**: Correct percentage with training
- **%Test**: Correct percentage with test

Subscrips: Related to the fuzzy reasoning method (FRM) used:

- **1**: Classical FRM (max-min)
- **2**: Normalized sum FRM

Results' Analysis (I)

Compactness:

- 2SLAVE algorithm usually obtains the rule sets with the lower number of rules. However, these rules present a high number of variables and conditions per variable, that makes the interpretability of the rule sets decrease
- GP-COACH learns rule sets with a quite small number of rules, having them also a low number of variables and condition per variable
- Tsakonas algorithm obtains rule sets with a low number of variables and conditions per variable, but with a high number of fuzzy rules
- GP-COACH is able to learn more compact rule sets than the ones obtained by FRBCS_GP

Results' Analysis (II)

Accuracy:

- GP-COACH is the algorithm that obtains the best test accuracy results with all the data sets, except with the Wdbc data set, were FRBCS_GP slightly outperforms its results
- 2SLAVE obtains good test accuracy results, comparable to the ones obtained by FRBCS_GP, but they do not outperform the ones obtained by GP-COACH
- Tsakonas algorithm obtains the worst results on test accuracy because of the simplicity of its learned rules

Index

- Introduction
- GP-COACH Algorithm
- Experimental Framework and Results
- Conclusions

Conclusions

- We have proposed GP-COACH, a Genetic Programming based method to obtain COmpact and ACcurate FRBCSs for High-dimensional problems:
  - Evolutionary process (with GCCL codification approach) that uses a context-free grammar to learn DNF rules ➔ rules with lower antecedent conditions
  - Niche formation mechanism (Token Competition) and a two types of rules (primary and secondary) to increase the diversity into the population ➔ gives out a fewer number of rules with a high generalization capability
- The experimental study has demonstrated the good performance of our proposal:
  - GP-COACH is able to learn rule sets with a small number of rules, variables and conditions per variable for high-dimensional problems
  - GP-COACH has outperformed the rest of algorithms with regard to the accuracy results on training and test data, showing a high performance for high-dimensional problems

Thank you for your attention!