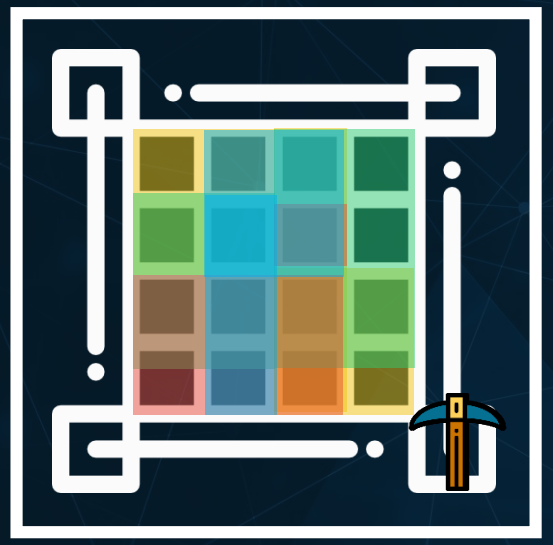
ECMLPKDD'18

September 13, 2018

ECML/PKDD 2018



REFINE&MINE:

ANYTIME SUBGROUP DISCOVERY IN NUMERICAL DOMAINS WITH GUARANTEES

AUTHORS.

Aimene BELFODIL Adnene BELFODIL

Mehdi KAYTOUE

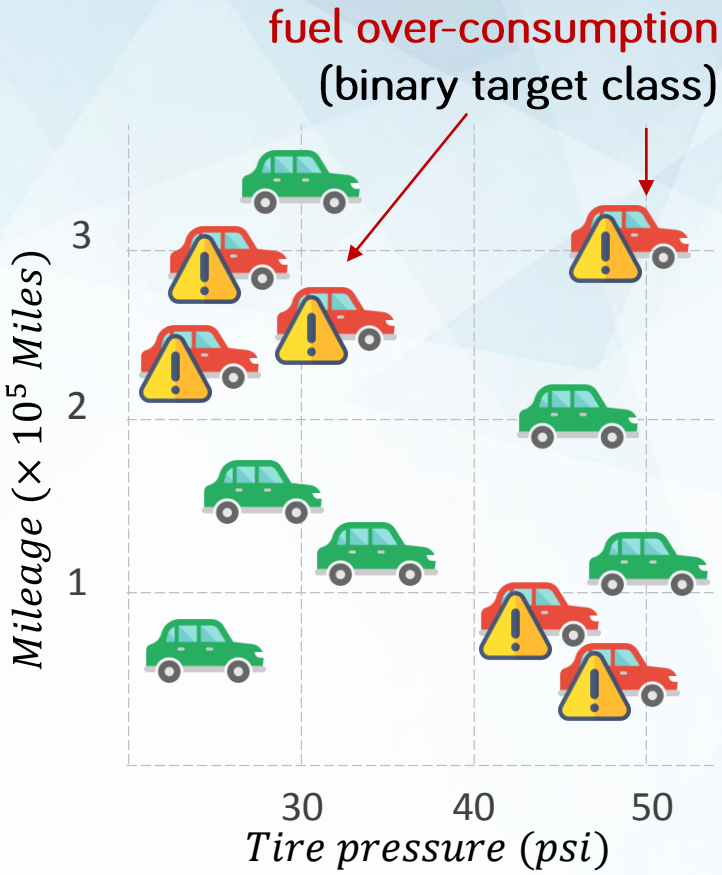
\°/ All icons used are courtesy of flaticon.com

LIRIS INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON UCBL Lyon 1 cnrs

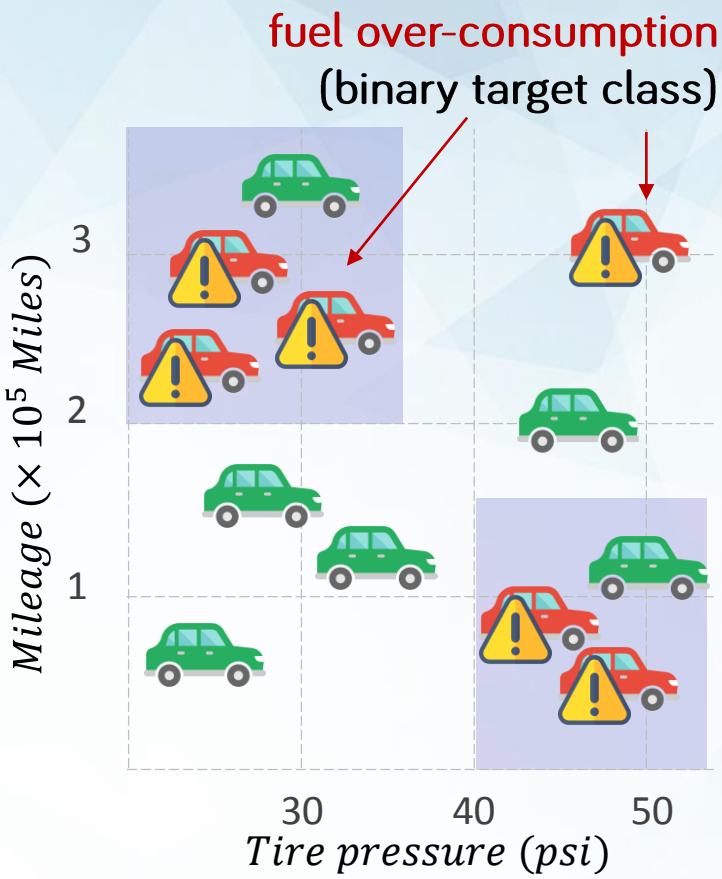
INTRODUCTION

PROBLEM DEFINITION

Subgroup Discovery



Subgroup Discovery



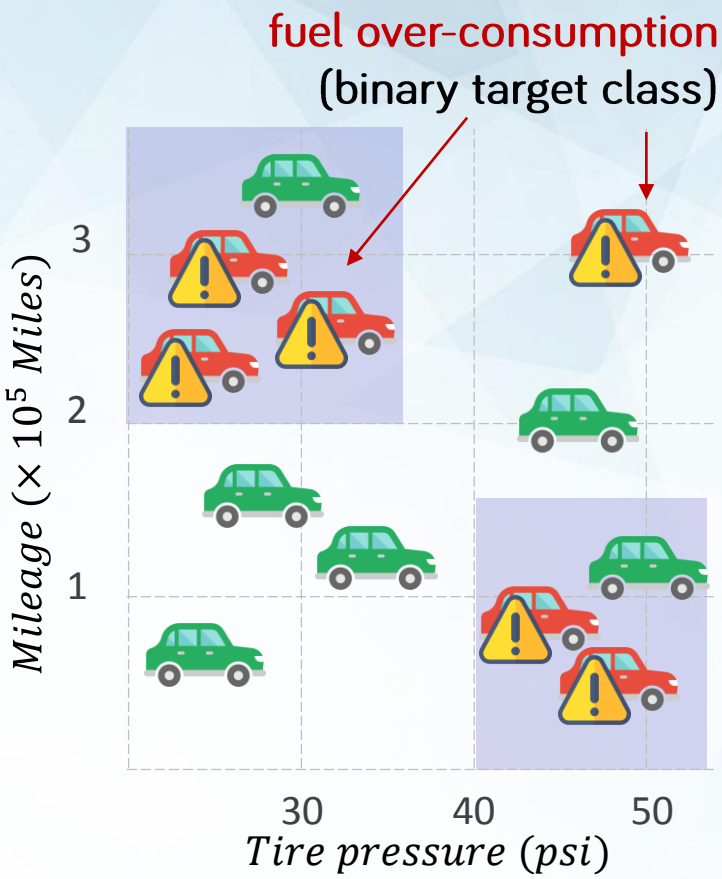
Hypothesis.

Fuel over-consumption is fostered when:

1. $Tire\ pressure \leq 35$ psi **and** $Milage \geq 2 \times 10^5\ miles$
2. $Tire\ pressure \geq 40$ psi **and** $Milage \leq 1.7 \times 10^5\ miles$

Subgroups - Discriminant Interval patterns

Subgroup Discovery



Hypothesis.

Fuel over-consumption is fostered when:

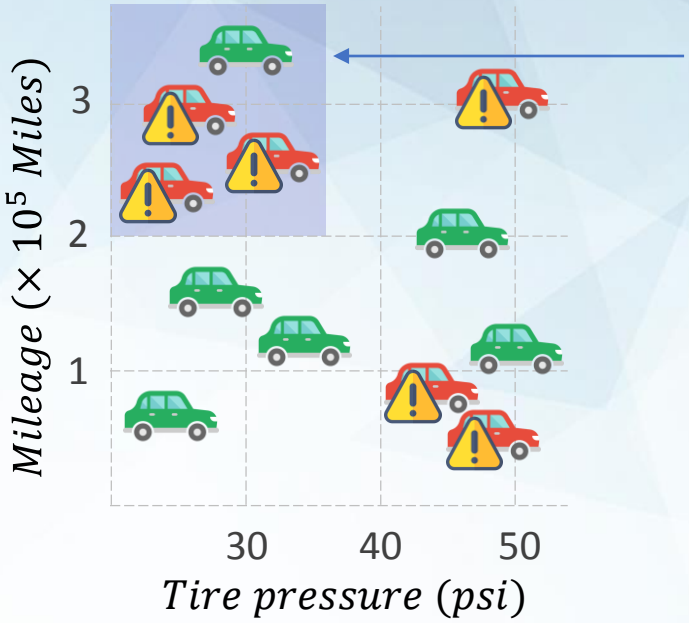
1. $Tire\ pressure \leq 35$ psi **and** $Milage \geq 2 \times 10^5\ miles$
2. $Tire\ pressure \geq 40$ psi **and** $Milage \leq 1.7 \times 10^5\ miles$

Subgroups - Discriminant Interval patterns

Subgroup discovery. The task of discovering patterns that accurately discriminate the target class from others.

*We are interested only in datasets where all the attributes are numerical.

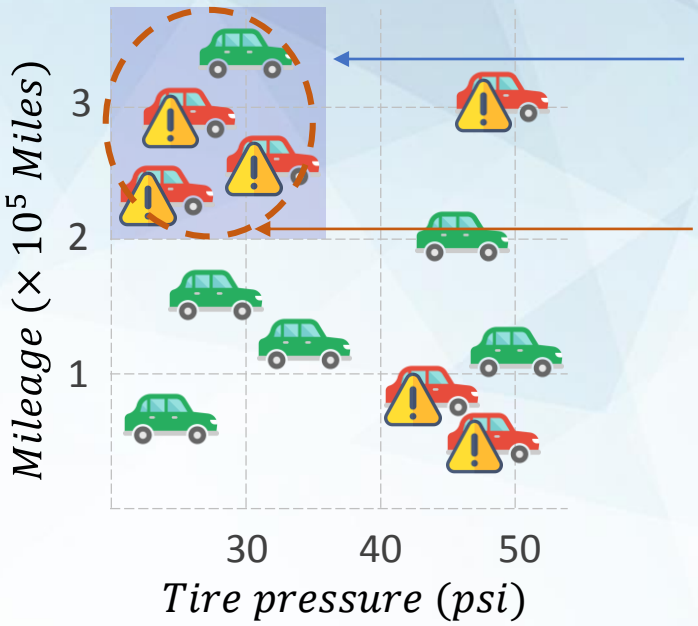
Preliminary notions



Interval pattern intent. Conjunction of interval restrictions:

E.g. $x \in [0, 35]$ **and** $y \in [2, +\infty)$

Preliminary notions

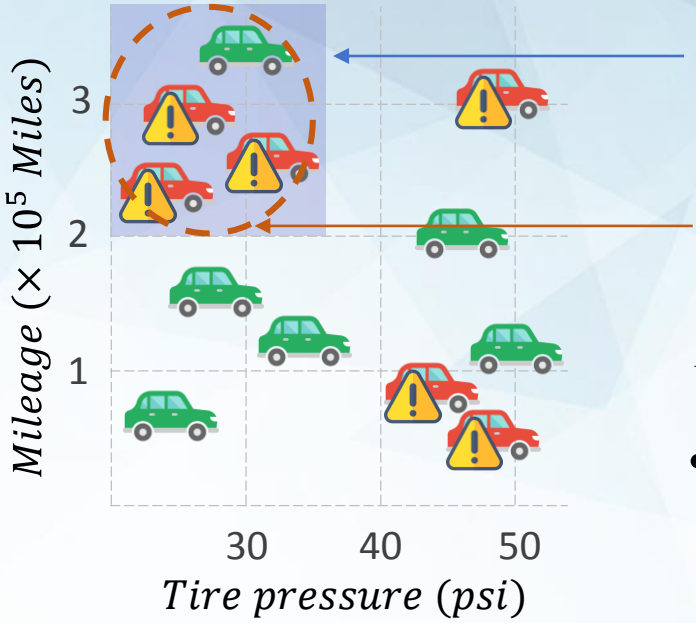


Interval pattern intent. Conjunction of interval restrictions:

E.g. $x \in [0, 35]$ **and** $y \in [2, +\infty)$

Interval pattern extent. Set of instances for which the pattern hold.

Preliminary notions



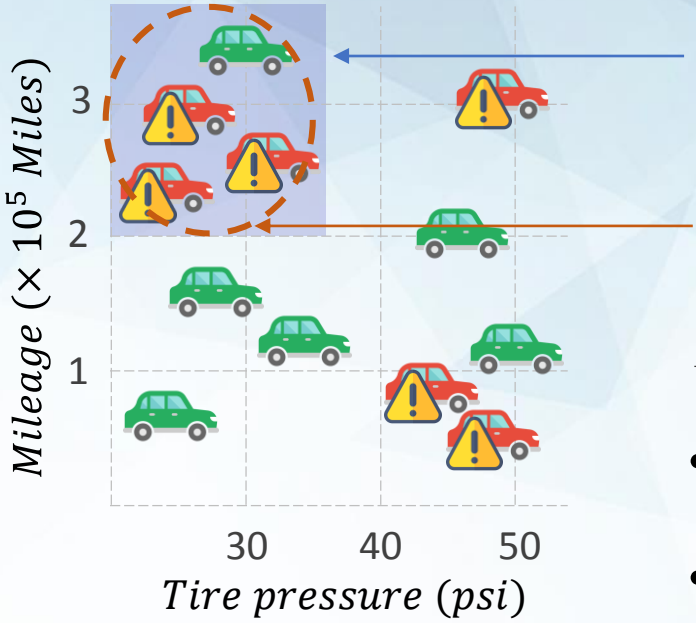
Interval pattern intent. Conjunction of interval restrictions:

E.g. $x \in [0, 35]$ **and** $y \in [2, +\infty)$

Interval pattern extent. Set of instances for which the pattern hold.

- A Quality measure ϕ evaluates the discriminative power of patterns.

Preliminary notions



Interval pattern intent. Conjunction of interval restrictions:

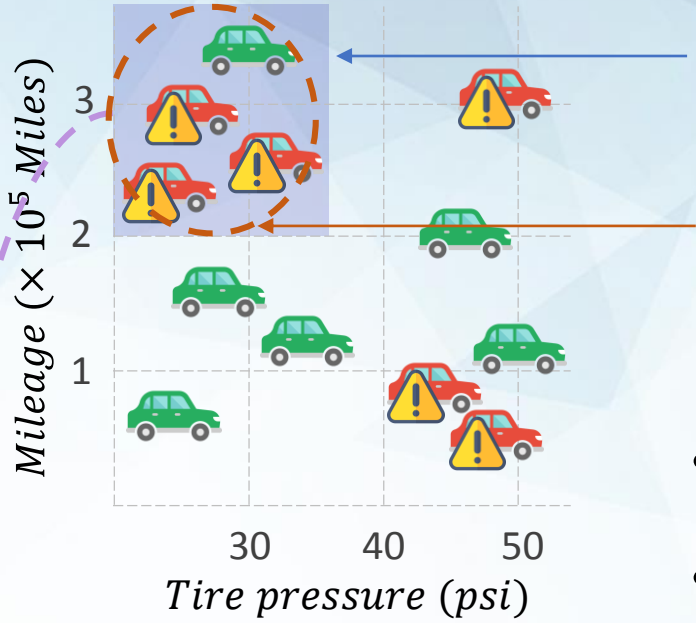
E.g. $x \in [0, 35]$ **and** $y \in [2, +\infty)$

Interval pattern extent. Set of instances for which the pattern hold.

- A Quality measure ϕ evaluates the discriminative power of patterns.
- We use measures increasing with tpr^* and decreasing with the fpr^*.

* *True Positive Rate* * *False Positive Rate*

Preliminary notions



Interval pattern intent. Conjunction of interval restrictions:

E.g. $x \in [0, 35]$ **and** $y \in [2, +\infty)$

Interval pattern extent. Set of instances for which the pattern hold.

- A Quality measure ϕ evaluates the discriminative power of patterns.
- We use measures increasing with tpr^* and decreasing with the fpr^*.

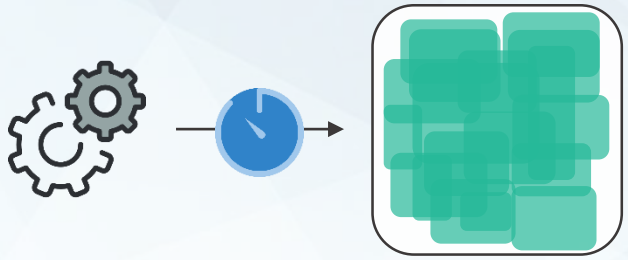
* *True Positive Rate* * *False Positive Rate*

$$\phi(p) = \frac{3}{6} - \frac{1}{6} = 0.33$$

$$\phi(pattern) = tpr(pattern) - fpr(pattern)$$

*Informedness is order-equivalent to the well-known WRAcc measure.

State-of-the-art (a small one)



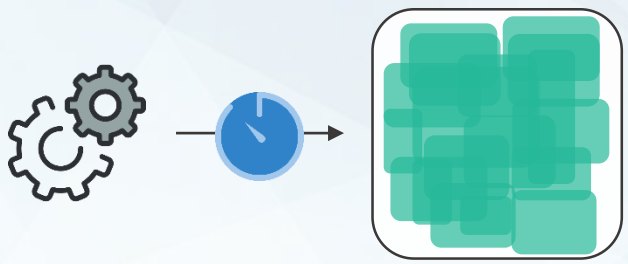
Exhaustive Search

Generally unfeasible.

e.g. A 200 × 3 dataset has up to 8 Billions Interval Patterns.

M. Kaytoue, S. O. Kuznetsov, A. Napoli. Revisiting Numerical Pattern Mining with FCA. In IJCAI 2011.

State-of-the-art (a small one)

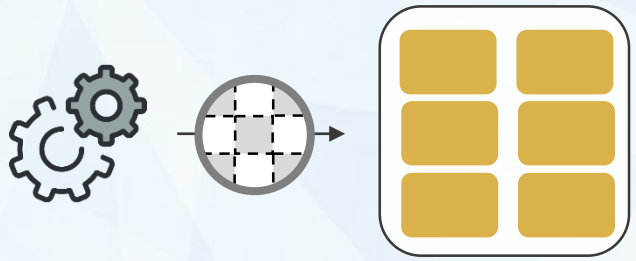


Exhaustive Search

Generally unfeasible.

e.g. A 200 × 3 dataset has up to 8 Billions Interval Patterns.

M. Kaytoue, S. O. Kuznetsov, A. Napoli. Revisiting Numerical Pattern Mining with FCA. In IJCAI 2011.



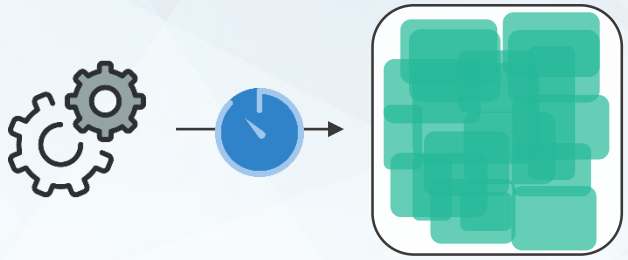
Prior Discretization

Loss of information.

e.g. overlapping intervals are ignored. Look for [1,2] and [2,3] patterns but ignore [1,3].

