INTERPRETABILITY WITH DIVERSIFIED-BY-DESIGN RULES; SKOPE-RULES, A PYTHON PACKAGE

RONAN.GAUTIER@BPCE.FR, GREGOIRE.JAFFRE@BPCE.FR, BIBI.NDIAYE@BPCE.FR
WITH THE PARTICIPATION OF N. GOIX, F. GARDIN, J-M. SCHERTZER

MOTIVATION

Complex Machine Learning models sometimes achieve **high performance**, but lead to **opaque decisions**. Due to regulation or severe consequences on errors, **more interpretability is often necessary** [1]. Here, we present **Skope-Rules**, a rule-based **interpretable model**.

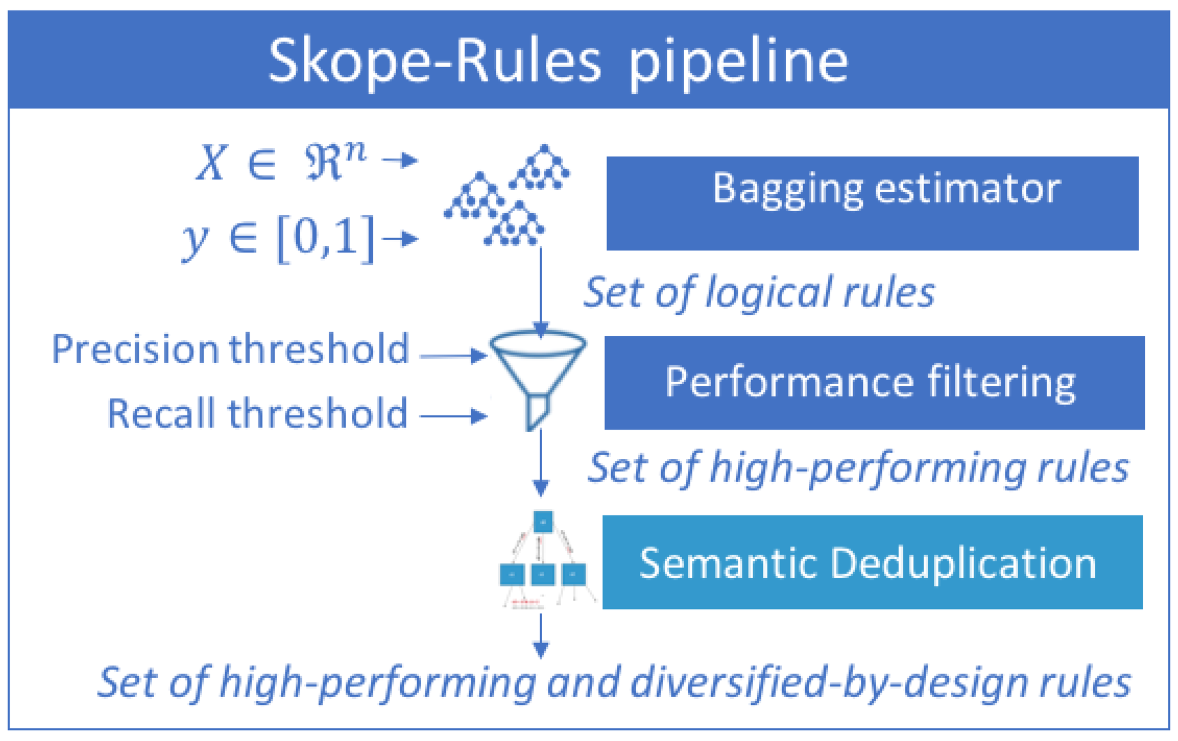
SKOPE-RULES: AN INTERPRETABLE RULE-BASED CLASSIFIER

Skope-Rules, the proposed interpretable model, aims at **learning decision rules** for "scoping" a target class, i.e. detecting instances of this class with high precision.
The problem of generating such rules has been widely considered, see e.g. RuleFit [2], Slipper [3], LRI [4], MLRules [5].

