Cross-categorized-seeds

Iván Paz

Universitat Politècnica de Catalunya
TOPLAP-Barcelona Hangar.org SEMIMUTICAS National Autonomous University of Mexico

live ⇒ coding music; IMPA January 2019

Motivation

On-the-fly programing imposes cognitive challenges.
Specially using parametric devices (e.g synthesizers) and low level programing languages (e.g SuperCollider).

1. non-linear variation.
2. huge parameter spaces.

It looks like this

```
 Ndef(\x,
      {
         arg freq, freq1, amp, mix, room, damp;
         var sig;
         sig = SinOsc.ar([freq, freq + 1, freq1, freq1 - 2],0, amp);
         sig = FreeVerb.ar(sig, SinOsc.kr(mix), SinOsc.kr(room),
SinOsc.kr(damp))
      })
```

A possible approach is to explore the space and select some parameter combinations that create the structure of the piece. We used these data as a “starting” point for the performance.

However, remembering many combinations again, imposes cognitive challenges.

Obs.

The algorithm is based in three observations:

1. Coding a pice on-the-fly is conducting the parameter values to tell a story. [1]
2. We explore the parameter space to create the structure of a piece.
3. We can use this structure to create labeled data.

[1] e.g "vertical harmony" can be seen as the study of the parametric space where the parameters that take note values.

Proceed as follows:

1. Create the structure of the piece.
2. Explore/select/label the material (parameter combinations).

Then apply inductive-rule-learning which allows to:

1. create an interpretable chart-model-information.
2. suggest-extend the (material) input data.

Example

Piece structure:

calm harsh

p_1	p_2	p_3	category
1.4142	5	0.1	calm
3	3.14	0.1	calm
441.33	662	0.1	harsh

IF p_1 is 1.4142 OR 3 AND p_2 is 5 OR 3.14 AND p_3 is 0.1 THEN calm

Algorithm

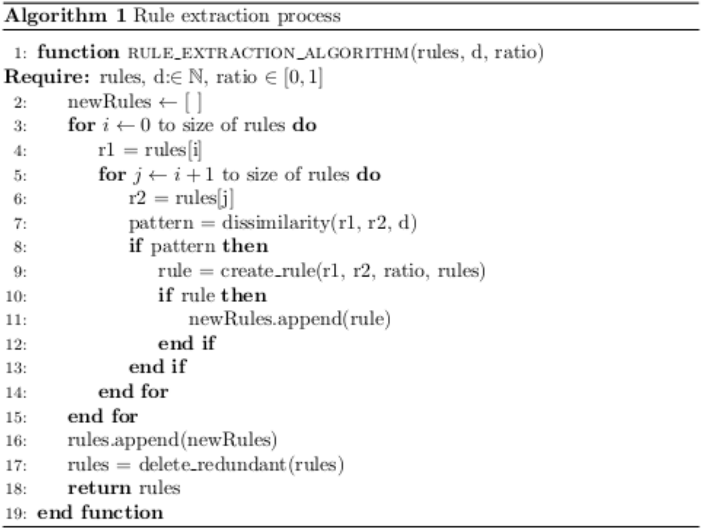
How the algorithm works ?

It search for patterns/regularities in the data.

It suggest new material based on the similarities found.

1. What similar means → similarity finction
2. How the new material is created → create rule function

These are the key subroutines (functions) of the algorithm



Algorithm 1 Rule extraction process

```
function RULE_EXTRACTION_ALGORITHM(rules, d, ratio)
Require: rules, d:$\in \mathbb{N}$, ratio $\in [0, 1]$
    newRules ← [ ]
    for $i \leftarrow 0$ to size of rules do
        r1 = rules[i]
        for $j \leftarrow i + 1$ to size of rules do
            r2 = rules[j]
            pattern = dissimilarity(r1, r2, d)
            if pattern then
                rule = create_rule(r1, r2, ratio, rules)
                if rule then
                    newRules.append(rule)
                end if
            end if
        end for
    end for
    rules.append(newRules)
    rules = delete_redundant(rules)
    return rules
end function
```

Figure: The RuLer algorithm

The similarity function

$$s(r_1, r_2) \leq d$$

This cam be any function (e.g domain specific or generic).

Depending on the function it can or can not satisfy certain properties like the triangle inequality or producing an output that does not depend on the order of the input data.

similarity(r_1,r_2) $\leq$ d

rule	p1	p2	category	-
r_1	$\{1\}$	$\{5\}$	harsh	Hamming distance ≤ 1
r_2	$\{1\}$	$\{3\}$	harsh	-
$r_{1,2}$	$\{1\}$	$\{3,5\}$	harsh	-
$r_{1,2}$	$\{1\}$	$\{3,5\}$	harsh	Hamming distance ≤ 1
r_*	$\{1,11\}$	$\{3,5,7\}$	harsh	-

rule	p1	p2	category	-
$r_{1,2}$	$\{1\}$	$\{3,5\}$	harsh	non-empty intersections ≤ 1
r_*	$\{1,11\}$	$\{3,5,7\}$	harsh	-
$r_{1,2,*}$	$\{1,11\}$	$\{1,3,5,7\}$	harsh	-

Note that when a pattern is found the rule is created by taking the unions of the variable values.

Create rules function

Controls the “risk” on the generalization process.
e.g to create a new rule take the unions of the variable values.
e.g demanding not to generalize beyond certain proportion of the input data.

How much new material?

```
0:
[[20], [150], [0.25], [101], [0.3], [102], [0.5], [0.33], [20], [150], [0.01], [101], [0.3], [102], [0.5], [0.01], main]
[[200], [150], [0.28], [150], [0.6], [160], [0.4], [0.38], [200], [150], [0.01], [150],[0.6], [160], [0.4], [0.01], main]
[[100], [100], [0.25], [100], [0.6], [102], [0.4], [0.33], [100], [100], [0.25], [100],[0.6], [102], [0.4], [0.33], main]
12:
[[200, 100, 20], [100, 150], [0.25, 0.28], [100, 101, 150], [0.3, 0.6], [160, 102], [0.5, 0.4], [0.33, 0.38], [200, 100,
20], [100, 150], [0.25, 0.01], [100, 101, 150], [0.3, 0.6], [160, 102], [0.5, 0.4], [0.33, 0.01], main]
```

Possibilities and limitations

1. Create new material.
2. The consistency of the new material depends on the linear/non-linear variation of the spaces.
3. Interpretable (with some restrictions).
4. Metric or similarity/dissimilarity functions and the create rule function require expert knowledge for its selection (some general choises that may work).
5. Current work include implementing on-the-fly tools.

“Thank you”.postln; ivanpaz@cs.upc.edu