M. Artzmueller, F. Puppe. SD-Map – A Fast Algorithm for Exhaustive Subgroup Discovery. In PKDD 2006.

State-of-the-art (a small one)

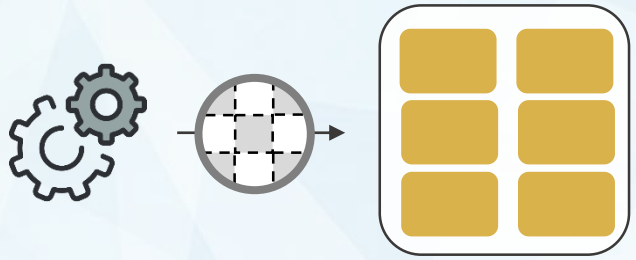


Exhaustive Search

Generally unfeasible.

e.g. A 200 × 3 dataset has up to 8 Billions Interval Patterns.

M. Kaytoue, S. O. Kuznetsov, A. Napoli. Revisiting Numerical Pattern Mining with FCA. In IJCAI 2011.

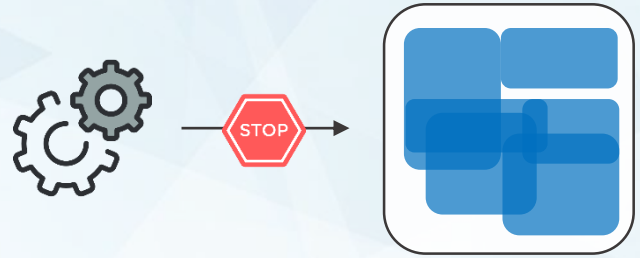


Prior Discretization

Loss of information.

e.g. overlapping intervals are ignored. Look for [1,2] and [2,3] patterns but ignore [1,3].

M. Artzmueller, F. Puppe. SD-Map – A Fast Algorithm for Exhaustive Subgroup Discovery. In PKDD 2006.



Heuristics and sampling

No guarantee on the outputted patterns at the end or upon interruption

G. Bosc, J.F. Boulicaut, C. Raïssi, M. Kaytoue. Anytime discovery of a diverse set of patterns with MCTS. In DMKD 2018.

Problem Statement

Algorithm searching for discriminant interval patterns ...

Few patterns
after 5”

More patterns
after 1’

Problem Statement

Algorithm searching for discriminant interval patterns ...

Few patterns after 5”

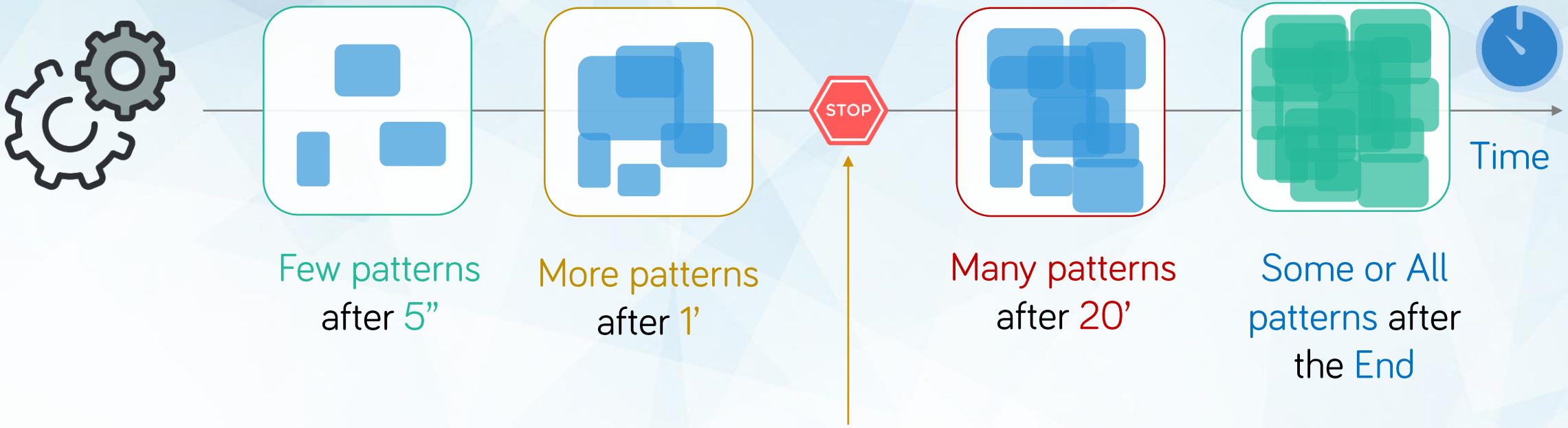
More patterns after 1’

Many patterns after 20’

Some or All patterns after the End

Problem Statement

Algorithm searching for discriminant interval patterns ...



Few patterns after 5”

More patterns after 1’

Many patterns after 20’

Some or All patterns after the End

All state-of-the-art subgroup discovery algorithms provide no guarantees on the outputted discriminant interval patterns when interrupted during execution

Contributions (1)

Propose the new algorithm

Refine&Mine

Anytime Algorithm

Provide patterns anytime

with two guarantees on them

Contributions (1)

Propose the new algorithm

Refine&Mine

Anytime Algorithm

Provide patterns anytime with two guarantees on them

Guarantee 1: Accuracy

Bound the quality difference between the best possible pattern and the best already found pattern

Contributions (1)

Propose the new algorithm

Refine&Mine

Anytime Algorithm

Provide patterns anytime with two guarantees on them

* The more time it is given, the lower are the bounds.

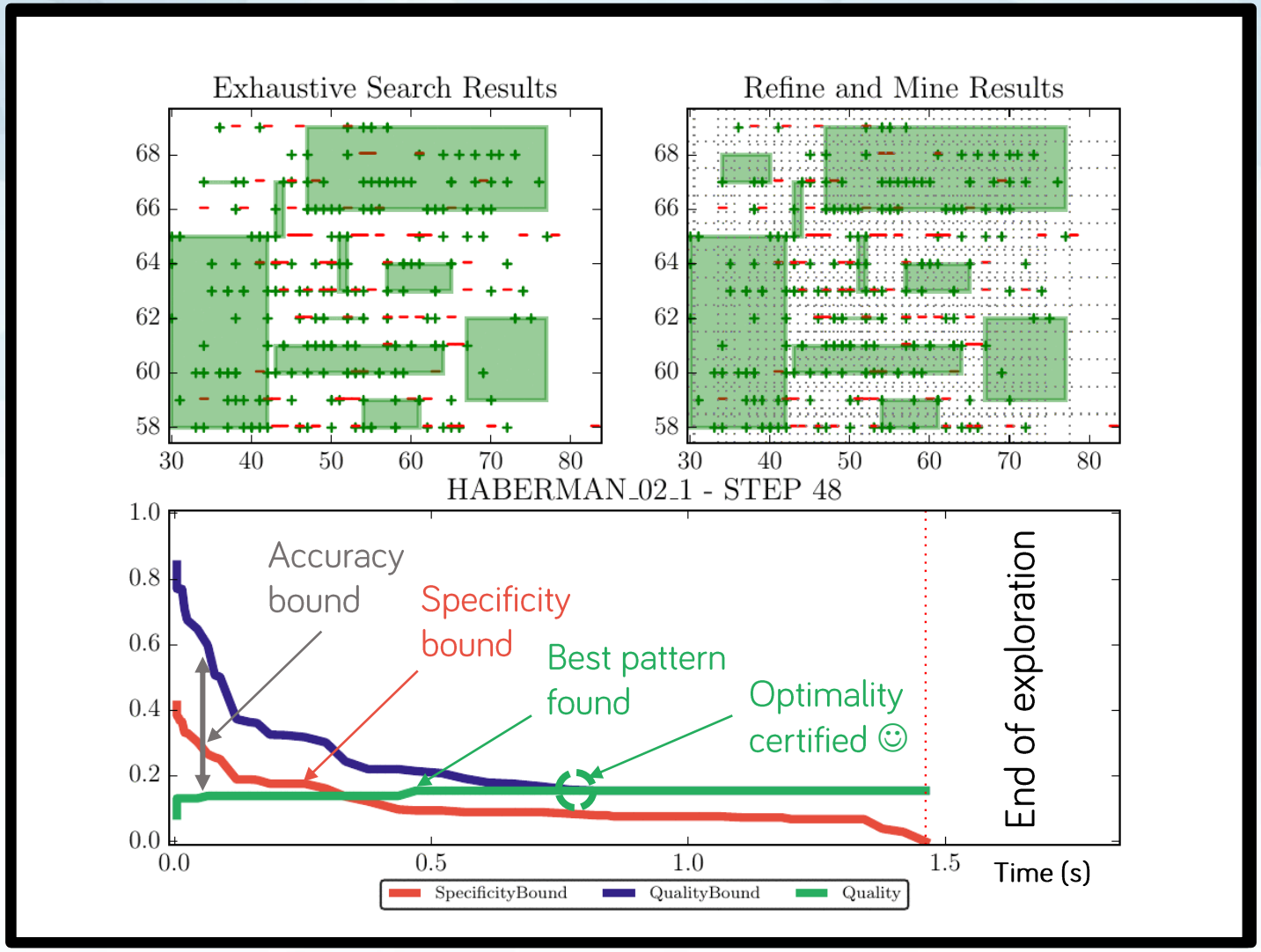
Guarantee 1: Accuracy

Bound the quality difference between the best possible pattern and the best already found pattern

Guarantee 2: Specificity

Bound the distance of found patterns to ground truth patterns to ensure diversity of found patterns.

Contributions (2)



OUTLINE

- Overview and Contributions
- Refine&Mine in a nutshell
- Guarantees computation
- Building blocks of Refine&Mine
- Wrap-up and concluding remarks

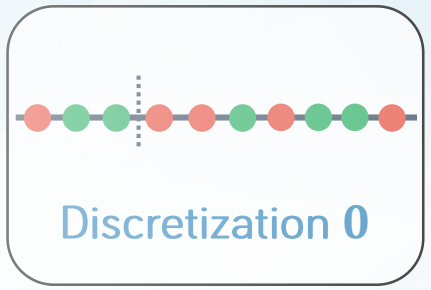
REFINE&MINE

REFINE&MINE: In a Nutshell (1)

Refine&Mine

Sequence of discretization refinements and mining patterns within the discretization.

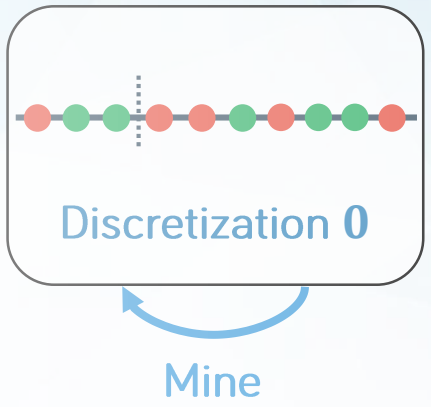
REFINE&MINE: In a Nutshell (1)



Refine&Mine

Sequence of discretization refinements and mining patterns within the discretization.

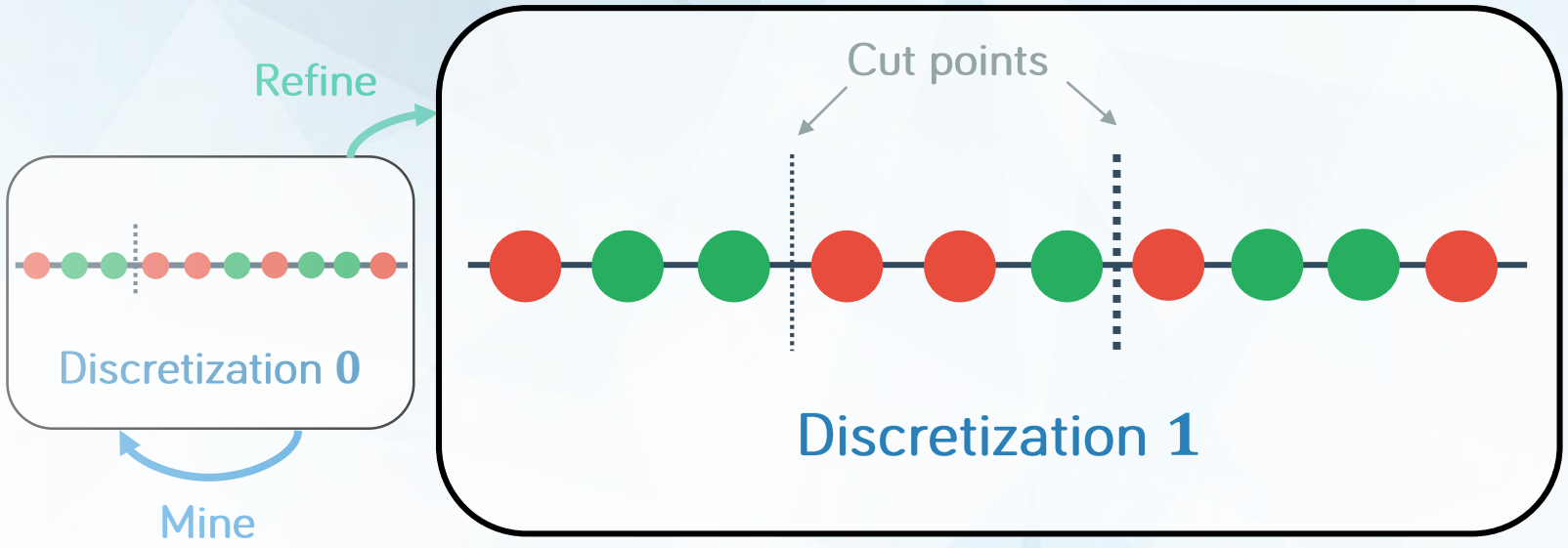
REFINE&MINE: In a Nutshell (1)



Refine&Mine

Sequence of discretization refinements and mining patterns within the discretization.

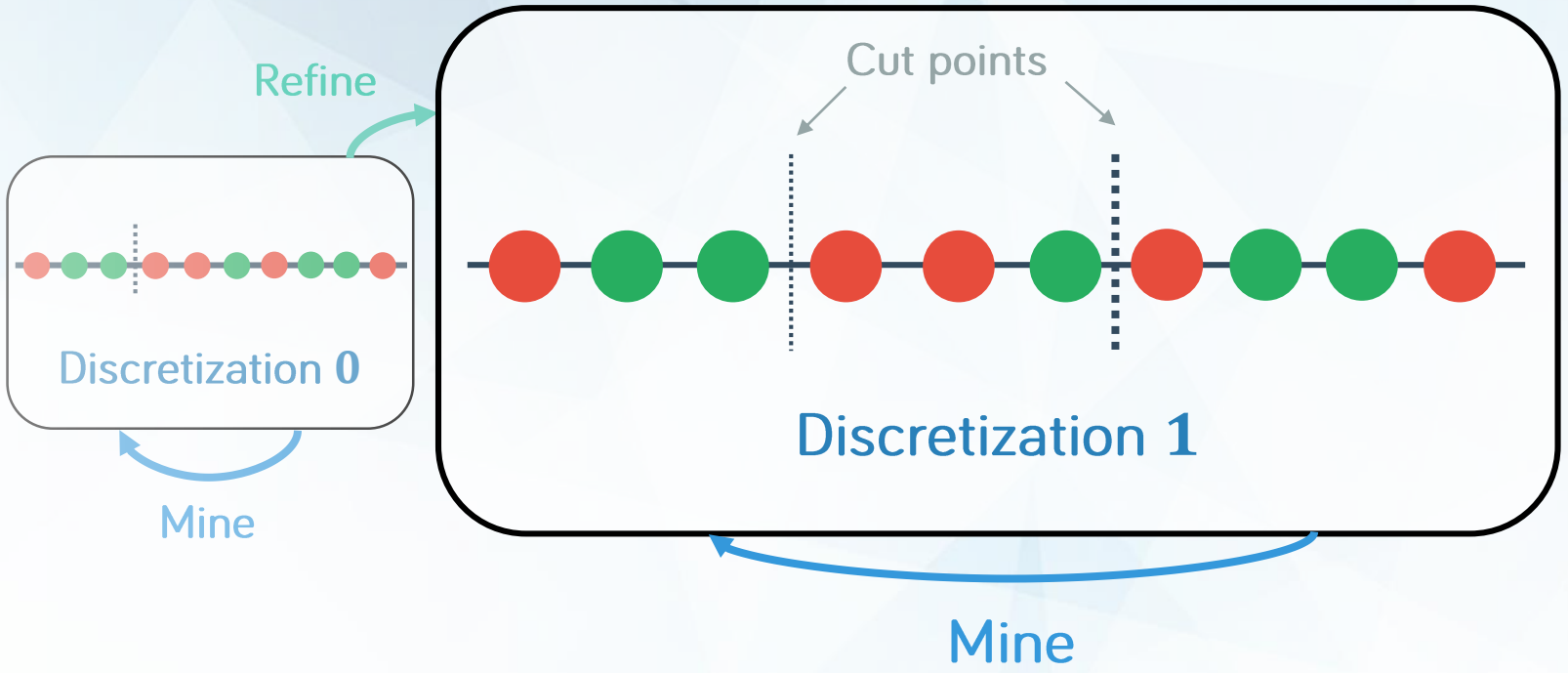
REFINE&MINE: In a Nutshell (1)



Refine&Mine

Sequence of discretization refinements and mining patterns within the discretization.

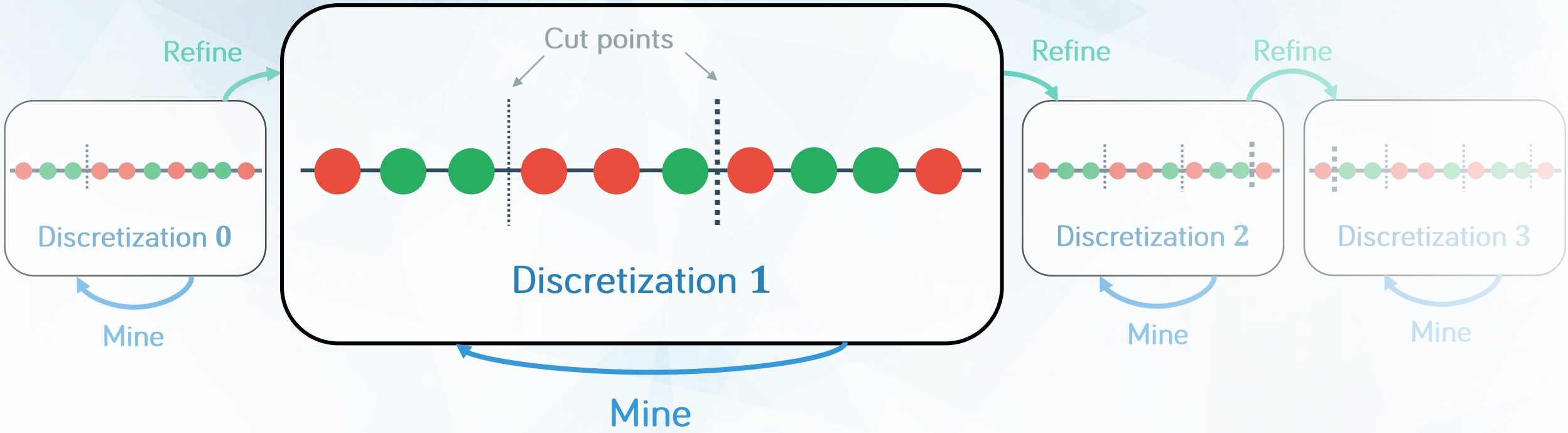
REFINE&MINE: In a Nutshell (1)



Refine&Mine

Sequence of discretization refinements and mining patterns within the discretization.

