# Automatic Learning of Simple and Accurate Fuzzy Rule Systems for Regression

## Doctoral Meeting

Ismael Rodríguez

17th July 2015

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
CAMPUS VIDA
CiTIUS Centro Singular de Investigación en Tecnoloxías da Información

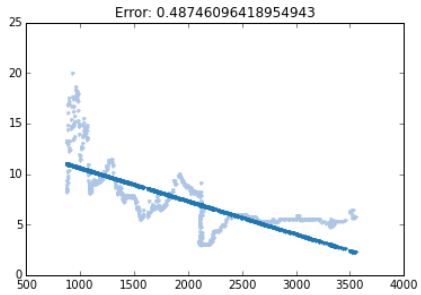
# Regression Problems

Regression

- Predict a real-valued output using values of input variables

Supervised learning

- Learn from labeled data
- Set of examples: pairs of input (vector) and output (real value)



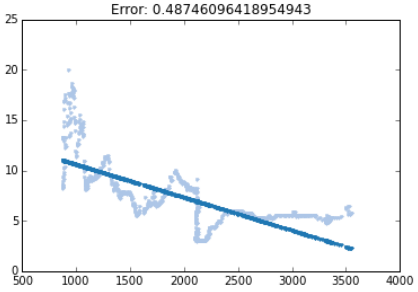
# Solving from the Scratch

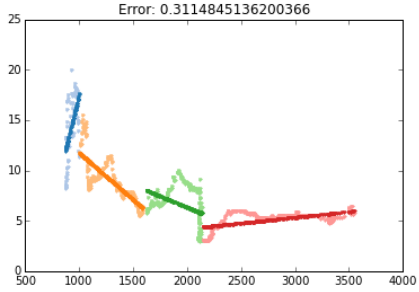
Why from the scratch?

- Problems with no prior knowledge
- First approach useful for the expert

Divide and Conquer

- Split until local input-output relation is nearly linear



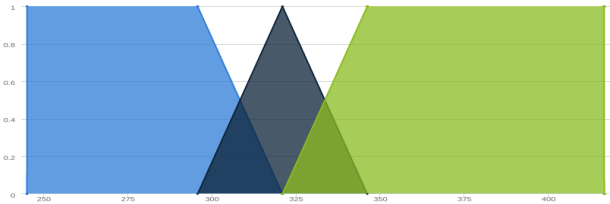
# Fuzzy Rules

Fuzzy Sets

- An element $x$ belongs to the set to some degree, e.g. $\mu_A(x) = 0.7$
  - ▷ The surface of this building is High with degree 0.6

Fuzzy Rule

- IF **x is A** THEN **y is B**
  - ▷ IF **surface is High** THEN **cool load is Very High**
- Meaning of A and B is provided by Fuzzy Sets



# TSK Fuzzy Rules

The output is a linear combination of the inputs

- IF $x_1$ **is A** and $x_2$ **is B** THEN $\boldsymbol{y = \beta_0 + \beta_1 x_1 + \beta_2 x_2}$
  - ▷ IF **surface is low** AND **wall area is high** THEN cool load = **0.2** surface + **1.7** wall
- Each rule has a different combination of the inputs

Inference

- Each rule is fired with different degree ($\mu_r(x)$) depending on the input values
- Each rule estimates a different value for the output ($\hat{y}_r$)
- Final output is the weigthed average:

$$\hat{y} = \frac{\sum_r \hat{y}_r \mu_r(x)}{\sum_r \mu_r(x)}$$

# Objective

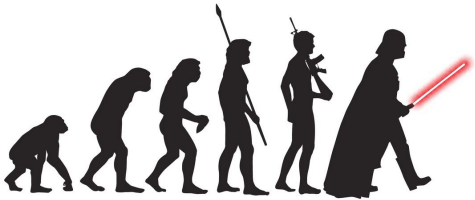
## Question

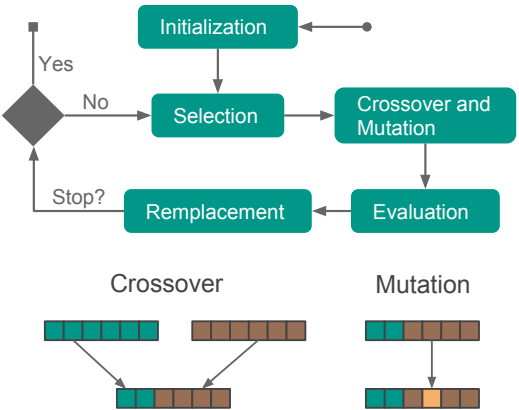
Is it possible to obtain **automatically simple** and **accurate** fuzzy rule models for regression problems without prior knowledge?