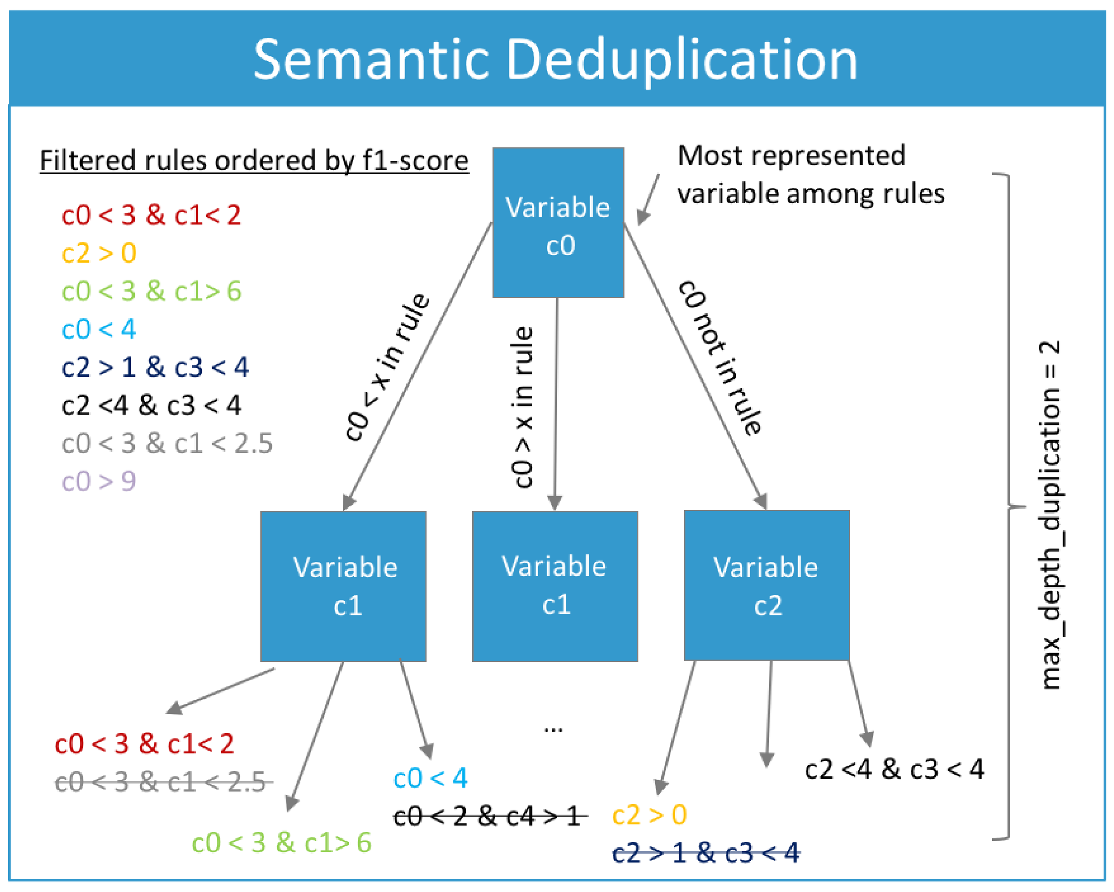
However, our approach mainly **differs** in the way that decision rules are chosen: **semantic deduplication based on variables composing each rule** as opposed to L1-based feature selection (RuleFit).

METHODOLOGY

- **Bagging estimator training**: Multiple decision tree classifiers, and potentially regressors (if a sample weight is applied), are trained. Note that each node in this bagging estimator can be seen as a rule.
- **Performance filtering**: Out-of-bag precision and recall thresholds are applied to select best rules.



- **Semantic deduplication**: A similarity filtering is applied to maintain enough diversity among the rules. The similarity measure of two rules is based on the number of their common terms. A term is a variable name combined with a comparison operator ($<$ or $>$).



IMPLEMENTATION

Skope-rules is a **Python package hosted on Scikit-Learn-Contrib** under the 3-Clause BSD license.

Code and documentation are available here:
`https://github.com/scikit-learn-contrib/skope-rules`
Installation: *pip install skope-rules*

EXAMPLE

- **Line 4-5:** Model is trained through standard scikit-learn API.
- **Line 8:** Predictions are made and can be used to evaluate performances of rules combined.
- **Line 11:** The computed rules are stored with their out-of-bag standalone performances (see below).