REFINE&MINE: In a Nutshell (1)

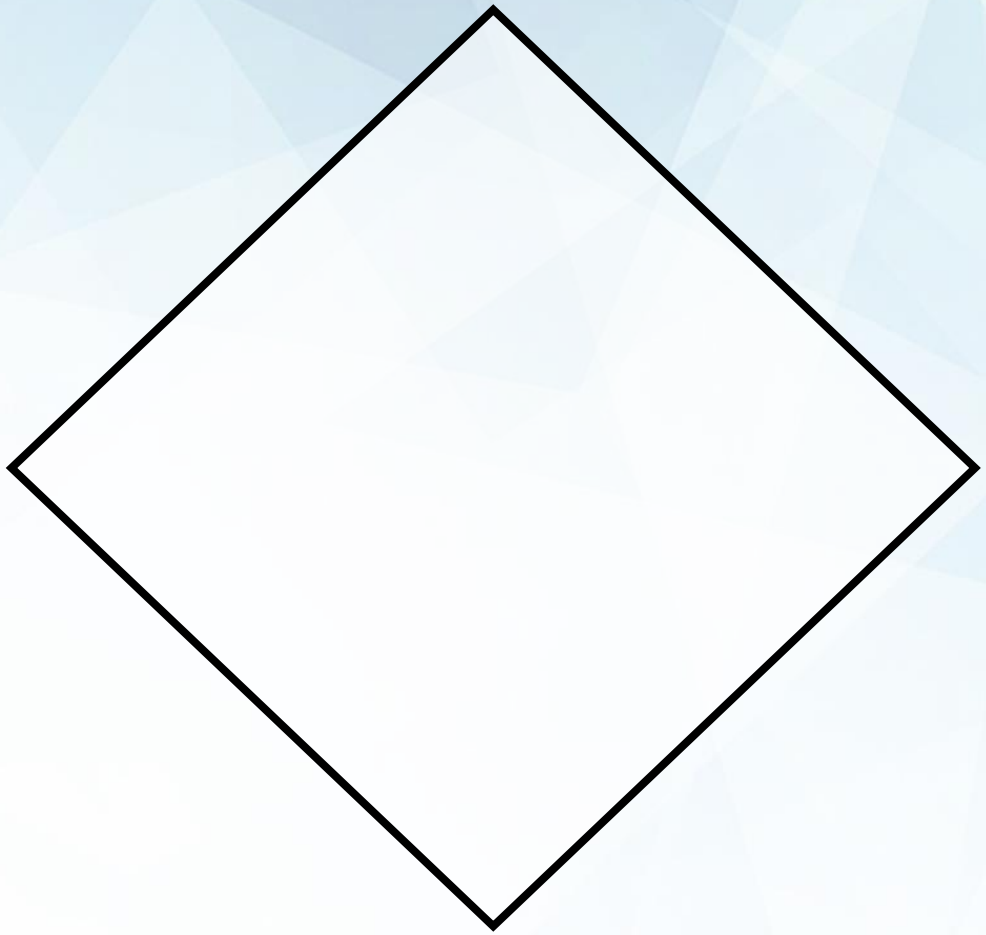


Refine&Mine

Sequence of discretization refinements and mining patterns within the discretization.

REFINE&MINE: In a Nutshell (2)

Refine

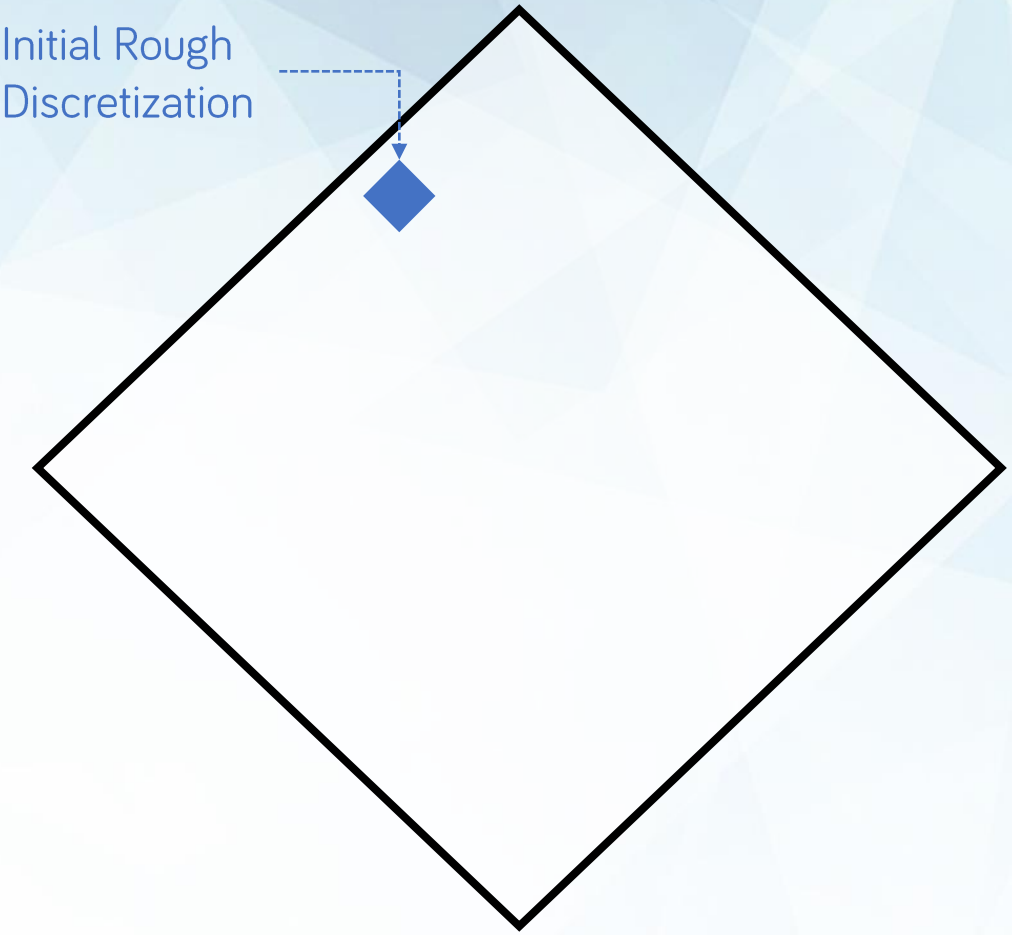


Mine

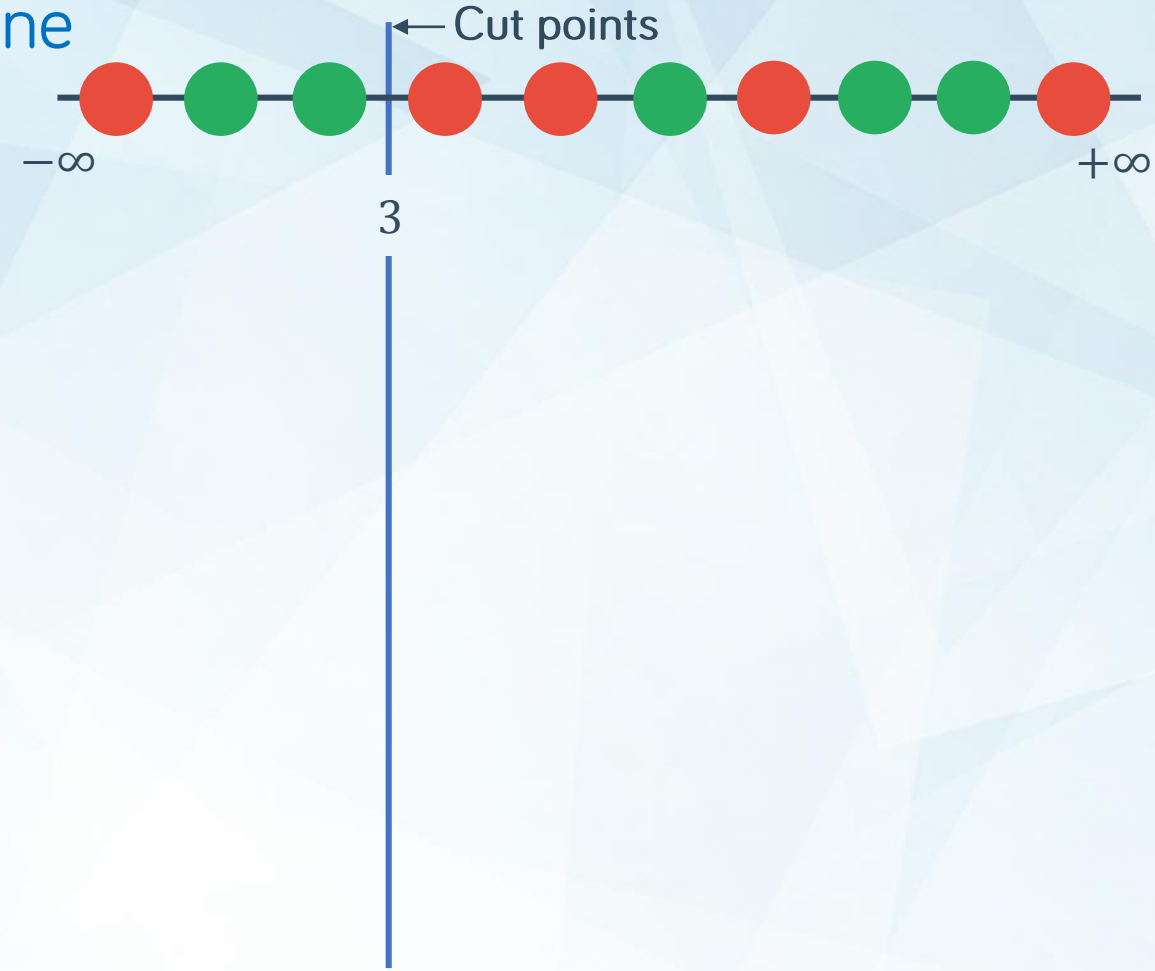
REFINE&MINE: In a Nutshell (2)

Refine

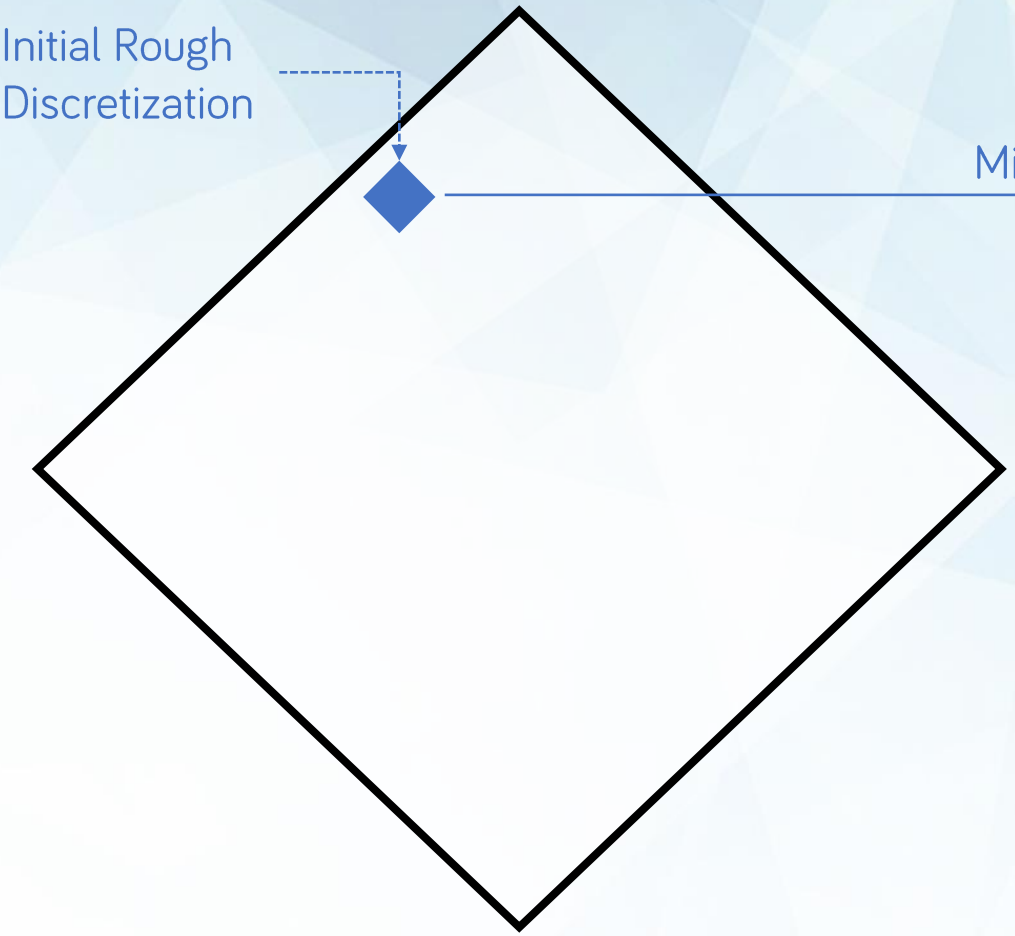
Initial Rough Discretization

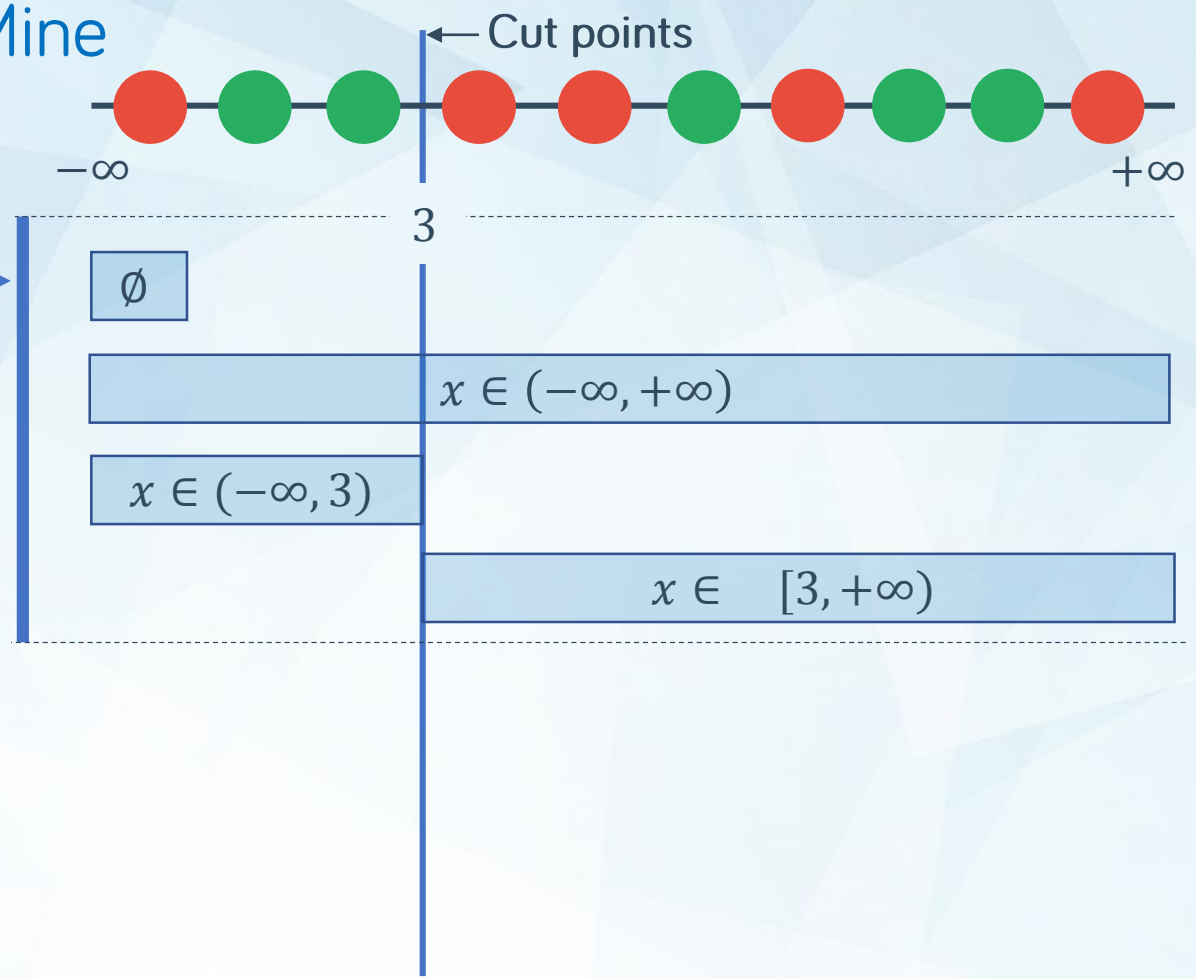


Mine

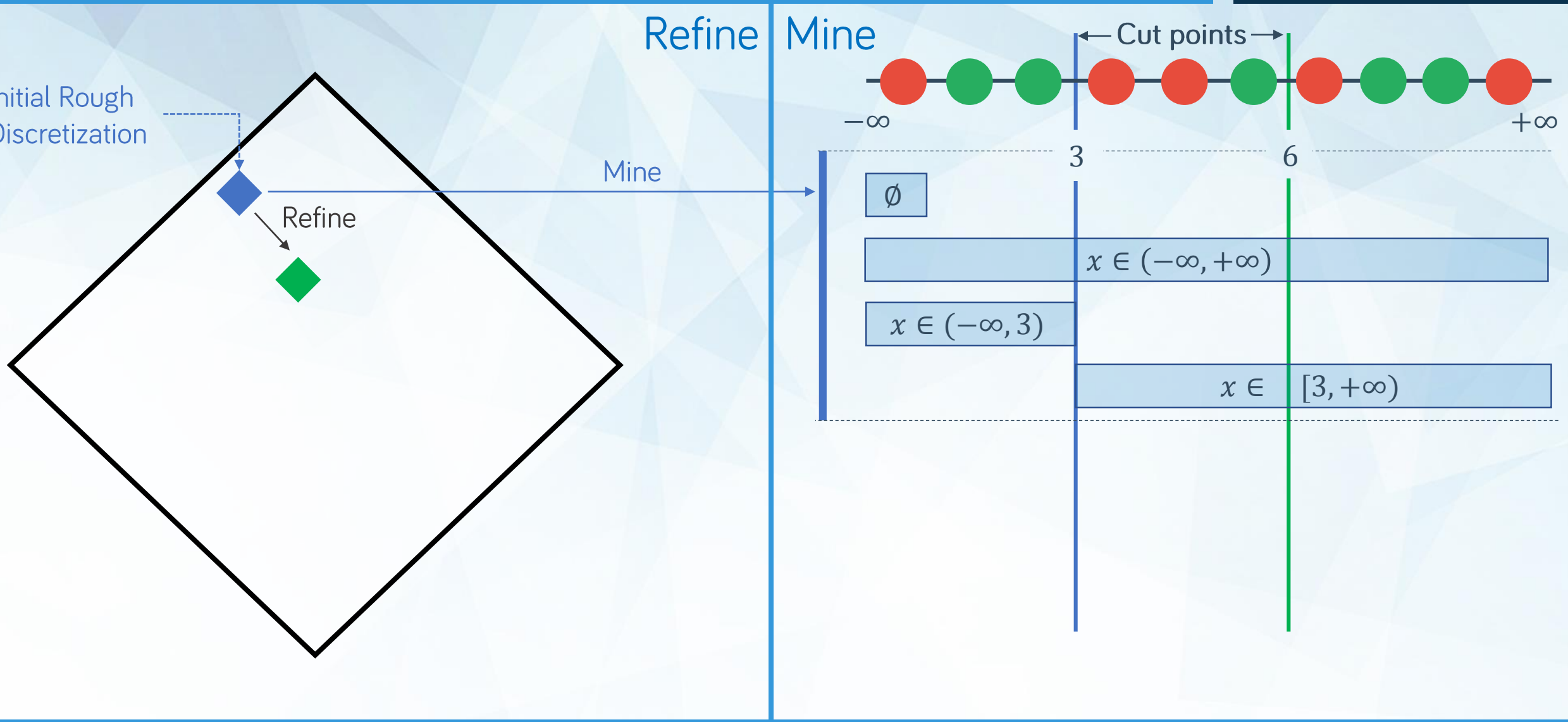


REFINE&MINE: In a Nutshell (2)

