Using the OLS algorithm to build interpretable rule bases: an application to a depollution problem

S. Destercke [1] S. Guillaume [2] and B. Charnomordic [3]

[1]IRSN (Institute of Radioprotection and Nuclear Safety), DPAM, SEMIC, LIMSI
Cadarache, France

[2]Cemagref (Agricultural and Environmental Engineering Research), TEMO
Montpellier, France

[3]INRA (French National Institute for Agricultural Research), LASB
Montpellier, France

FUZZ'IEEE 2007

Why and how?

Why? (motivations)

Among fuzzy learning methods, difficult to find one which ...

- treats regression problem,
- is numerically efficient (good predictive abilities and not require too many resources),
- builds an interpretable rule base (RB),

...at the same time (most efficient algorithms giving interpretable RBs are designed for classification problems).

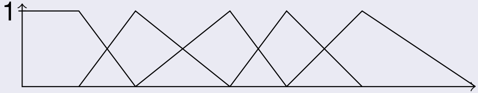
How? (our proposition)

Take a numerically efficient algorithm designed for regression problems, the OLS (Orthogonal Least Squares), and make it interpretable

Interpretability criteria: a reminder

Retained interpretability criteria:

- Interpretable input fuzzy partitions (domain coverage, reasonable number, distinguishable)
  - here, we take standardized fuzzy partitions with triangular membership functions.



- Reasonable number of rules in the RB
- Limited number of distinct rule conclusions

Building a fuzzy system

Given N samples, to optimize a zero order Sugeno fuzzy system by Least Squares comes down to the problem

$$\min\,(\widehat{y}-y)^2 \equiv min\left(\sum_{k=1}^{N}\frac{\sum_r\left(\bigwedge_{i=1}^{p}\mu(x_i^k)\right)\theta_i}{\sum_r\bigwedge_{i=1}^{p}\mu(x_i^k)}-y^k\right)^2$$

where p is the number of premises (input space dimension)

Solving it require to optimize r (RB),$\mu(x_i^k)$ (membership fc.) and θ_i (rule conclusions) $\rightarrow$ difficult and non-linear problem!

OLS: how it works

- **Original algorithm**

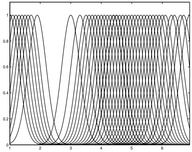
Linearize by fixing membership functions ($\mu(x_i^k)$) → Select most important rules by **orthogonal** variance decomposition (r) → Optimize conclusions by **Least Square** fitting (θ_i) → Rule base with optimized conclusions

- **Modified algorithm**

OLS: how it works

- **Original algorithm**

One gauss. MF per sample



Select rules that explain the most variance by Gram-Schmidt decomposition

By **L-S** optimization, each rule has a distinct conclusion

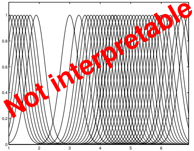
Linearize by fixing membership functions ($\mu(x_i^k)$) → Select most important rules by **orthogonal** variance decomposition (r) → Optimize conclusions by **Least Square** fitting (θ_i) → Rule base with optimized conclusions

- **Modified algorithm**

OLS: how it works

- **Original algorithm**

One gauss. MF per sample



Select rules that explain the most variance by Gram-Schmidt decomposition

Few rules

By **L-S** optimization, each rule has a distinct conclusion

Poorly interpretable

Not interpretable

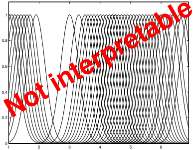
Linearize by fixing membership functions ($\mu(x_i^k)$)	Select most important rules by **orthogonal** variance decomposition (r)	Optimize conclusions by **Least Square** fitting (θ_i)	Rule base with optimized conclusions

- **Modified algorithm**

OLS: how it works

- **Original algorithm**

One gauss. MF per sample



Not interpretable

Select rules that explain the most variance by Gram-Schmidt decomposition

Few rules

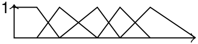
By **L-S** optimization, each rule has a distinct conclusion

Poorly interpretable

Not interpretable

Linearize by fixing membership functions ($\mu(x_i^k)$)	▶	Select most important rules by **orthogonal** variance decomposition (r)	▶	Optimize conclusions by **Least Square** fitting (θ_i)	▶	Rule base with optimized conclusions

Build interpretable partitions fitting data by hierarchical process.



Eventually restrict number of selected rules

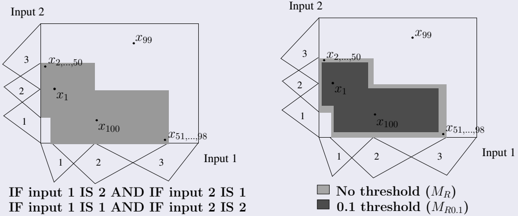
After LS optimization, reduce number of distinct conclusions by k-means algorithm

Interpretable

- **Modified algorithm**

How do we evaluate the final fuzzy system?

- Coverage Index $CI_\alpha = \frac{\#\text{ Active samples}}{\#\text{ Samples}}$, a sample being active if it fires at least one rule over threshold α.