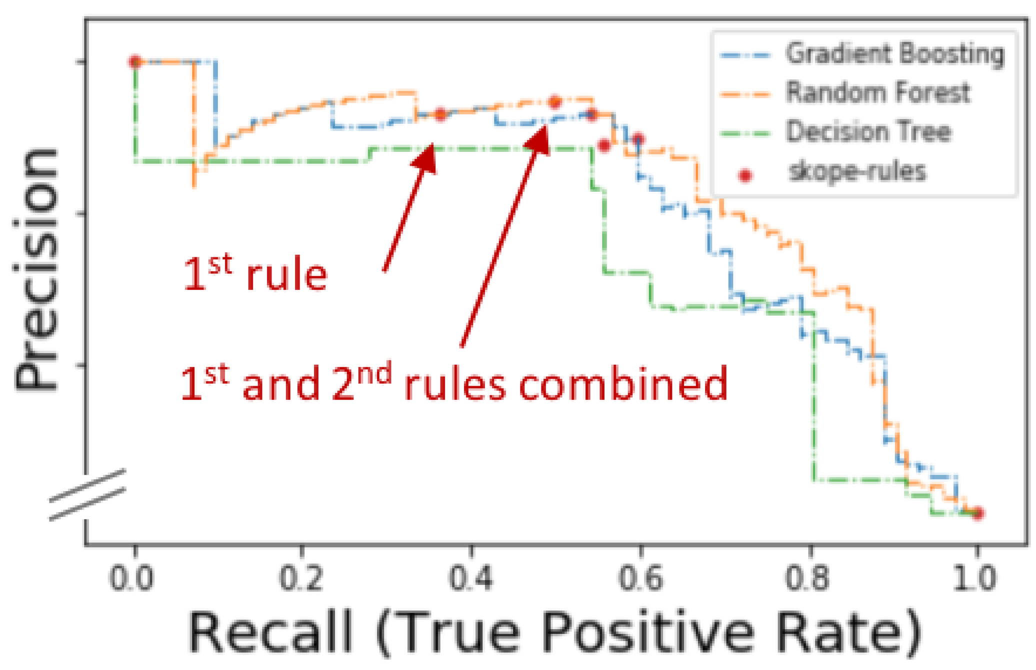
```
1  from skrules import SkopeRules
2
3  # Training
4  clf = SkopeRules(feature_names=X_train.columns)
5  clf.fit(X_train, y_train)
6
7  # Making predictions
8  y_score = clf.score_top_rules(X_test)
9
10 # Seeing the rules
11 clf.rules_
```

Example of output

debit_flows < 952 and credit_flows > 2411 and is_client_gold > 0.5, (0.95, 0.16, 1)

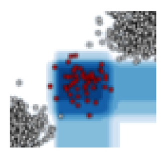
credit_balance > 327 and debit_flows < 523, (0.82, 0.33, 2)

Out-of-bag **precision** of the rule
Out-of-bag **recall** of the rule
This rule appeared in 2 trees

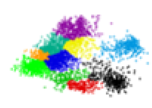


USAGE OF SKOPE-RULES

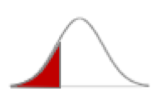
Skope-Rules is suited for **different applications** and can be used:



As a global interpretable model: This is the natural use of Skope-Rules, for a binary classification task. Rules isolate the 1s from 0s.



As a cluster describer: In a clustering task, Skope-Rules is very useful to describe each segment. Each cluster can be post-processed and approximated with a set of interpretable rules.



As a distribution queue describer: Skope-Rules is also relevant to describe how any subsample differs from a population. More precisely, it can be used to describe subsamples defined by highest (or lowest) values of a given variable.
In particular, Skope-Rules has revealed to be effective at understanding highest (or lowest) prediction scores of an other complex classifier.

REFERENCES

[1] *Doshi-Velez et al.* Accountability of AI Under the Law: The Role of Explanation, 2017
[2] *Friedman and Popescu.* Predictive learning via rule ensembles,Technical Report, 2005
[3] *Cohen and Singer.* A simple, fast, and effective rule learner, National Conference on AI, 1999
[4] *Weiss and Indurkhyar.* Lightweight rule induction, ICML, 2000
[5] *Dembczynski, Kotlowski and Slowinski.* Maximum Likelihood Rule Ensembles, ICML, 2008

PERSPECTIVES

- Mathematical formalization
- Improvement of the prediction API when combining rules
- Develop paralleled implementations