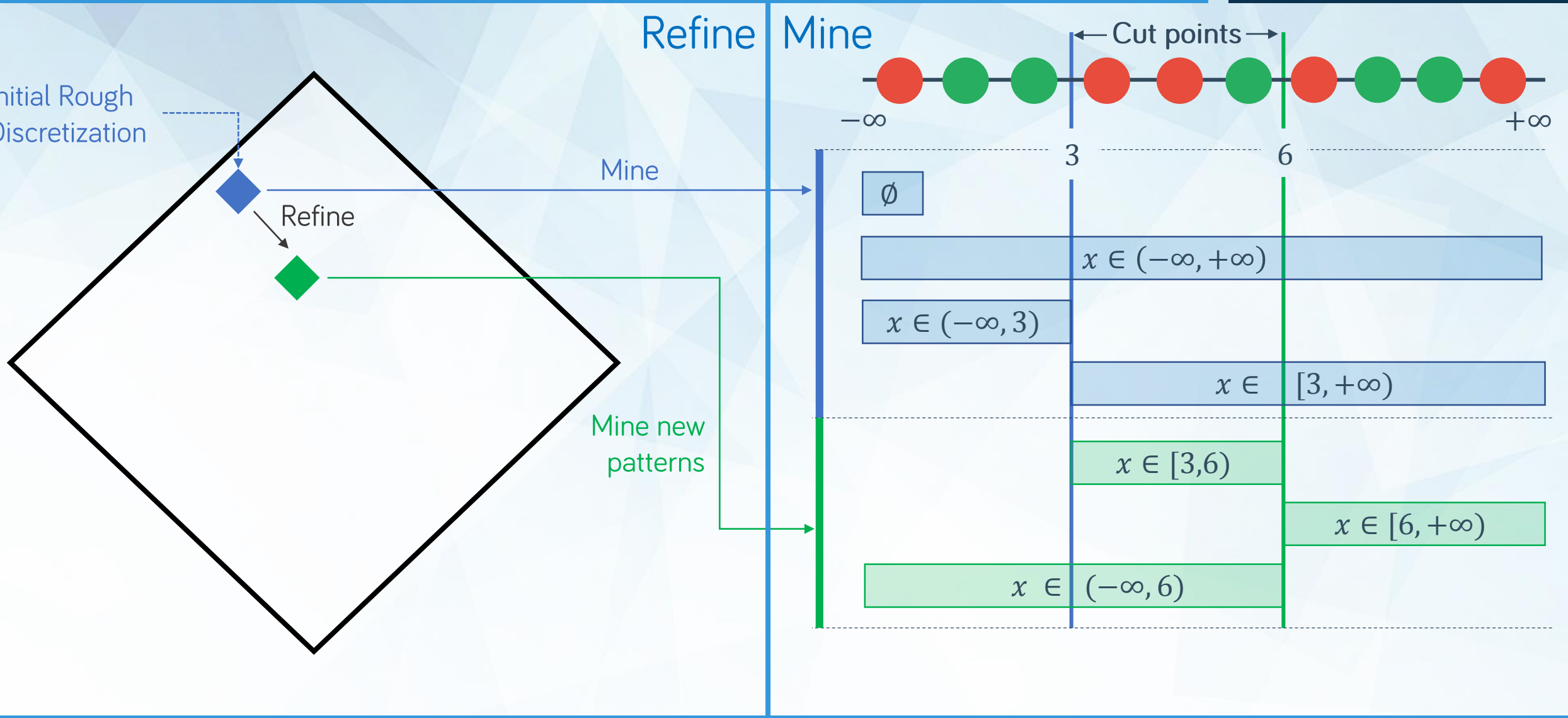




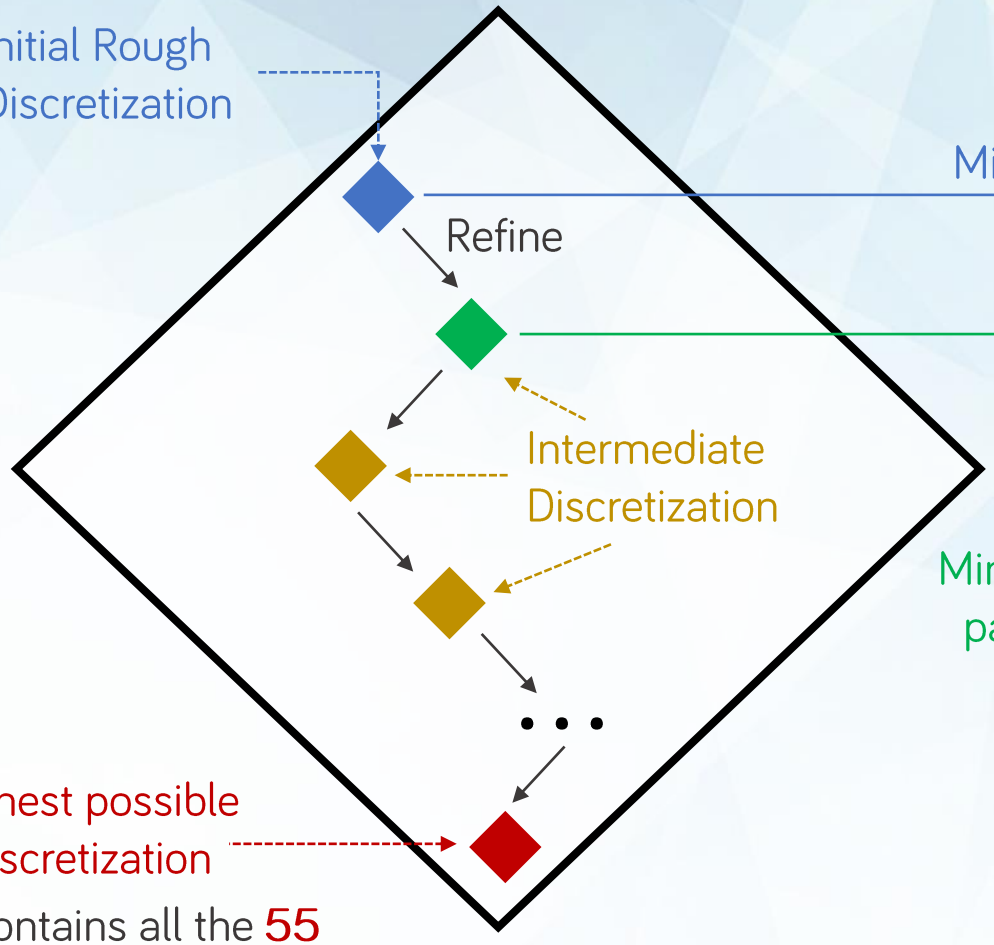
REFINE&MINE: In a Nutshell (2)



REFINE&MINE: In a Nutshell (2)



REFINE&MINE: In a Nutshell (2)

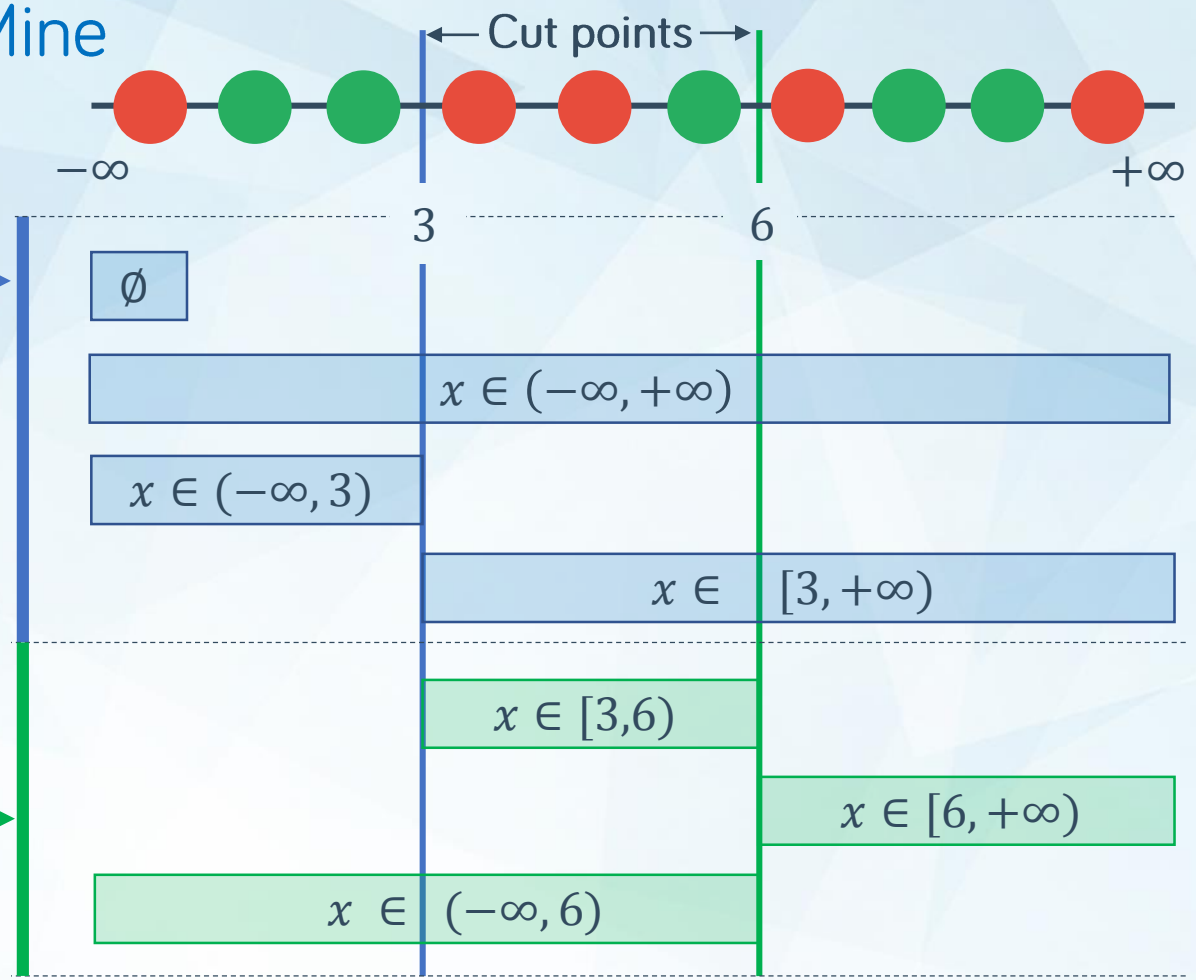


Contains all the 55 interval patterns

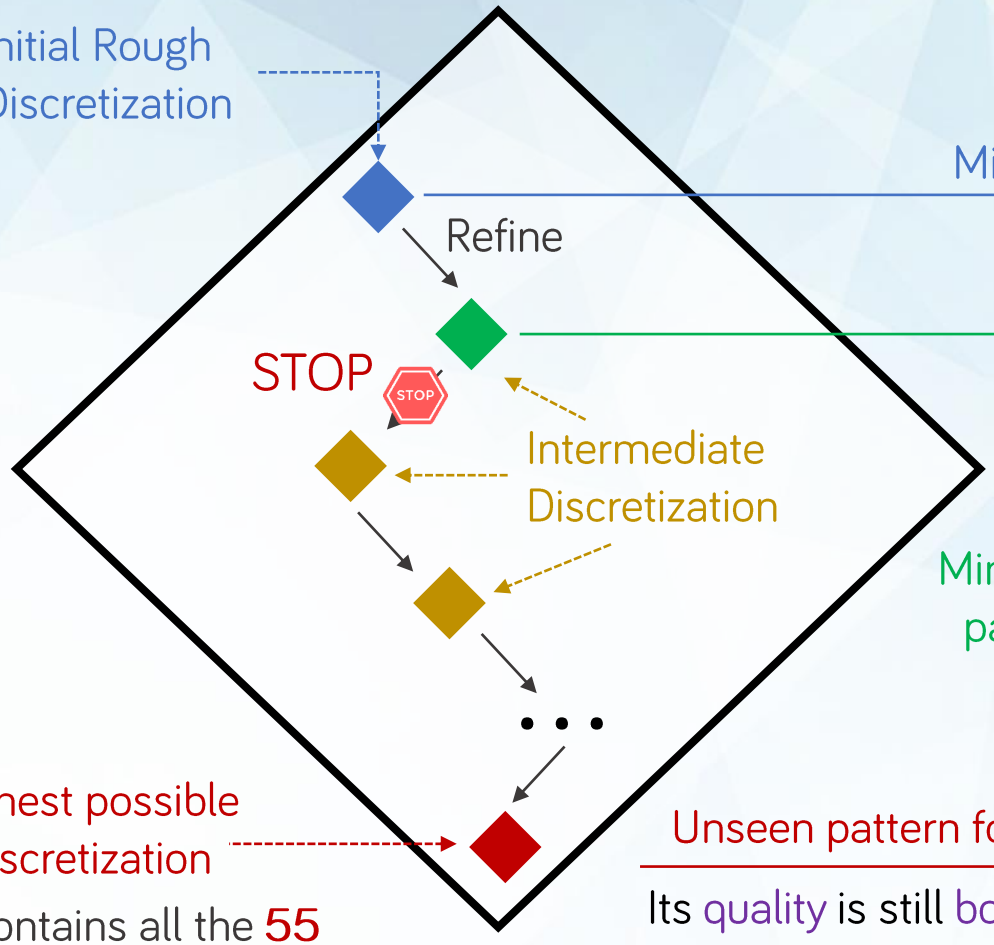
Refine | Mine

Mine

Mine new patterns



REFINE&MINE: In a Nutshell (2)

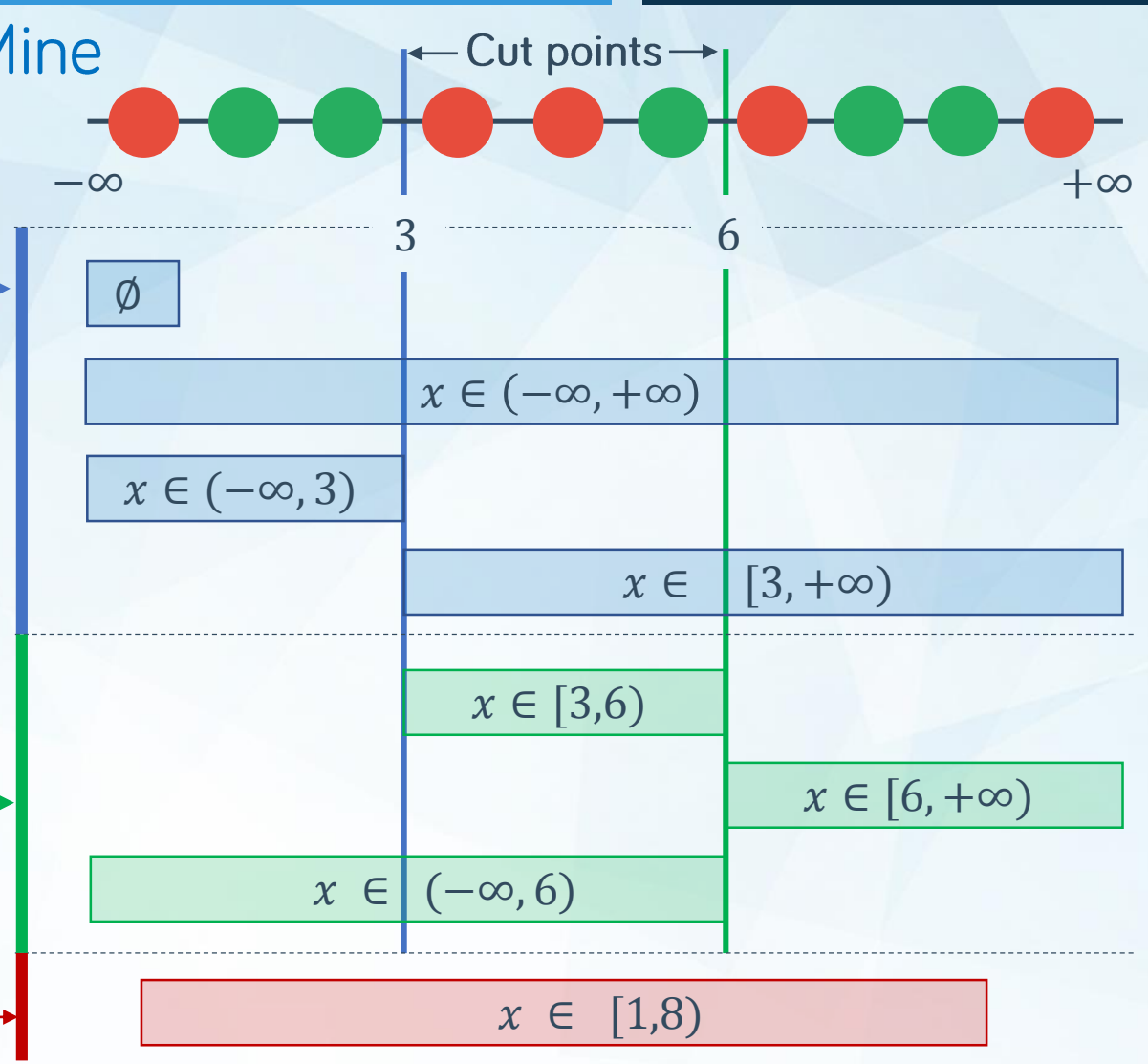


Mine

Mine new patterns

Unseen pattern for now

Its quality is still bounded by a guarantee.



Guarantees provided by Refine&Mine

Accuracy Guarantee

e.g. Informedness

$$\text{Accuracy} = \phi(\text{best pattern in the dataset}) - \phi(\text{best found pattern})$$

Accuracy Guarantee

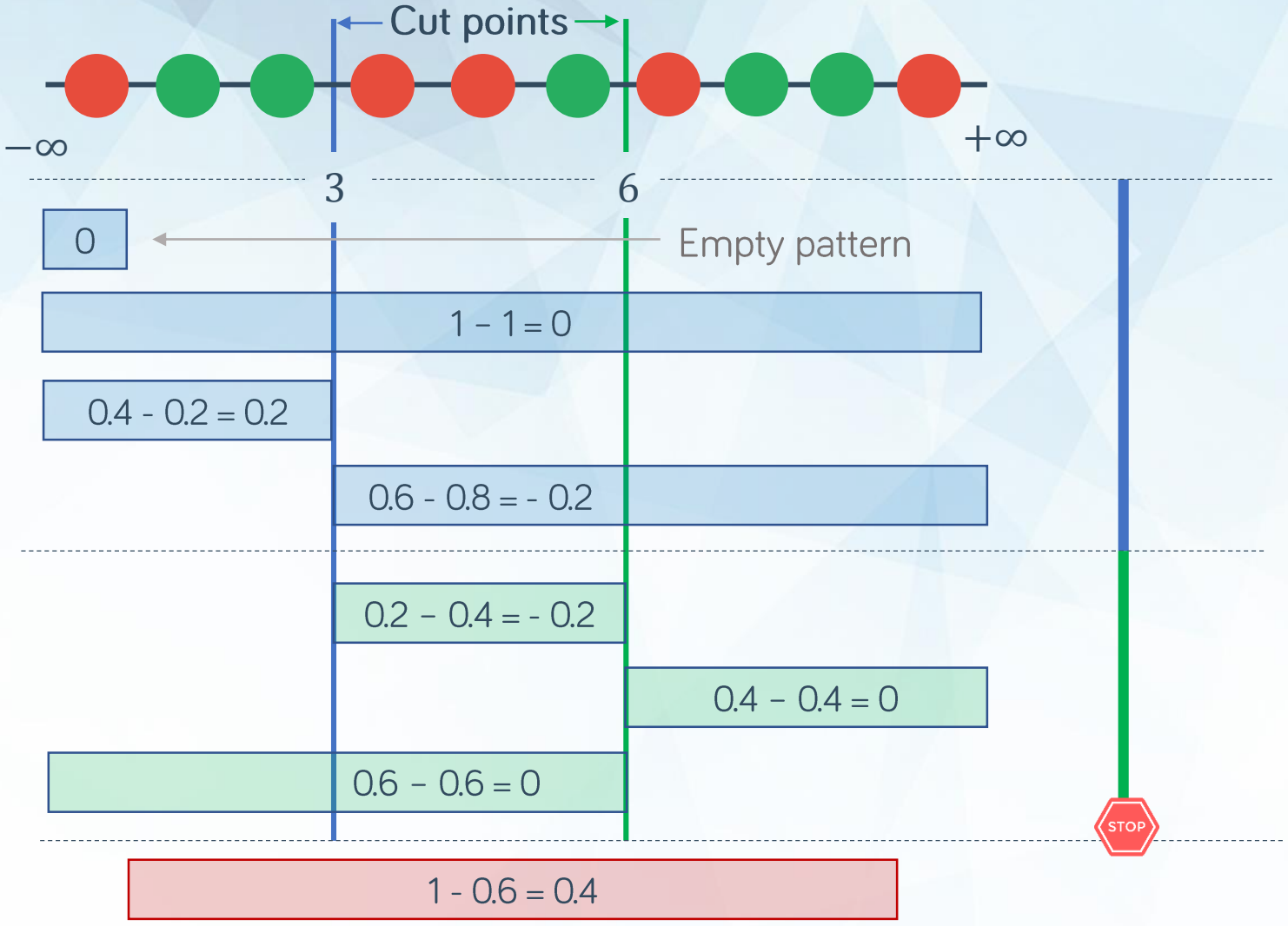
e.g. Informedness

$$\text{Accuracy} = \phi(\text{best pattern in the dataset}) - \phi(\text{best found pattern})$$