| Automatic Learn | Simple | Accurate |
|---|---|---|
| Fuzzy partitions | Few labels | Linear regions |
| Rule set | Low number of rules | Precise rules |
| Weights of consequents | Regularization | Regularization |

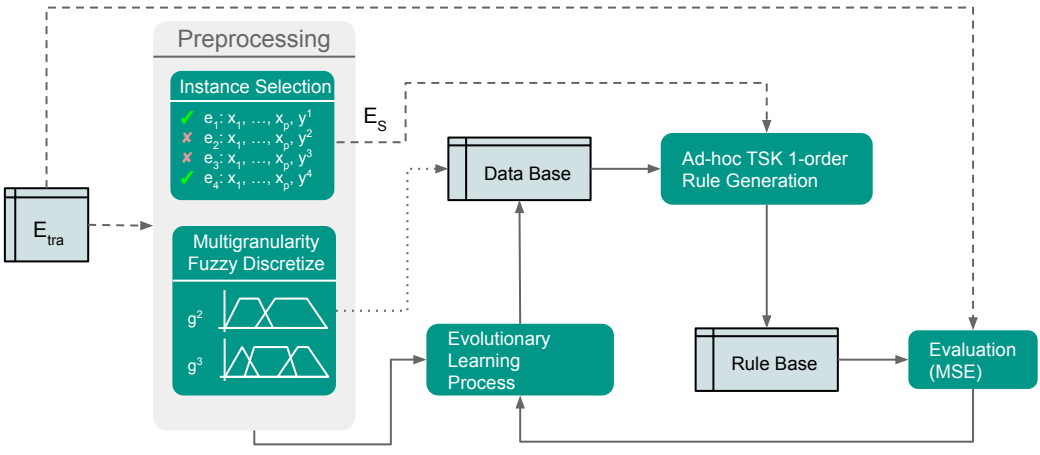
# Evolutionary Algorithms

Works like natural evolution





# FRULER


Preprocessing
Instance Selection
✓ e1: x1, ..., xp, y1
✘ e2: x1, ..., xp, y2
✘ e3: x1, ..., xp, y3
✓ e4: x1, ..., xp, y4
Multigranularity Fuzzy Discretize
g2
g3
Etra
ES
Data Base
Ad-hoc TSK 1-order Rule Generation
Evolutionary Learning Process
Rule Base
Evaluation (MSE)

# Instance Selection

Based on conditional nearest neighbor

- Nearest example with same class
- Nearest example with different class

Two phases:

- Class conditional
  - ▷ Enlarge margin between "classes"
- Thin-out
  - ▷ Removes examples not important for the decision boundary

## Adapt to Regression

- Discretization of the output
  - ▷ Kernel Density Estimation
  - ▷ Local minimum = split points
- Error measure in 1-NN for regression
  - ▷ More precision
- Relax the stopping criteria, allowing iterations without improvement
  - ▷ Error in regression varies more than accuracy in classification

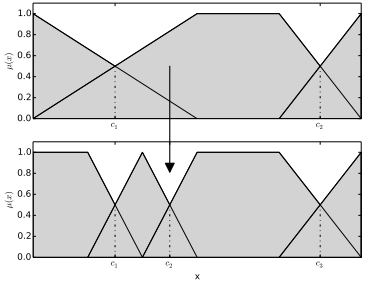
# Multi-Granularity Fuzzy Discretization

Obtains several partition levels, from 1 label to N labels

- There is no information about how many labels are needed

For each input variable

- Searches the split points iteratively
- Only a new split point is added at each step, obtaining two new intervals
- Selects the split point that maximizes the linear relation in each resulting interval
- Stops when BIC worsens for a consecutive number of iterations
- Fuzzify the crisp intervals relaxing the frontiers

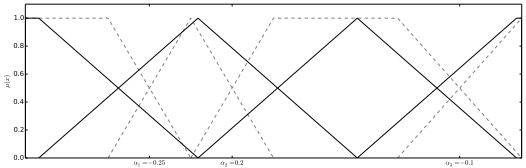


# Evolutionary Algorithm

## Individual Codification

The split points are suboptimal

- Allow a small lateral displacement




Parameters to learn:

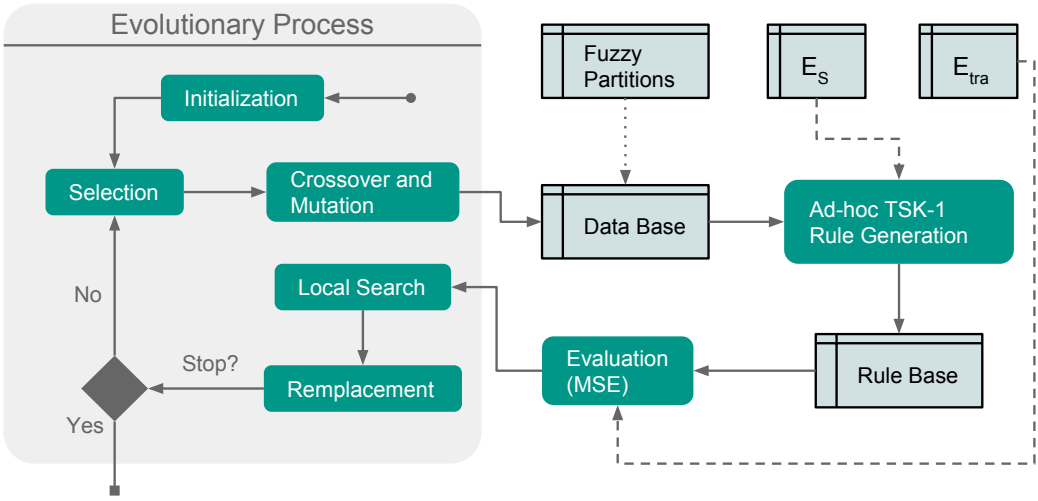
- Granularity level in each input
- Displacement of each split point

Codification

- $[g_1, g_2, \ldots, g_p]$
- $[[s_1^1, \ldots, s_1^n], \ldots, [s_p^1, \ldots, s_p^m]]$

# Evolutionary Algorithm

## Operators



# TSK Rule Base Generation

## Antecedents

1. Apply the displacements
2. Wang & Mendel: obtain the rule set that covers all examples

## Consequents

- Learn all consequents at once
- Elastic-Net (L1+L2): $\hat{\beta} = min_\beta \frac{1}{2} \sum_i (y - \beta x_i)^2 + \alpha\lambda \sum_j |\beta_j| + (1-\alpha)\lambda \sum_j \beta_j^2$
- Solve with Stochastic Gradient Descent
  - $\alpha = 0.95$: behaves like L1 but with low weights like L2
  - Learning rate and regularization parameter $\lambda$ with fast grid Search

# Comparison with State-of-the-art

## Precision

| algorithms | FRULER | $FS_{MOGFS}^{e}$+TUN$^{e}$ | L-METSK-HD$^{e}$ | A-METSK-HD$^{e}$ |
|---|---|---|---|---|
| ELE1 | 2.012 | **1.954** | 1.925 | 2.022 |
| PLA | 1.219 | 1.194 | 1.218 | **1.136** |
| QUA | 0.0181 | **0.0178** | 0.019 | 0.0181 |
| ELE2 | 6,729 | 10,548 | 20,095 | **3,192** |
| FRIE | **0.731** | 3.138 | 3.084 | 1.888 |
| MPG6 | **3.727** | 4.562 | 4.469 | 4.478 |
| DELAIL | 1.458 | 1.528 | 1.621 | **1.402** |
| DEE | **0.080** | 0.093 | 0.095 | 0.103 |
| DELELV | 1.045 | 1.086 | 1.119 | 1.031 |
| ANA | 0.008 | **0.003** | 0.006 | 0.004 |
| MPG8 | **4.084** | 4.747 | 5.61 | 5.391 |
| ABA | 2.393 | 2.509 | 2.581 | **2.392** |
| CON | **20.598** | 32.977 | 38.394 | 23.885 |
| STP | **0.353** | 0.912 | 0.78 | 0.387 |
| WAN | **0.888** | 1.635 | 1.773 | 1.189 |
| WIZ | **0.663** | 1.011 | 1.296 | 0.944 |
| FOR | **2,214** | 2,628 | 4,633 | 5,587 |
| MOR | **0.007** | 0.019 | 0.028 | 0.013 |
| TRE | **0.027** | 0.044 | 0.052 | 0.038 |
| BAS | 305,777 | **261,322** | 320,133 | 368,820 |
| CAL | 2.110 | 2.95 | 2.638 | **1.71** |
| MV | 0.083 | 0.158 | 0.244 | **0.061** |
| HOU | **8.005** | 9.4 | 10.368 | 8.64 |
| ELE | **2.934** | 9 | 8.9 | 7.02 |
| CA | **4.634** | 5.216 | 5.88 | 4.949 |
| POLE | 110.898 | 102.816 | 150.673 | **61.018** |
| PUM | 0.367 | 0.292 | 0.594 | **0.287** |
| AIL | **1.404** | 2 | 1.822 | 1.51 |