$CI_0 = 0.99$ $CI_{0.1} = 0.02$

- Numerical accuracy measured by classical RMSE on active data $PI = \sqrt{\frac{1}{n}(\widehat{y} - y)^2}$ (n=number of active samples for given α)

Application: what and why?

What?

- Water depollution process by anaerobic digestion

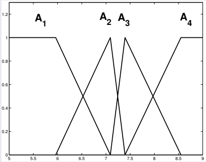
- 589 samples coming from a fixed-bed reactor of $1m^3$
- Input: 7 variables
- Output: Expert value between 0-1, characterizing current state as acidogenic or not

why?

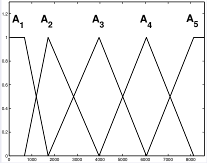
- Process requires little energy and produces renewable energy
- But bacteria population grow slow and sensitive to environment changes
- Need to build system to detect quickly unstable state threatening the population (i.e. fault detection)
- Acidogenic state particularly critical
- Using OLS to analyze data with expert help and improve detection systems.

Application: building input partitions

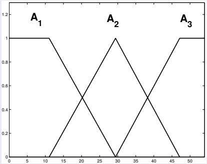
- Applying hierarchical process on data gives partitions for four variables (pH, Volatile Fatty Acids, Input flow *Qin*, CH_4 concentration)



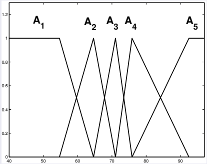
pH



vfa



Input flow rate (*Qin*)



CH_4

Application: analysis of first results

- OLS on 589 samples : final RB has 53 rules and PI=0.046

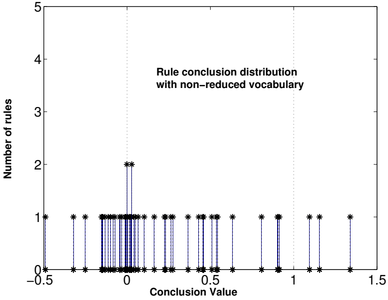
Remarks on these first result

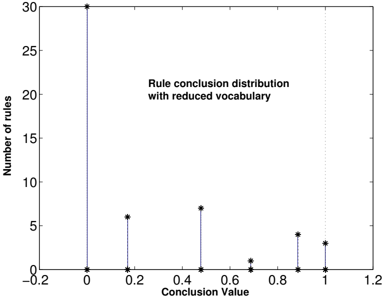
- **Rule ordering**: On 589 samples, only 35 have output > 0.5, while among the first 10 rules, 8 have output > 0.5 (with 6 close to 1) $\rightarrow$ Selecting rule by variance tends to privilege "faulty" samples, which contribute more to the variance.
- **Out of range conclusions**: Some computed rule conclusions are outside $[0, 1]$, due to the unconstrained least-square optimization.
- **Detection and treatment of outliers**: two of the first rules
  - If pH is "high" (A_3) and ..., then output is 0.999
  - If pH is "very high" (A_4) and ..., then output is 1

  were inconsistent with knowledge (acidogenic state incoherent with a basic pH). The 2 samples corresponding to these rules were removed.

Application: distinct conclusion reduction

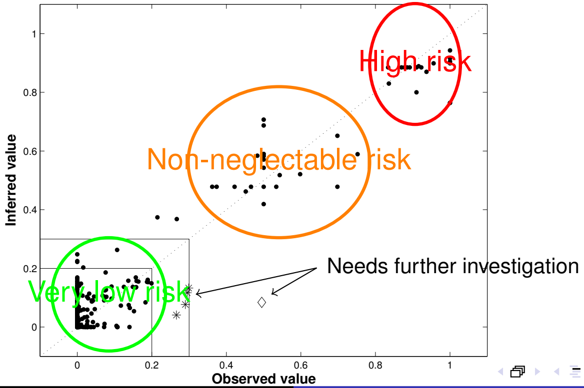
In final system, number of distinct conclusions was brought from 49 to 6





Application: final system summary

	Rules	PI(α=0)	CI_0	$CI_{0.1}$
Modified OLS	51	0.054	100 %	100 %
Original OLS	51	0.074	100 %	30 %



Application: summary

Applying the modified OLS algorithm allowed us to

- Remove erroneous data from sample base
- Extract rules corresponding to critical situations
- Point out interesting experimental points for experts
- Build a final interpretable system with a good qualitative predictive quality (and whose numerical efficiency competes with the one of the original method)

Moreover, OLS algorithm (by its principle) seems particularly fitted to problems when important samples are also rare, like fault detection problems.

modified OLS: advantages/defects/perspectives

Advantages

- Provides robust and interpretable rule bases for regression problems
- Focus on rare samples and on most important rules
- Can be used for knowledge extraction as well as for system modeling

Disadvantages

- Computational cost in high dimensional problems
- Variables have to be selected before applying OLS
- Learned Rules are complete (i.e. contain all inputs)

Perspectives

- Robustness study, Remedies to disadvantages cited above
- Extend to other methods based on orthogonalisation (e.g. TLS)
- Refine rule sel., e.g. by using backward-forward regression techniques