$$\text{Accuracy} \leq \text{Accuracy bound}$$

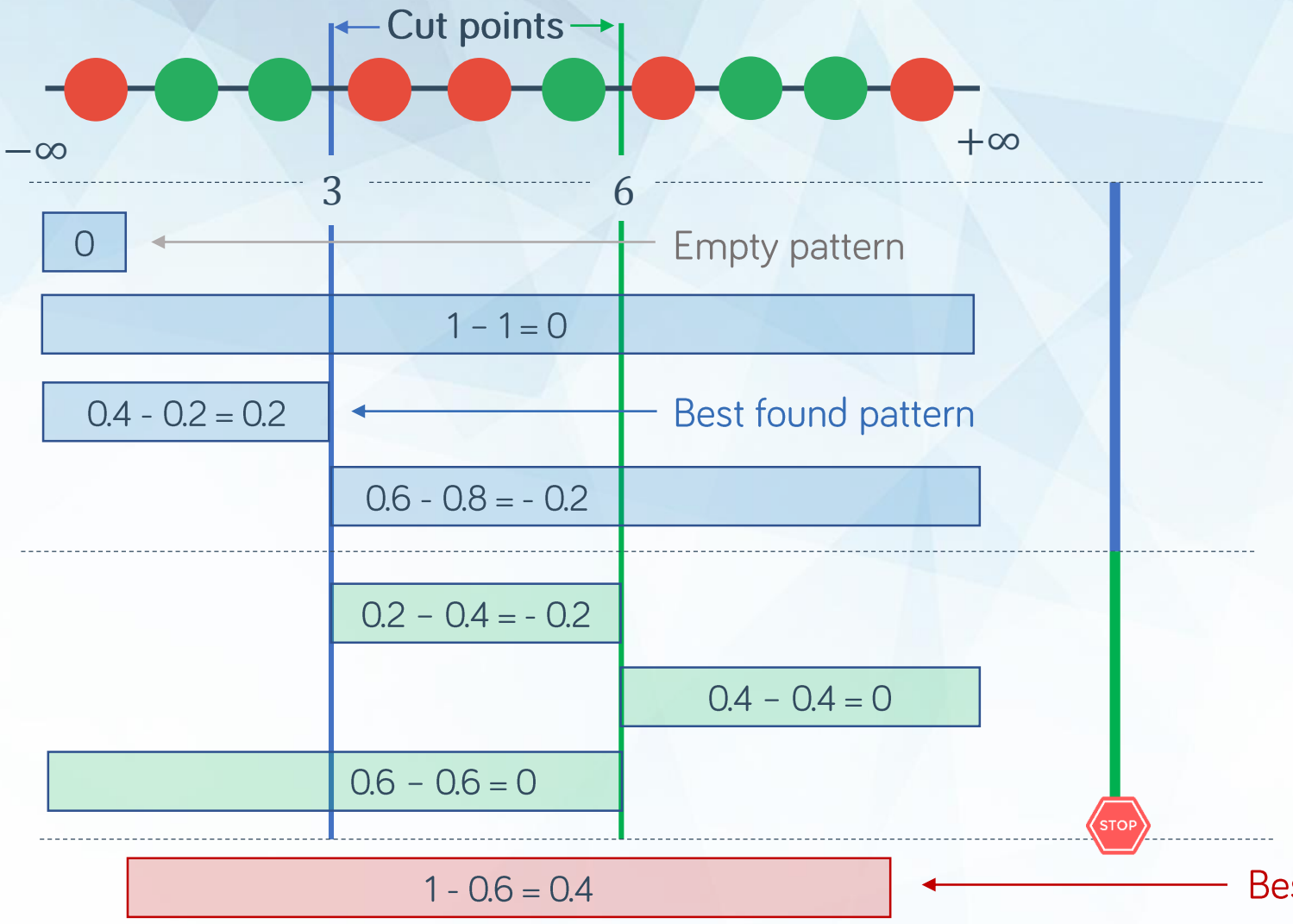
Refine&Mine computes this bound using only already found patterns.

Accuracy Guarantee – Introduction



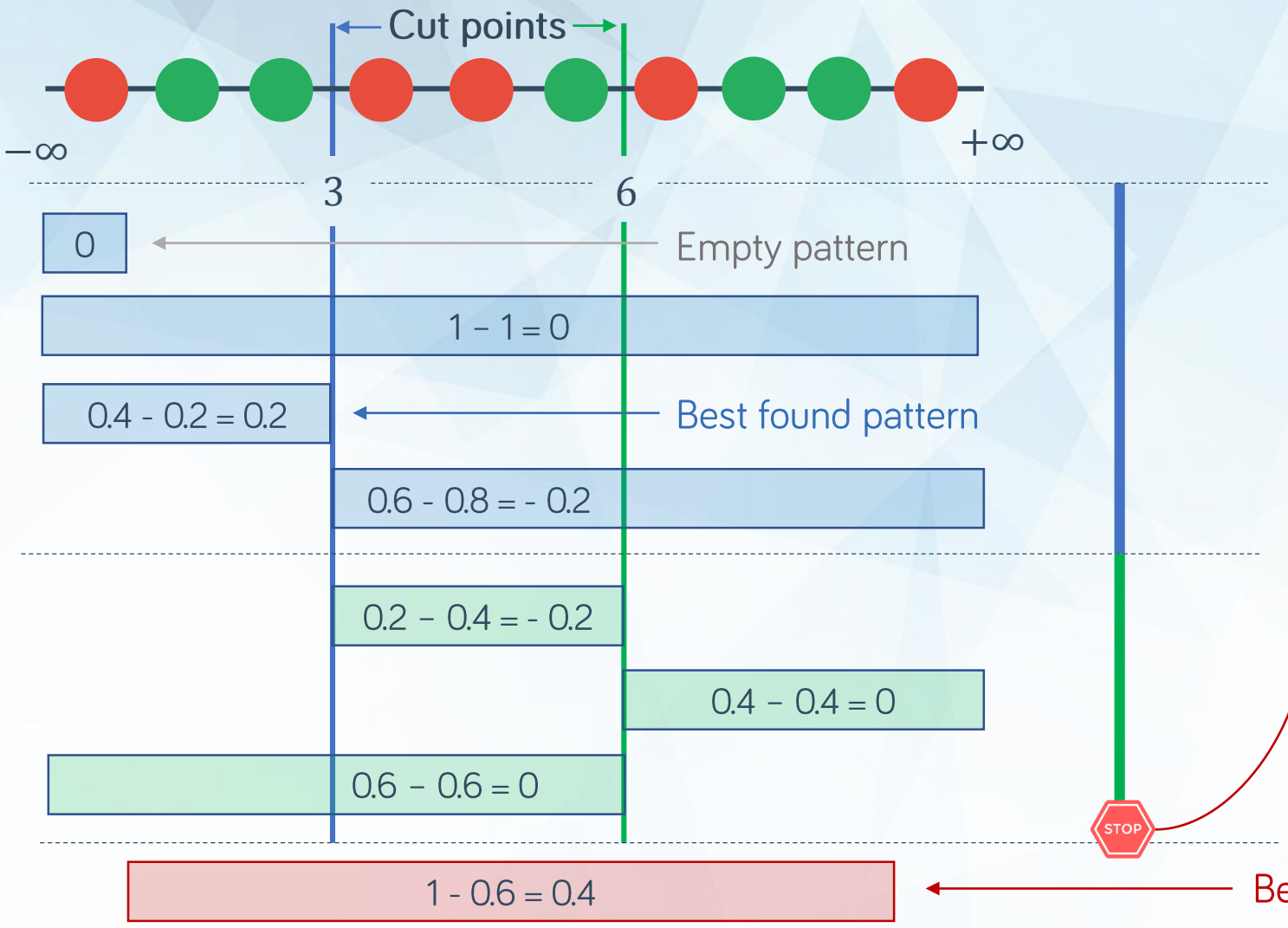
$$tpr(pattern) - fpr(pattern) = Informedness(pattern)$$

Accuracy Guarantee – Introduction



$$tpr(pattern) - fpr(pattern) = Informedness(pattern)$$

Accuracy Guarantee – Introduction



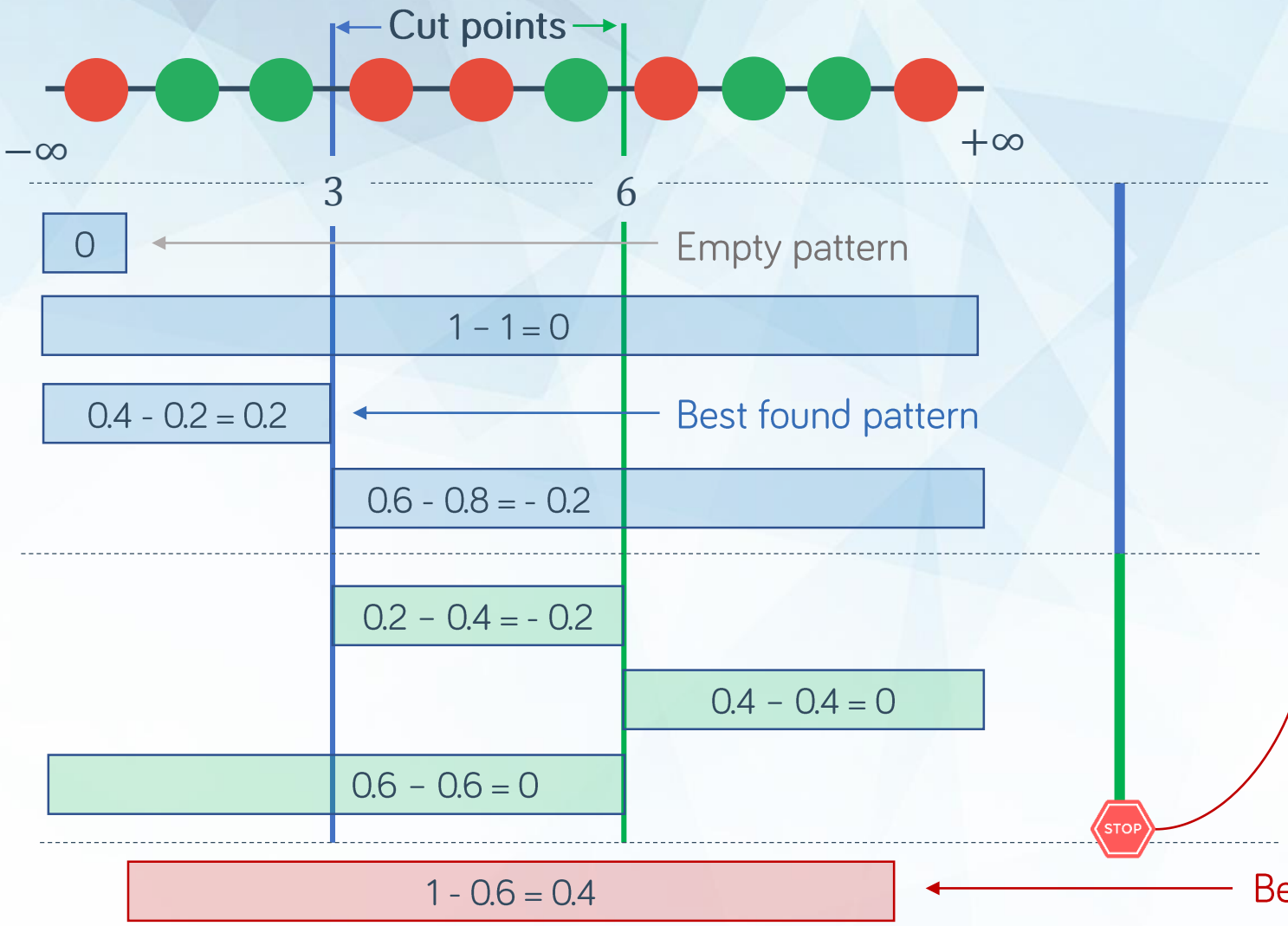
$$tpr(pattern) - fpr(pattern) = Informedness(pattern)$$

Refine&Mine gives a bound of

0.6 on Informedness

when interrupted.

Accuracy Guarantee – Introduction



$$tpr(pattern) - fpr(pattern) = Informedness(pattern)$$

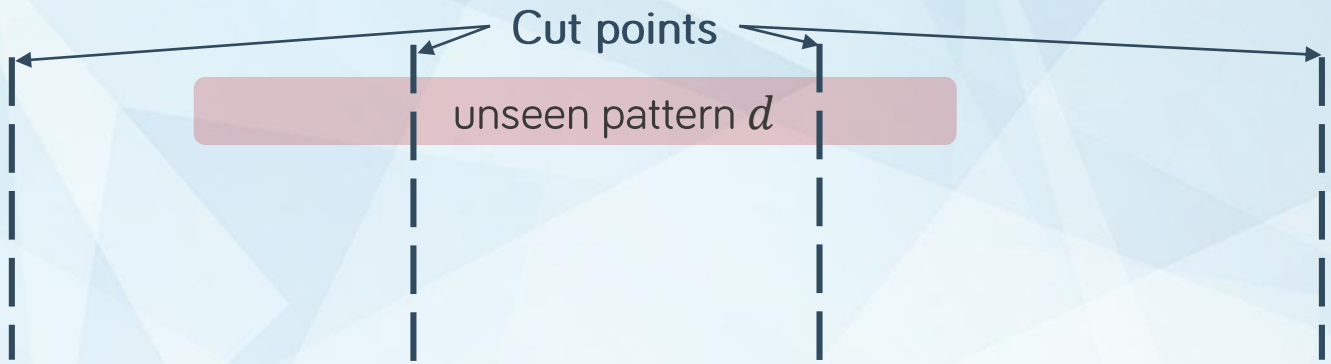
Refine&Mine gives a bound of

0.6 on Informedness

when interrupted.

? How?

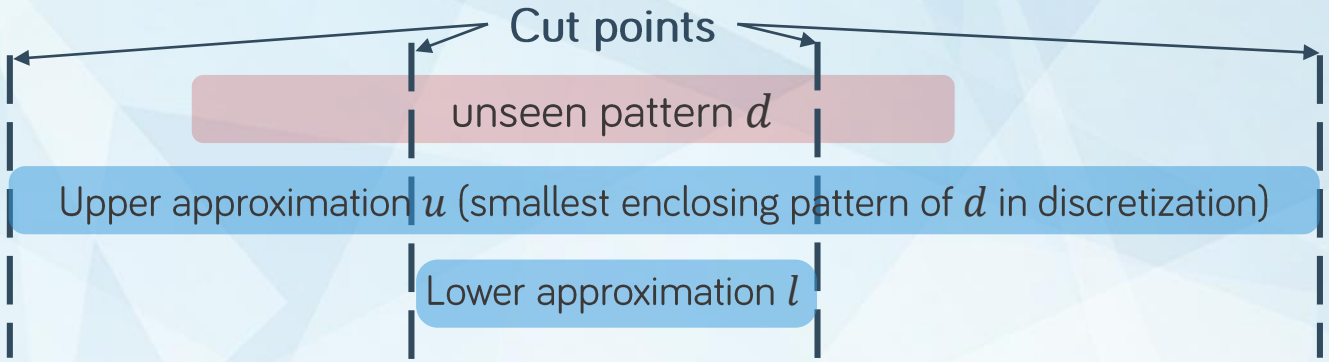
Accuracy Guarantee – Upper and lower approximations



Z. Pawlak. Rough sets. In International Journal of Parallel Programming 1982.
A. Tarski. A lattice-theoretical fixpoint theorem and its applications. In Pacific journal of Mathematics 1955.

Accuracy Guarantee – Upper and lower approximations

Any unseen interval pattern can be encapsulated between two found patterns.

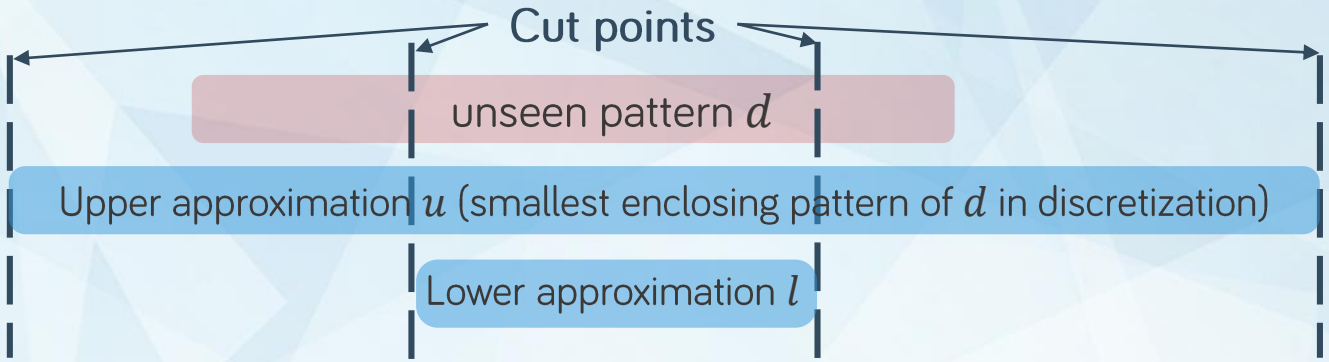


Z. Pawlak. Rough sets. In International Journal of Parallel Programming 1982.
A. Tarski. A lattice-theoretical fixpoint theorem and its applications. In Pacific journal of Mathematics 1955.

Accuracy Guarantee – Upper and lower approximations

I

Any unseen interval pattern can be encapsulated between two found patterns.



II

tpr and fpr measures are monotonous.

- $\text{tpr}(d) \leq \text{tpr}(u)$
- $\text{fpr}(d) \geq \text{fpr}(l)$

$\Rightarrow informedness(d) \leq \text{tpr}(u) - \text{fpr}(l)$

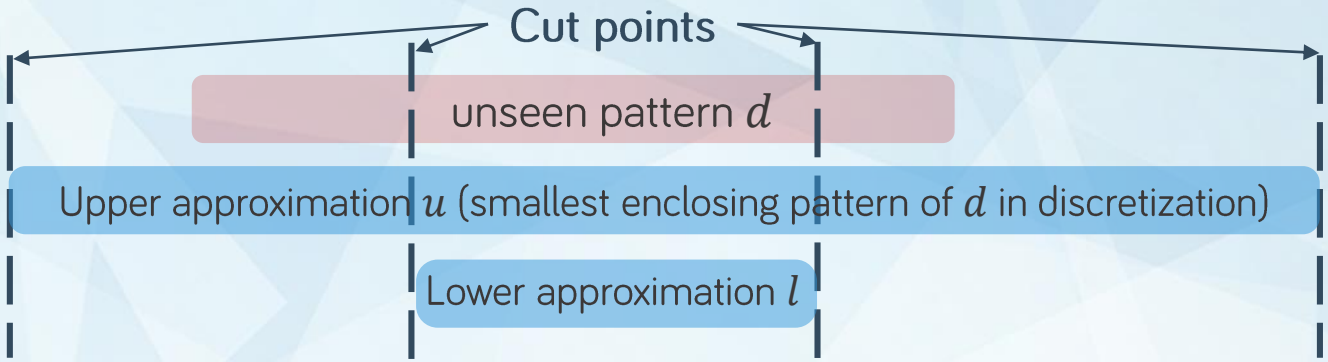
Z. Pawlak. Rough sets. In International Journal of Parallel Programming 1982.
A. Tarski. A lattice-theoretical fixpoint theorem and its applications. In Pacific journal of Mathematics 1955.

Accuracy Guarantee – Upper and lower approximations

I + II

Any unseen interval pattern can be encapsulated between two found patterns.

tpr and fpr measures are monotonous.



- $\text{tpr}(d) \leq \text{tpr}(u)$
- $\text{fpr}(d) \geq \text{fpr}(l)$

$\Rightarrow informedness(d) \leq \text{tpr}(u) - \text{fpr}(l)$

Now, we gave a bound on the quality of an unseen pattern using its two already found patterns.

Z. Pawlak. Rough sets. In International Journal of Parallel Programming 1982.
A. Tarski. A lattice-theoretical fixpoint theorem and its applications. In Pacific journal of Mathematics 1955.