Best in 15 of 28 datasets

# Comparison with State-of-the-art

## Number of rules

| algorithms | FRULER | $FS_{MOGFS}^{e}$+TUN$^{e}$ | L-METSK-HD$^{e}$ | A-METSK-HD$^{e}$ |
|---|---|---|---|---|
| ELE1 | **4.1** | 8.1 | 15 | 11.4 |
| PLA | **1.4** | 18.6 | 23 | 19.2 |
| QUA | 7.8 | **3.2** | 35.9 | 18.3 |
| ELE2 | **4.3** | 8 | 59 | 36.9 |
| FRIE | **8.0** | 22 | 95.1 | 66 |
| MPG6 | **13.7** | 20 | 99.6 | 53.6 |
| DELAIL | **2.5** | 6.2 | 98.3 | 36.8 |
| DEE | **7.9** | 18.3 | 96.4 | 50.6 |
| DELELV | **5.8** | 7.9 | 91 | 39.1 |
| ANA | **3.9** | 10 | 48.9 | 33.3 |
| MPG8 | **12.7** | 23 | 98.7 | 64.2 |
| ABA | **4.5** | 8 | 42.4 | 23.1 |
| CON | **8.9** | 15.4 | 96.5 | 53.7 |
| STP | 42.4 | **23** | 100 | 66.4 |
| WAN | **5.6** | 8 | 91.1 | 48 |
| WIZ | **8.9** | 10 | 55.4 | 29.1 |
| FOR | **5.6** | 10 | 93.7 | 40.6 |
| MOR | 7.9 | 7 | 40.9 | 27.2 |
| TRE | **4.5** | 9 | 42.8 | 28.1 |
| BAS | **6.2** | 17 | 95.7 | 59.8 |
| CAL | 15.4 | **8.4** | 99.8 | 55.8 |
| MV | **6.0** | 14 | 76.4 | 56.5 |
| HOU | 12.1 | **11.7** | 68.9 | 30.5 |
| ELE | **5.4** | 8 | 76.4 | 34.9 |
| CA | **7.1** | 14 | 71.3 | 32.9 |
| POLE | 40.8 | **13.1** | 100 | 46.3 |
| PUM | **7.8** | 17.6 | 87.5 | 63.3 |
| AIL | **8.5** | 15 | 99.1 | 48.4 |

Best in 22 of 28 datasets

# Statistical Analysis

| Algorithm | Ranking |
|---|---|
| FRULER | 1.714 |
| A-METSK-HD$^e$ | 2.036 |
| FS$_{\text{MOGFS}}{}^e$+TUN$^e$ | 2.786 |
| L-METSK-HD$^e$ | 3.464 |
| Holm p-value | 0.079 |

Table: Precision test

| Algorithm | Ranking |
|---|---|
| FRULER | 1.214 |
| FS$_{\text{MOGFS}}{}^e$+TUN$^e$ | 1.786 |
| A-METSK-HD$^e$ | 3 |
| L-METSK-HD$^e$ | 4 |
| Holm p-value | $< 1E-4$ |

Table: Complexity test

Tests perfomed using:

# Example of Learned Model

TSKView

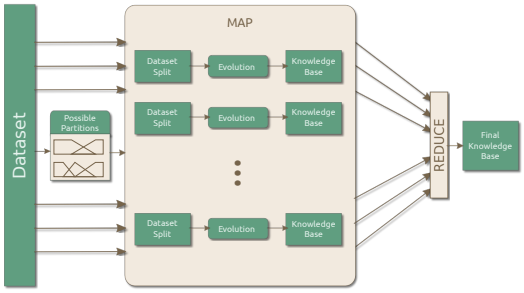
# Scaling FRULER

Computational time

- Less than 1 hour for small-medium datasets
- From 2h to 30h for large datasets

More Divide and Conquer:

1. Split dataset into Mappers
2. Subset of input variables for each split
3. Use FRULER for each split
4. Aggregate the resultant TSK Fuzzy Rule bases



# Conclusions

- FRULER learns simple and accurate TSK Fuzzy Rule Bases without prior knowledge
- Uses two general-purpose methods:
  - Instance Selection for regression
  - Non-uniform multi-granularity fuzzy discretization
- Automatic generation of TSK Fuzzy Rules from fuzzy partitions
  - Elastic-Net for low overfitting
  - Stochastic Gradient Descent for fast learning
- Compared with other three state-of-the-art approaches
  - High accuracy
  - Lower number of rules
  - Linguistic data base

Thank you!

# Questions?

ismael.rodriguez@usc.es