Accuracy Guarantee – Core notion

Q. So, do we need to look for pair of patterns in the discretization search space to give bound the quality measure of all unseen patterns?

Accuracy Guarantee – Core notion

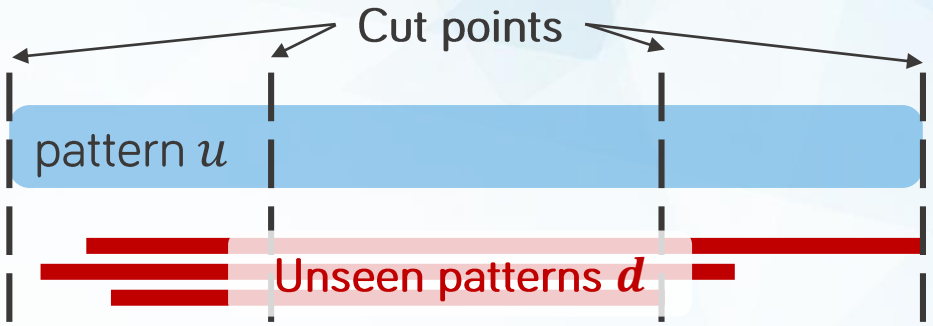
Q. So, do we need to look for pair of patterns in the discretization search space to give bound the quality measure of all unseen patterns?

A. No, we use the notion of core

Accuracy Guarantee – Core notion

Q. So, do we need to look for pair of patterns in the discretization search space to give bound the quality measure of all unseen patterns?

A. No, we use the notion of core

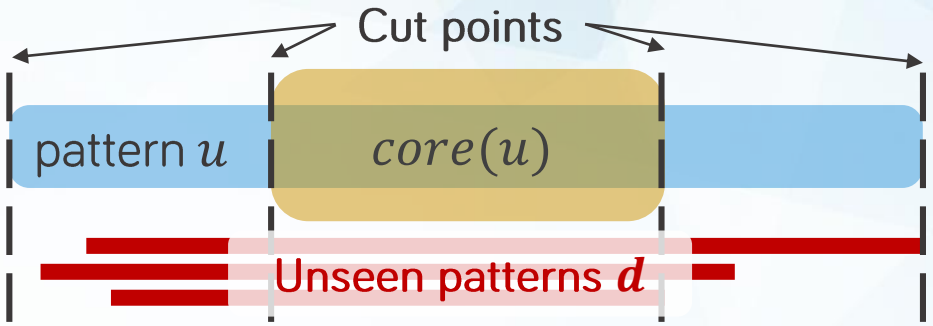


- Take all unseen patterns which upper approximation is u.

Accuracy Guarantee – Core notion

Q. So, do we need to look for pair of patterns in the discretization search space to give bound the quality measure of all unseen patterns?

A. No, we use the notion of core



- Take all unseen patterns which upper approximation is u.
- The **core** of a found pattern u the largest possible lower approximation of all these unseen patterns.

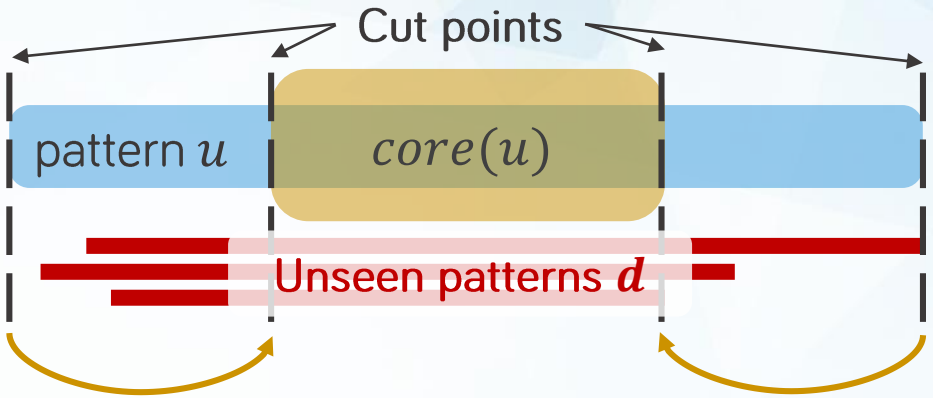
Upper approximation

Core(found pattern) ⊆ some unseen patterns ⊆ found pattern

Accuracy Guarantee – Core notion

Q. So, do we need to look for pair of patterns in the discretization search space to give bound the quality measure of all unseen patterns?

A. No, we use the notion of core



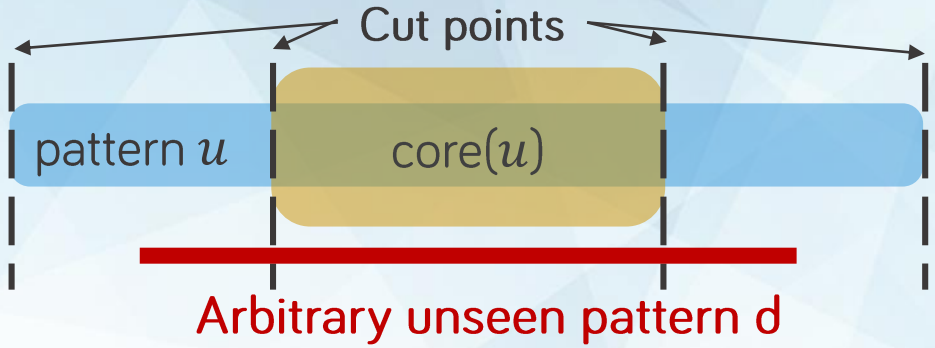
- Take all unseen patterns which upper approximation is u.
- The **core** of a found pattern u the largest possible lower approximation of all these unseen patterns.

Upper approximation

Core(found pattern) ⊆ some unseen patterns ⊆ found pattern

The core computation complexity is $O(d)$ with d is the number of dimensions.

Accuracy Guarantee – Computation

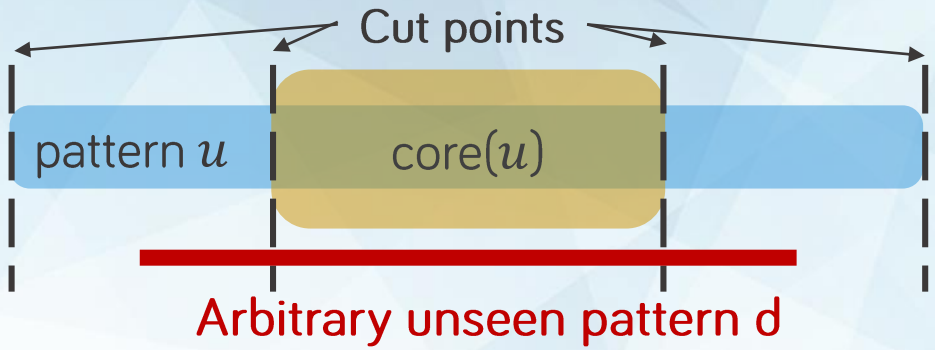


- Core(found pattern) ⊆ some unseen patterns ⊆ found pattern

Upper approximation

$$\text{Accuracy} = \sup_{d \in all} \phi(d) - \sup_{u \in found} \phi(u)$$

Accuracy Guarantee – Computation



- Core(found pattern) ⊆ some unseen patterns ⊆ found pattern

Upper approximation

$$\text{Accuracy} = \sup_{d \in all} \phi(d) - \sup_{u \in found} \phi(u)$$

$$\downarrow \leq$$

$$\text{Accuracy} \leq \sup_{u \in found} \phi\left(tpr(u), fpr\left(core(u)\right)\right) - \sup_{u \in found} \phi(u)$$

Specificity Guarantee

$$\text{Specificity} = \overrightarrow{distance}(\text{All found patterns}, \text{All pattern in the dataset})$$

Specificity Guarantee

$$\text{Specificity} = \overrightarrow{distance}(\text{All found patterns}, \text{All pattern in the dataset})$$

$$\text{Specificity} \leq \text{Specificity bound}$$

Same aforementioned notions are used to define the bound on specificity

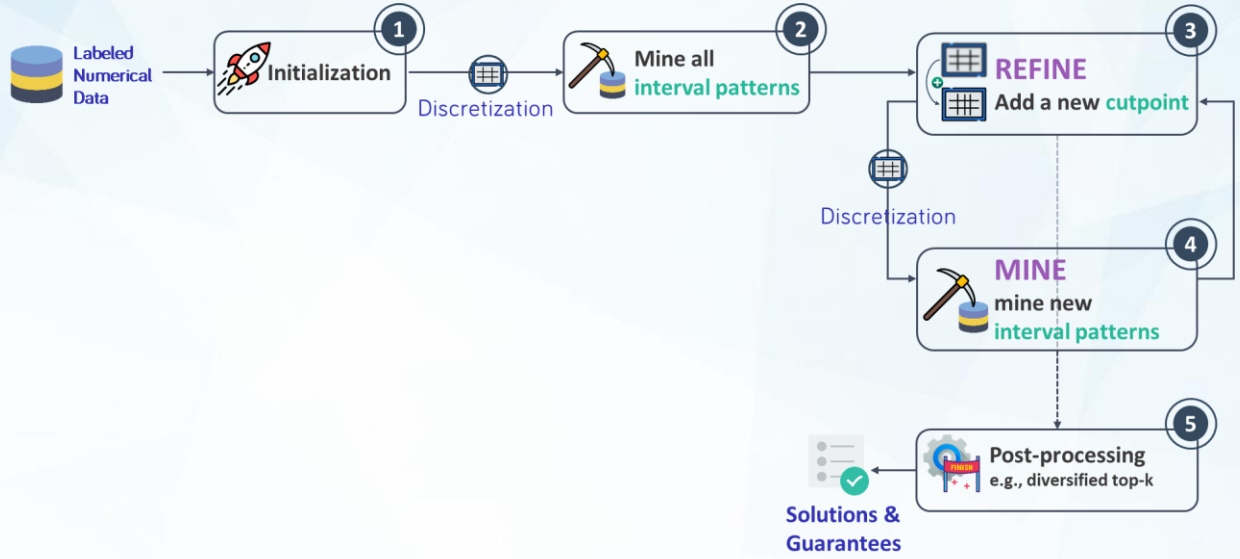
Refine&Mine computes this bound using only already found patterns.

OUTLINE

- Overview and Contributions
- Refine&Mine in a nutshell
- Guarantees computation
- Building blocks of Refine&Mine
- Wrap-up and concluding remarks

Refine&Mine

Step-by-step



REFINE&MINE: Step-by-step

Labeled
Numerical
Data

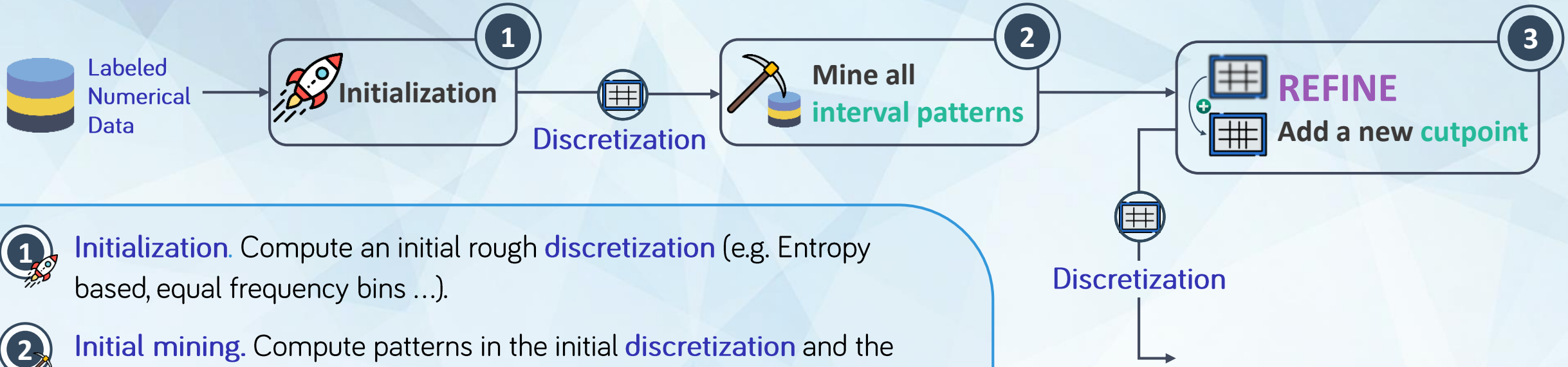
REFINE&MINE: Step-by-step

1 **Initialization.** Compute an initial rough **discretization** (e.g. Entropy based, equal frequency bins ...).

REFINE&MINE: Step-by-step

1. Initialization. Compute an initial rough discretization (e.g. Entropy based, equal frequency bins ...).
2. Initial mining. Compute patterns in the initial discretization and the first guarantees (e.g. use optimizations like closure operator) .

REFINE&MINE: Step-by-step

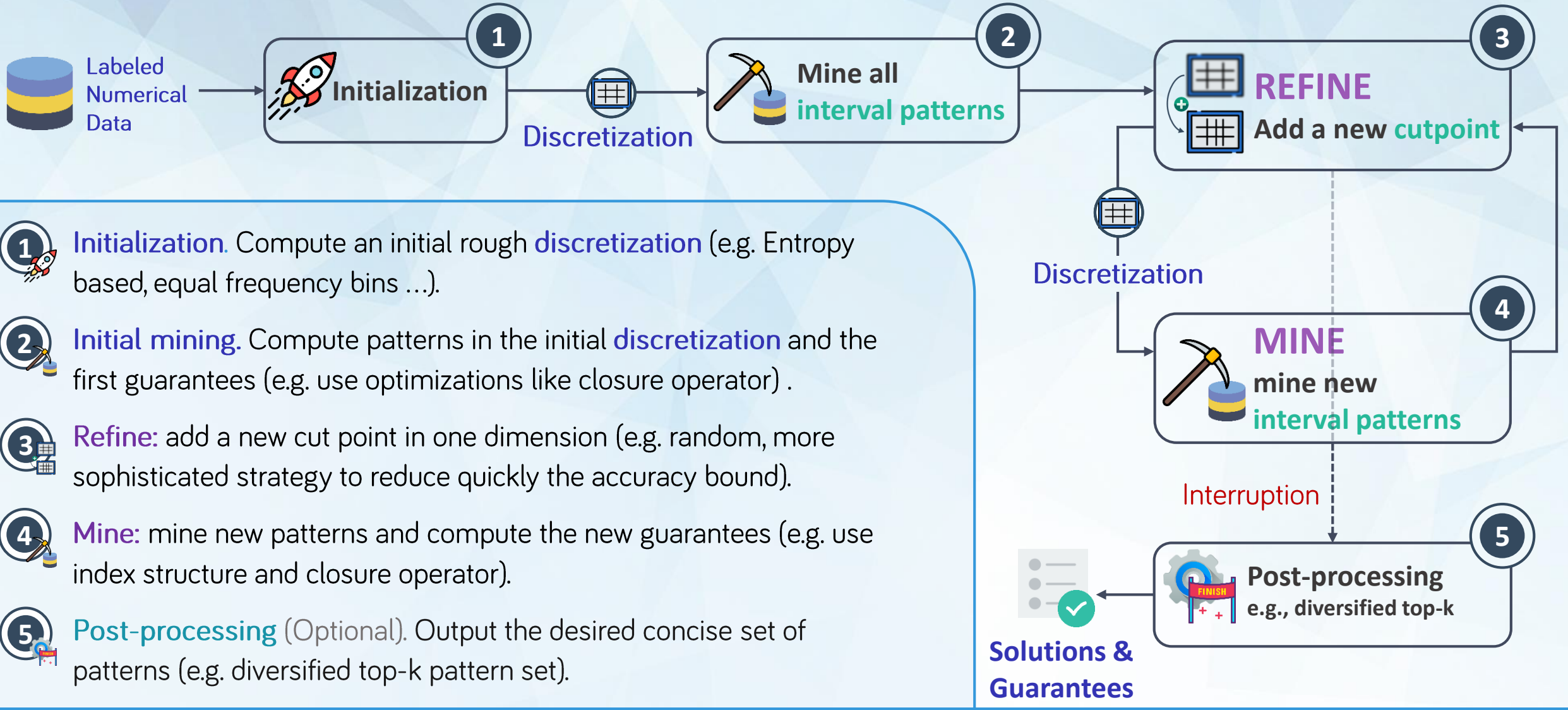


1. **Initialization.** Compute an initial rough **discretization** (e.g. Entropy based, equal frequency bins ...).
2. **Initial mining.** Compute patterns in the initial **discretization** and the first guarantees (e.g. use optimizations like closure operator) .
3. **Refine:** add a new cut point in one dimension (e.g. random, more sophisticated strategy to reduce quickly the accuracy bound).

REFINE&MINE: Step-by-step

1. **Initialization.** Compute an initial rough **discretization** (e.g. Entropy based, equal frequency bins ...).
2. **Initial mining.** Compute patterns in the initial **discretization** and the first guarantees (e.g. use optimizations like closure operator) .
3. **Refine:** add a new cut point in one dimension (e.g. random, more sophisticated strategy to reduce quickly the accuracy bound).
4. **Mine:** mine new patterns and compute the new guarantees (e.g. use index structure and closure operator).

REFINE&MINE: Step-by-step

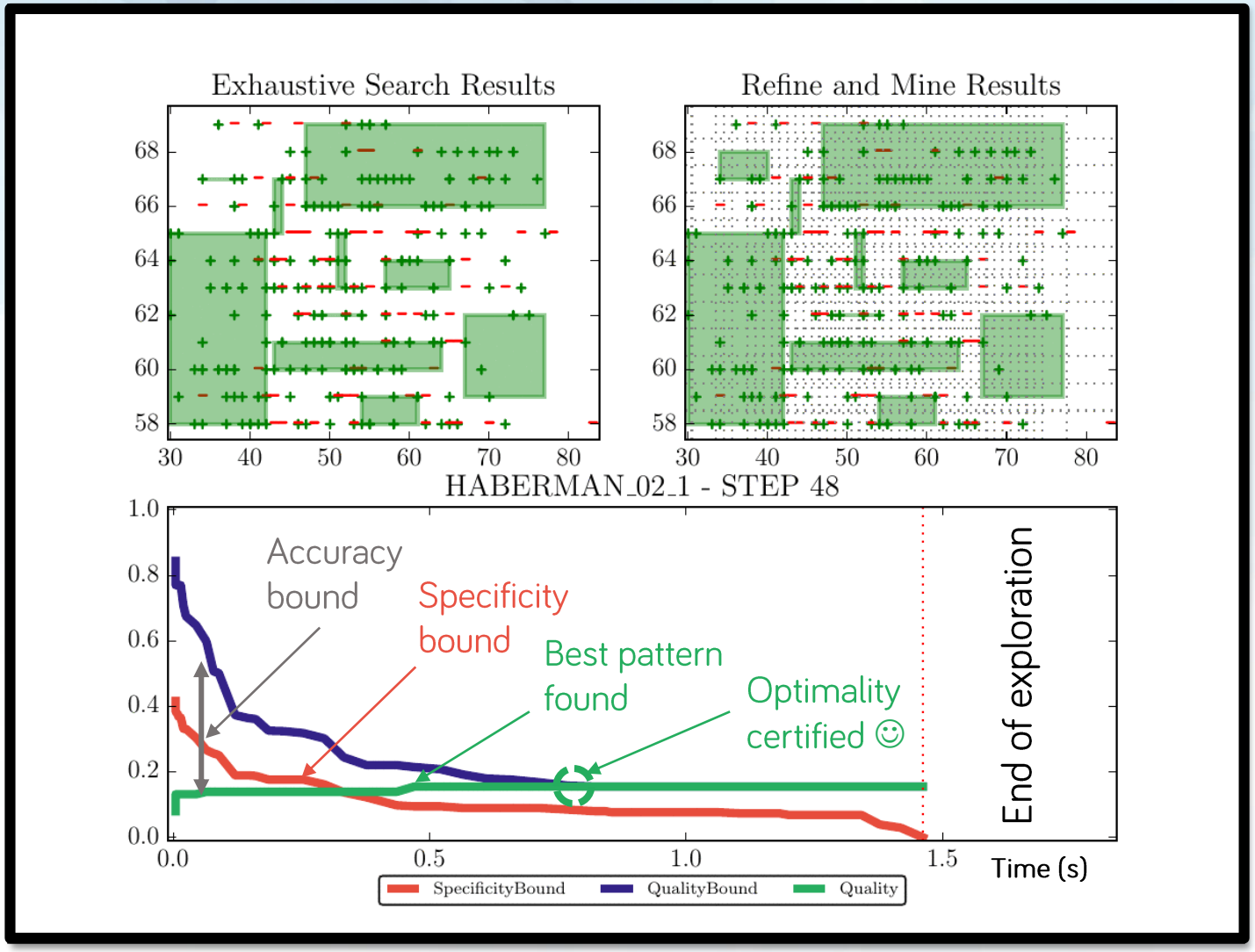


1. **Initialization**. Compute an initial rough **discretization** (e.g. Entropy based, equal frequency bins ...).
2. **Initial mining.** Compute patterns in the initial **discretization** and the first guarantees (e.g. use optimizations like closure operator) .
3. **Refine:** add a new cut point in one dimension (e.g. random, more sophisticated strategy to reduce quickly the accuracy bound).
4. **Mine:** mine new patterns and compute the new guarantees (e.g. use index structure and closure operator).
5. **Post-processing** (Optional). Output the desired concise set of patterns (e.g. diversified top-k pattern set).

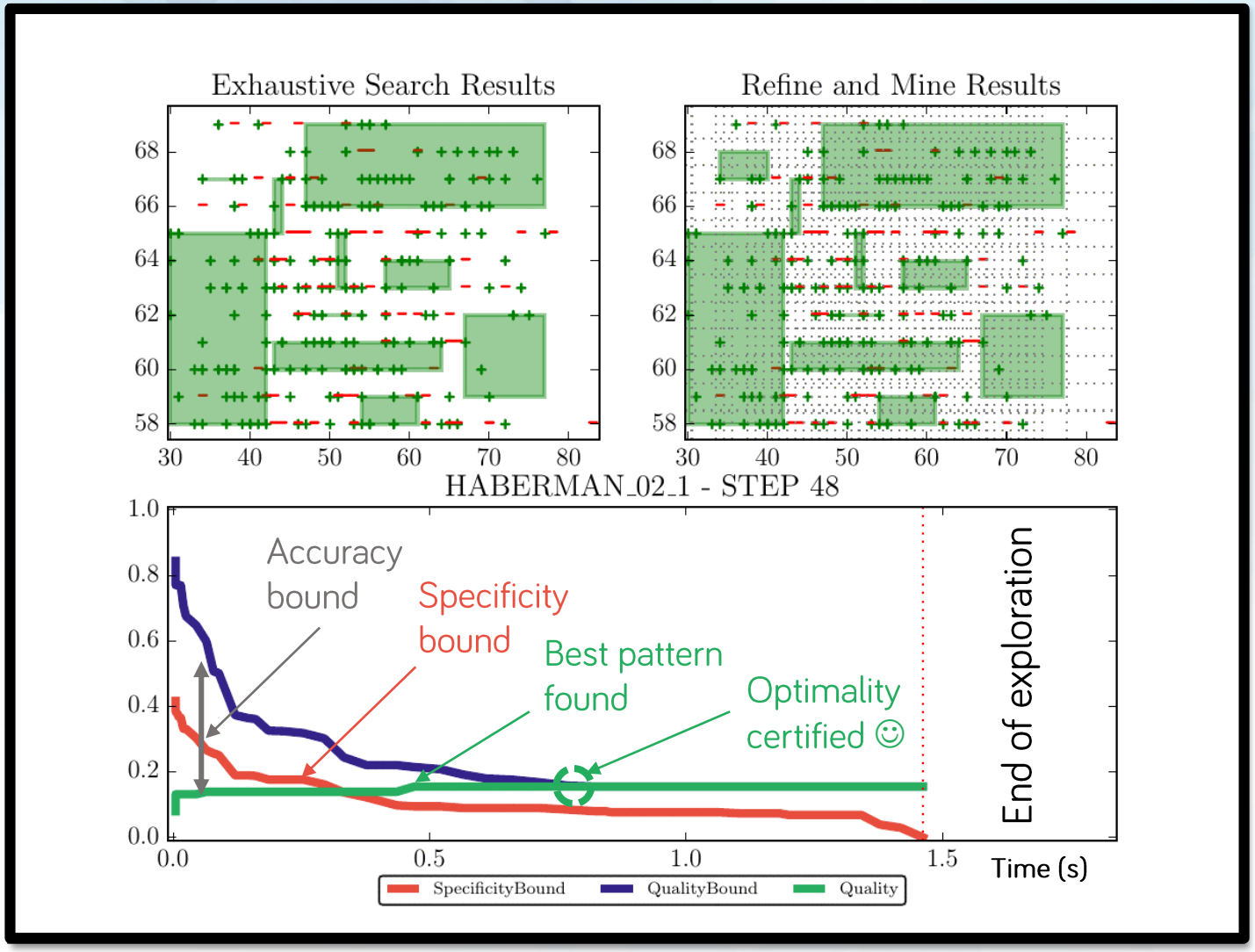
Refine&Mine

Wrap-up

REFINE&MINE: Wrap-up



REFINE&MINE: Wrap-up

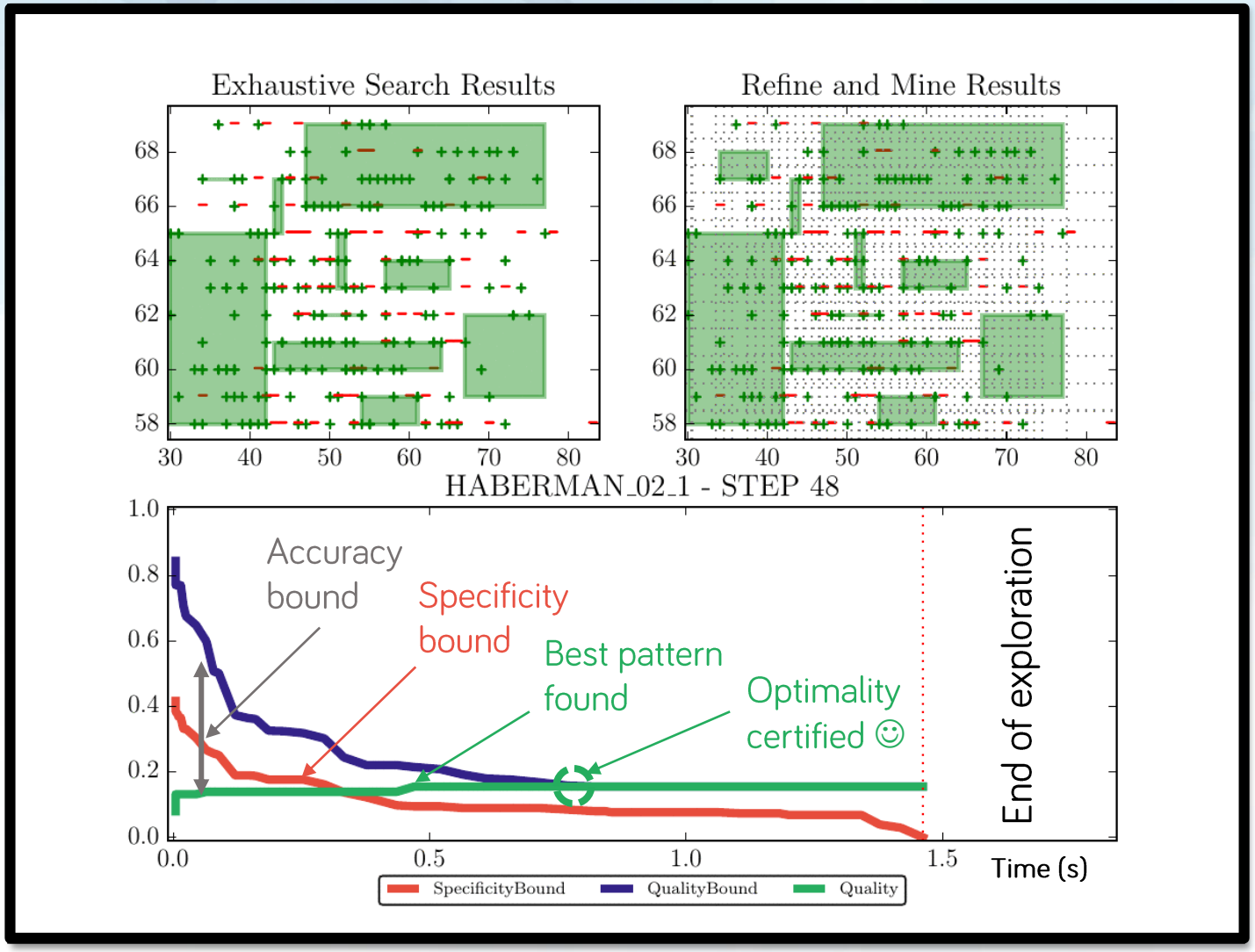


Empirical Evaluation

Experiments were performed over

25 UCI Datasets

REFINE&MINE: Wrap-up



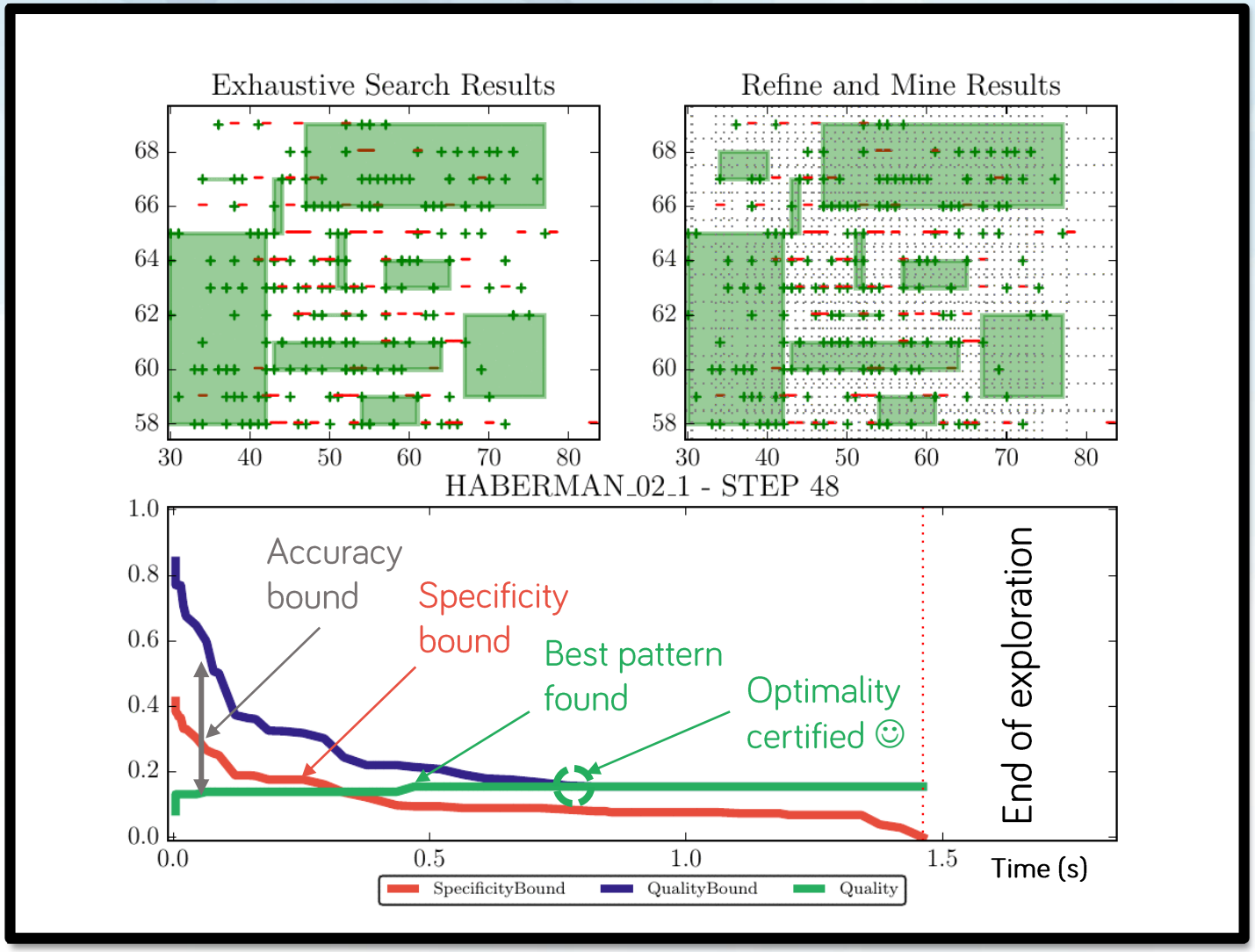
Empirical Evaluation

Experiments were performed over

25 UCI Datasets

- Refine&Mine achieves to find a 80% ground truth in less than 20% of the time required for the exhaustive search algorithm to finish

REFINE&MINE: Wrap-up



Empirical Evaluation

Experiments were performed over

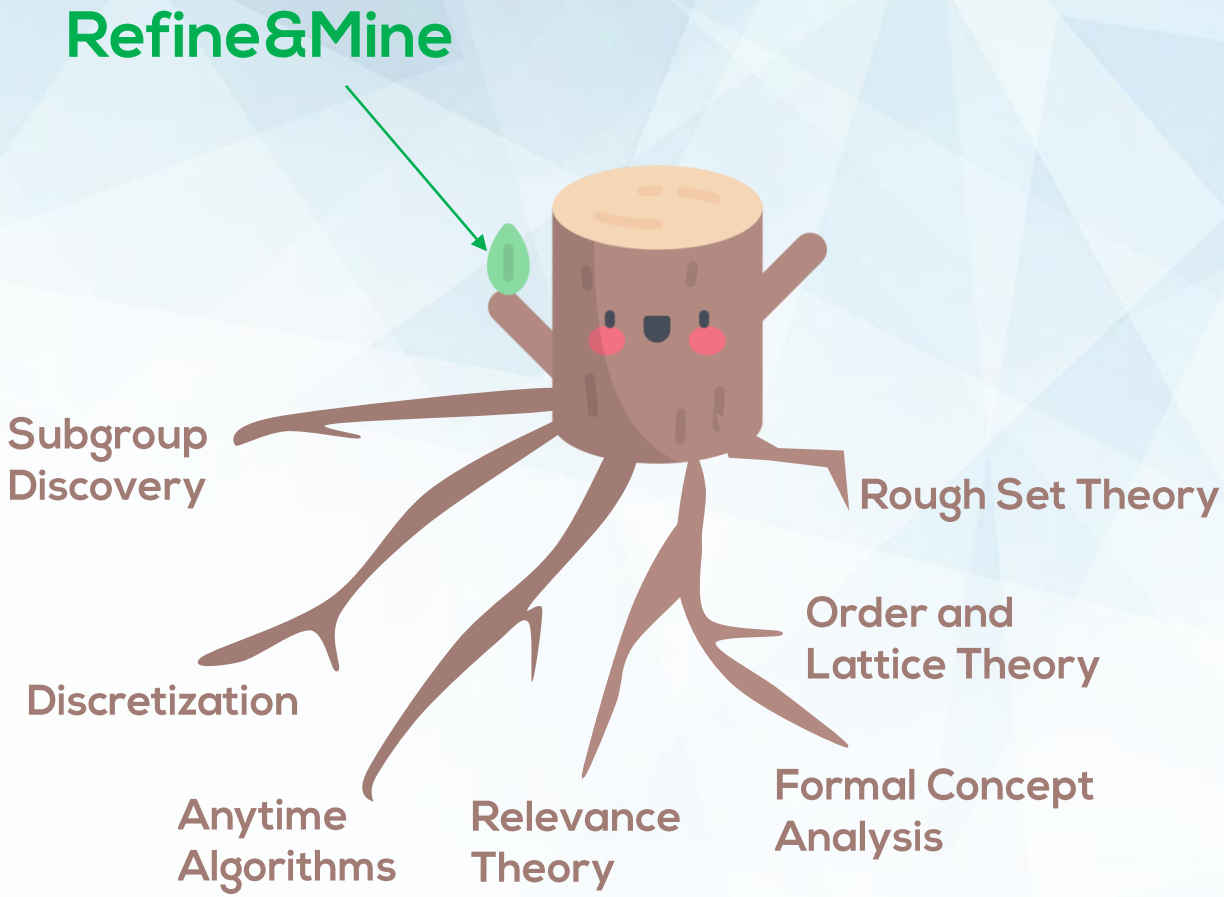
25 UCI Datasets

- Refine&Mine achieves to find a 80% ground truth in less than 20% of the time required for the exhaustive search algorithm to finish
- Refine&Mine achieved better results both in finding the best pattern and a high diversified results set compared to MCTS4DM [*]

[*] *G. Bosc, J.F. Boulicaut, C. Raïssi, M. Kaytoue. Anytime discovery of a diverse set of patterns with MCTS. In DMKD 2018.*

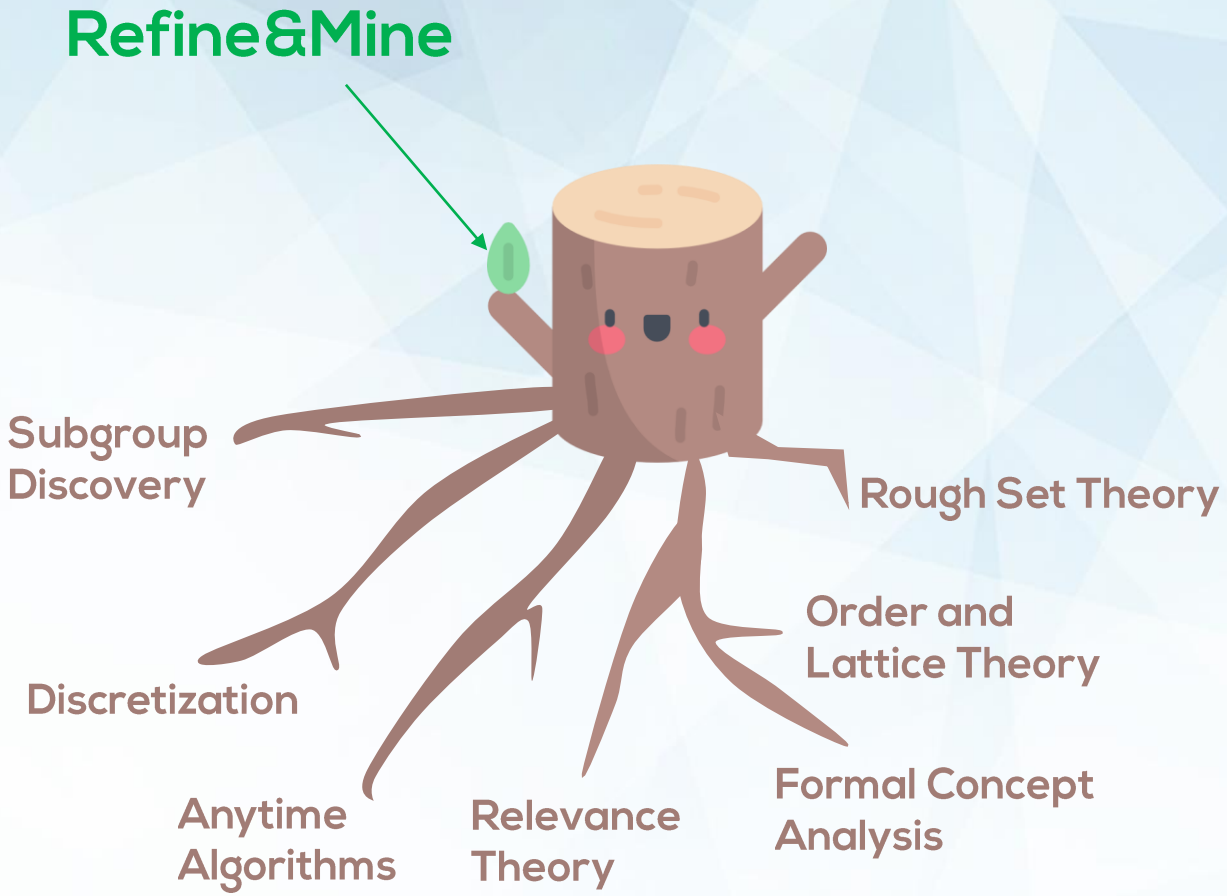
CONCLUSION & PERSPECTIVES

Conclusion



Proposed the new algorithm **Refine&Mine**

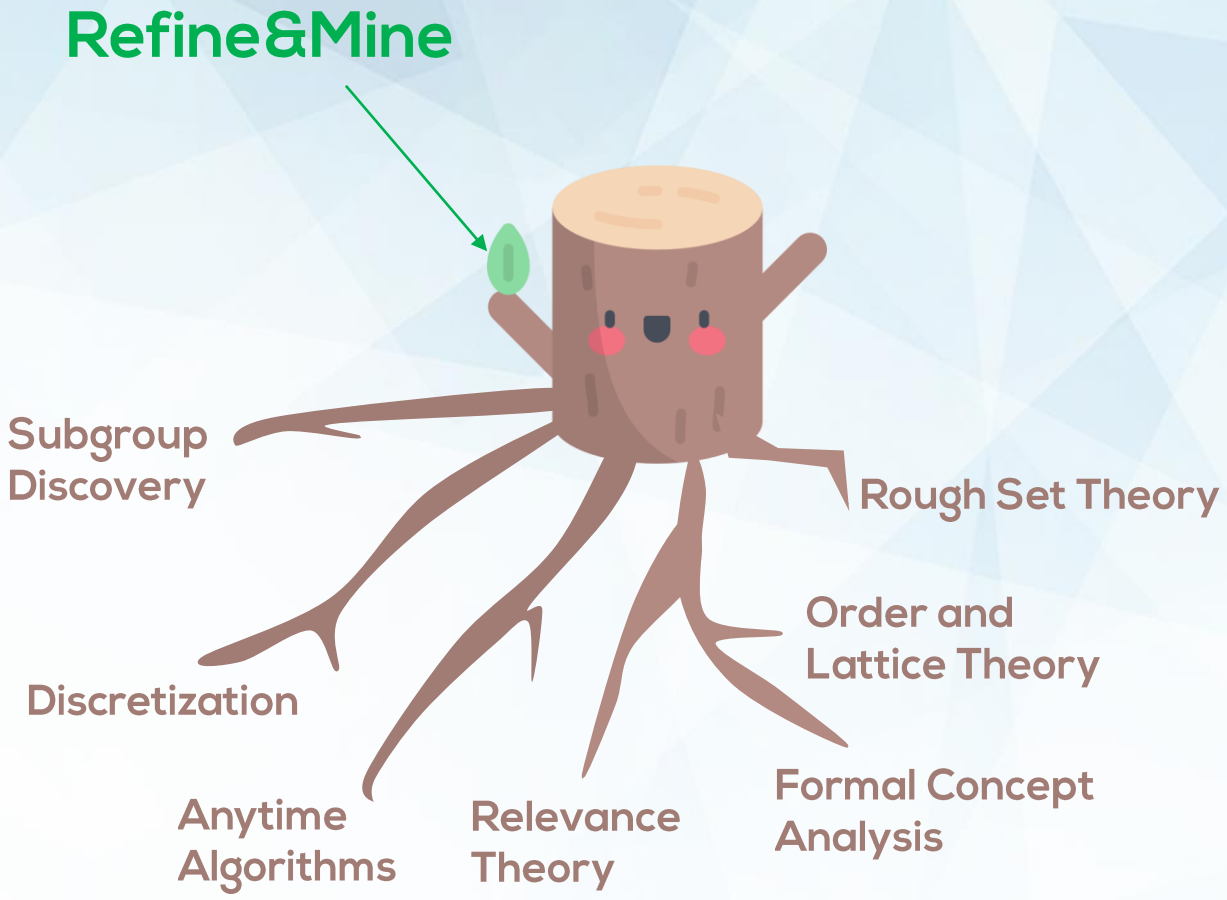
Conclusion



Proposed the new algorithm **Refine&Mine**

Anytime Algorithm

Conclusion

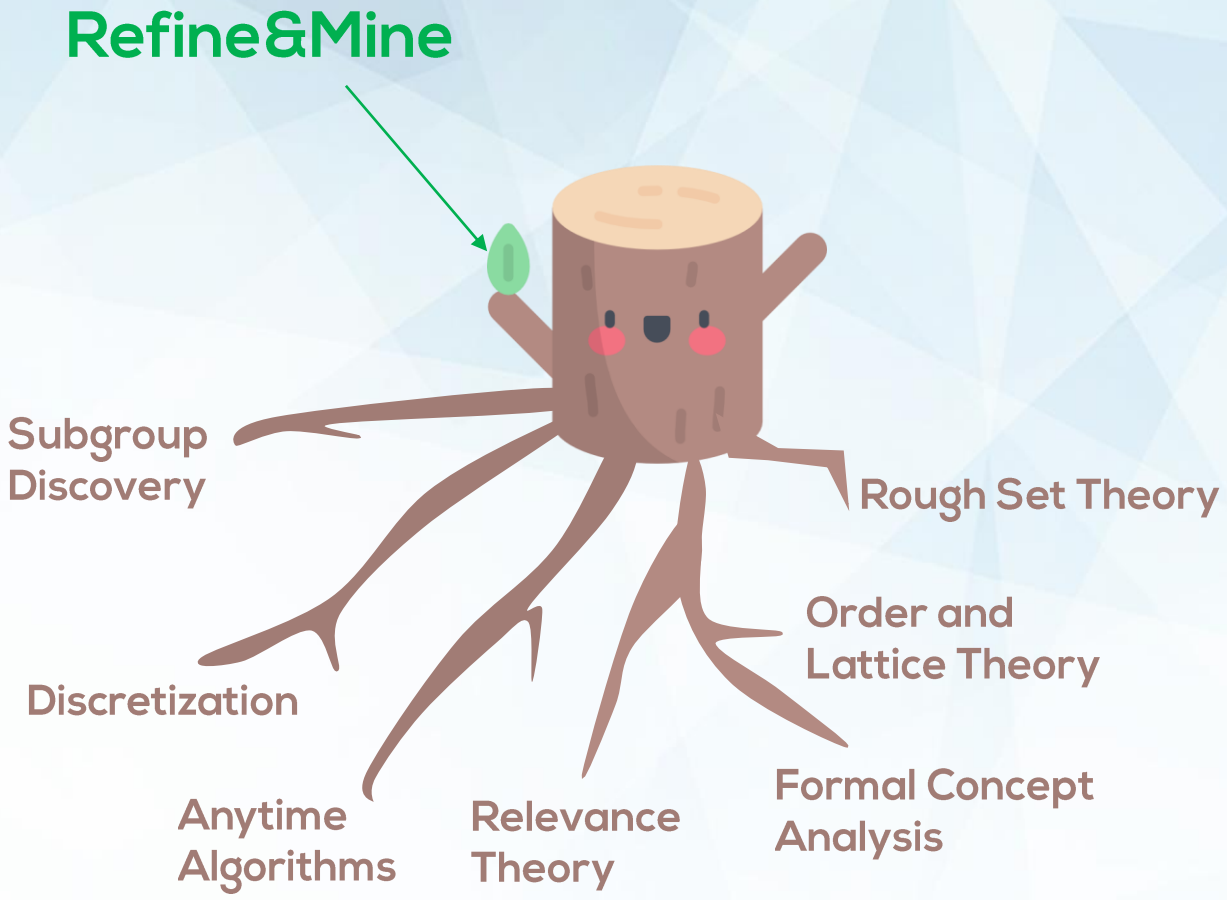


Proposed the new algorithm **Refine&Mine**

Anytime Algorithm

Guarantees on the found patterns anytime

Conclusion



Proposed the new algorithm **Refine&Mine**

Anytime Algorithm

Guarantees on the found patterns anytime

It performs an exhaustive search If given enough time

Future works

Handle High dimensional data.

Limit. We need **2 cut points in each dimension** to compute the first guarantees:

d attributes $\Leftrightarrow 6^d$ *patterns*

Future works

Handle High dimensional data.

Limit. We need **2 cut points in each dimension** to compute the first guarantees:
d attributes $\Leftrightarrow 6^d$ *patterns*

Handle datasets with hybrid attributes.

e.g. binary, nominal, numerical, sequences, graphs.

Future works

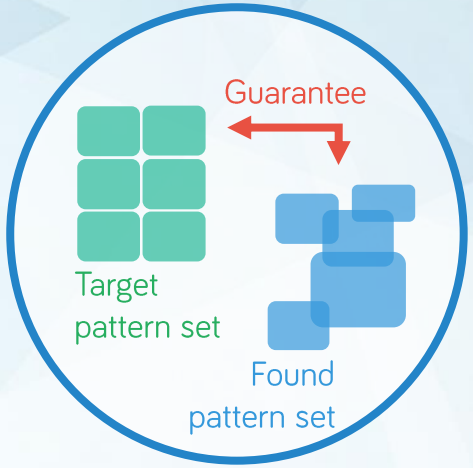
Handle High dimensional data.

Limit. We need **2 cut points in each dimension** to compute the first guarantees:

d attributes $\Leftrightarrow$ 6^d *patterns*

Handle datasets with hybrid attributes.

e.g. binary, nominal, numerical, sequences, graphs.



Provide guarantees on pattern sets rather than all patterns.

THANKS

FOR YOUR TIME

QUESTIONS

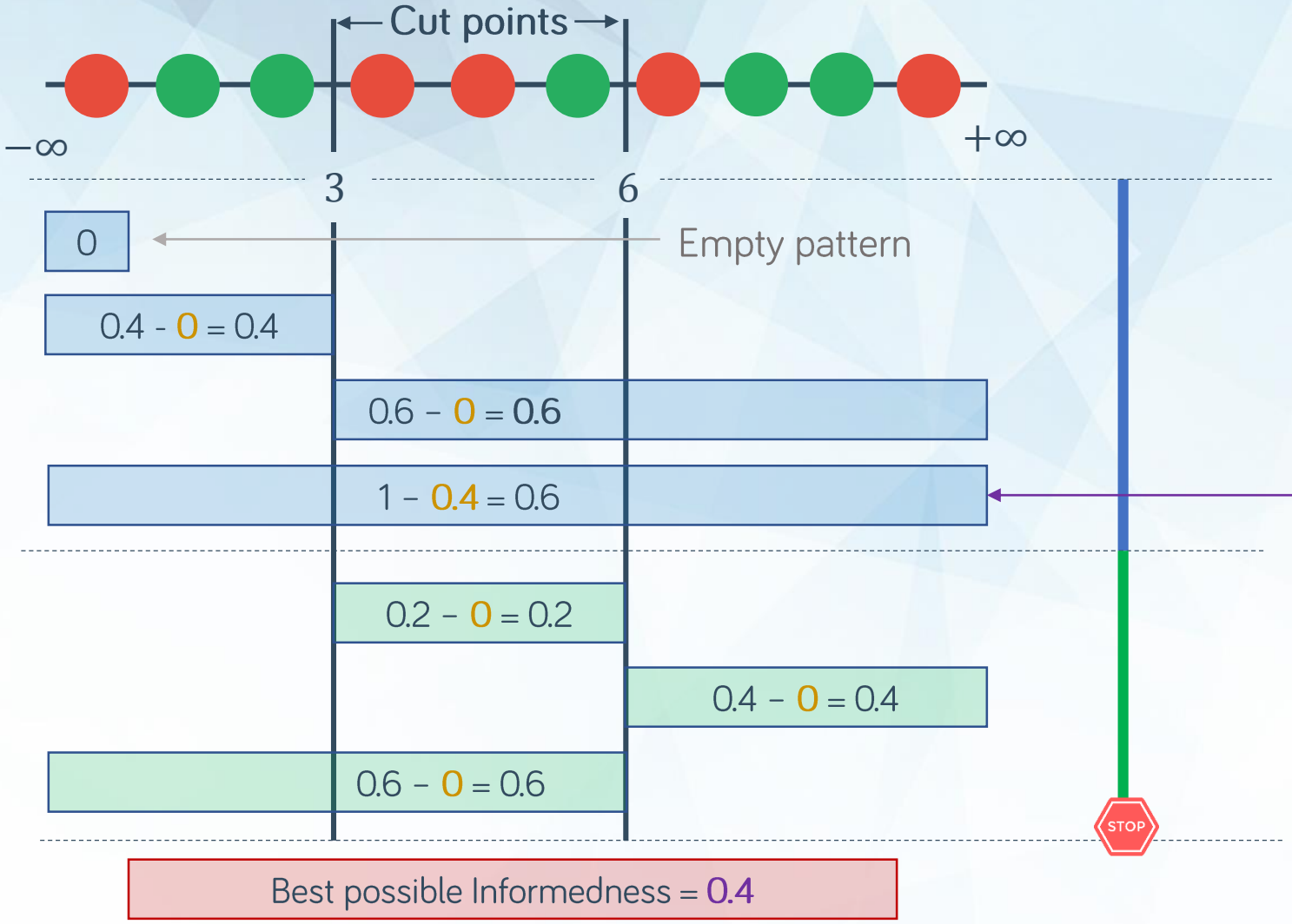
Contact : adnene.belfodil@insa-lyon.fr & aimene.belfodil@insa-lyon.fr

Materials: https://github.com/Adnene93/RefineAndMine

Looking forward to meeting up with you during the poster session :-)

ADDITIONAL DETAILS

Accuracy Guarantee – Example



$$tpr(pattern) - fpr(core(pattern)) = informedness\ bound(pattern)$$

- Best found Informedness = 0.2
- Best unseen Informedness = 0.4
- Highest Informedness bound = 0.6

- Accuracy = **0.4** − 0.2 = 0.2

- Accuracy bound = 0.6 − 0.2 = 0.4

First guarantee

Specificity Guarantee

$$\text{Specificity} = \overrightarrow{distance}(\text{All found patterns}, \text{All pattern in the dataset})$$

$$\text{Specificity} \leq \text{Specificity bound}$$

Refine&Mine computes this bound using only already found patterns.

Specificity Guarantee – Directed Distance

$$distance(c, d) = \frac{|c \,\Delta\, d|}{n} \in [0,1]$$

Symmetric difference

Pattern extents

Dataset size

The set of found patterns is compared to the set of ground truth patterns in this **direction** as follow:

- Each possible pattern is compared to its nearest found pattern.
- The top nearest distance is then the specificity

$$\overrightarrow{distance}(Found, All) = \sup_{d \in all} \inf_{c \in found} distance(c, d)$$

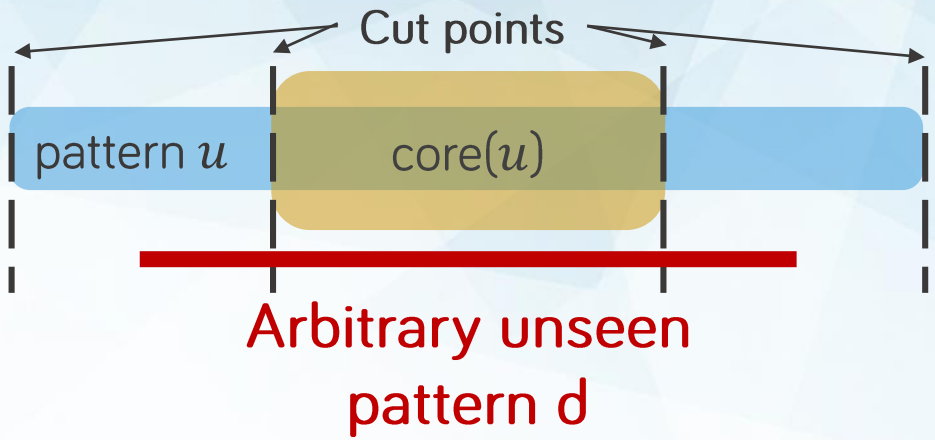
Specificity = Directed Hausdorff distance

Distance to the nearest found pattern

*The lower the **specificity** , the better found patterns are diversified

$Specificity = 0 \iff$ Found Patterns $\supseteq$ All Patterns.

Specificity Guarantee – Computation



$$\text{Specificity} = \sup_{d \in all} \inf_{c \in found} |c \,\Delta\, d|/n$$

- Core(found pattern) $\subseteq$ some unseen patterns $\subseteq$ found pattern

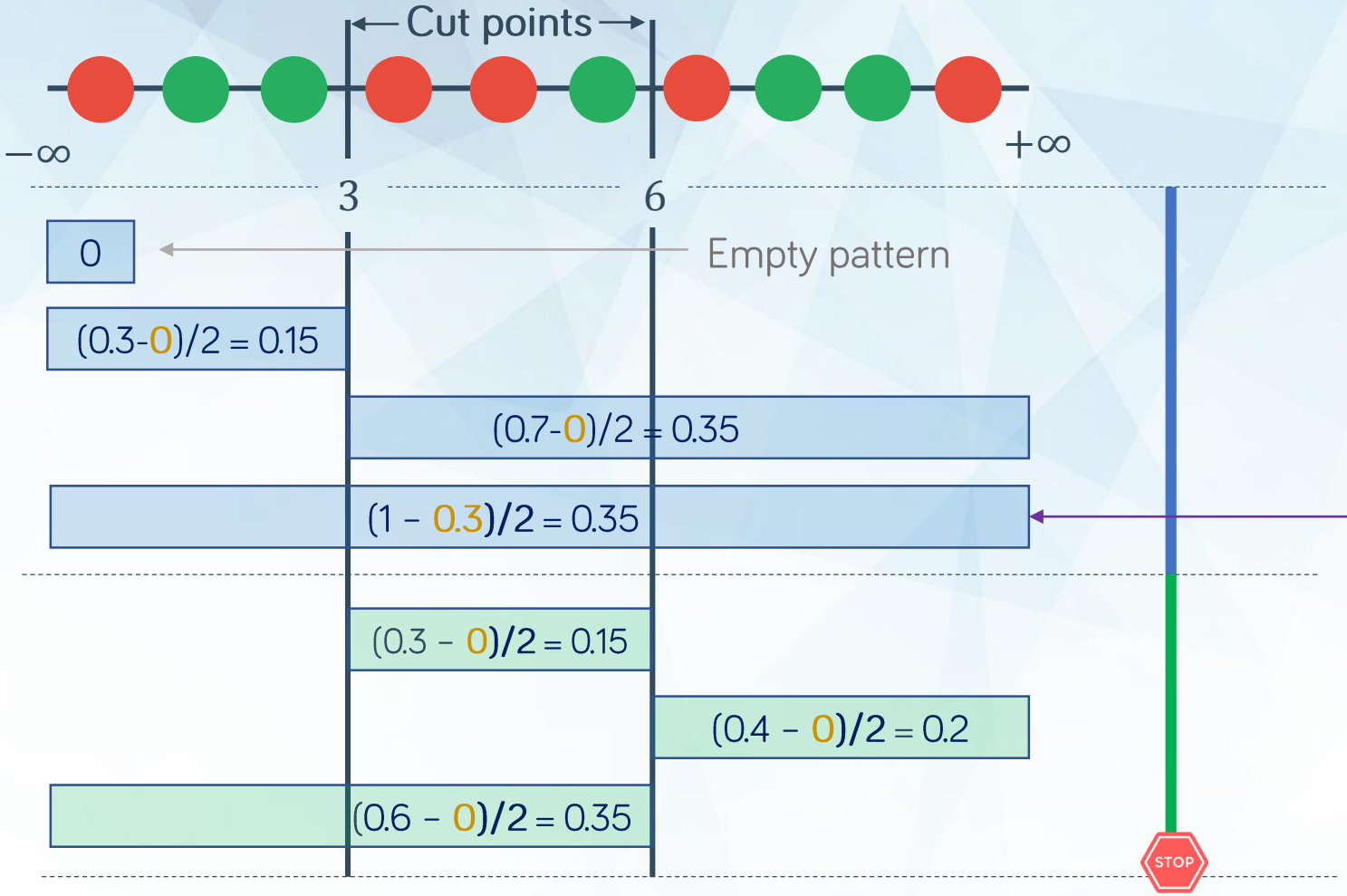
Upper approximation

- $\inf_{d \in all} \text{distance}(c,d) \leq \text{distance}(u,d) = \frac{|u|-|d|}{n}$
- $\inf_{d \in all} \text{distance}(c,d) \leq \text{distance}(core(u),d) = \frac{|d|-|core(u)|}{n}$

$$\text{Specificity} \leq \sup_{u \in found} \frac{|u| - |core(u)|}{2 \cdot n}$$

*This is a simplified formula.

Specificity Guarantee – Example



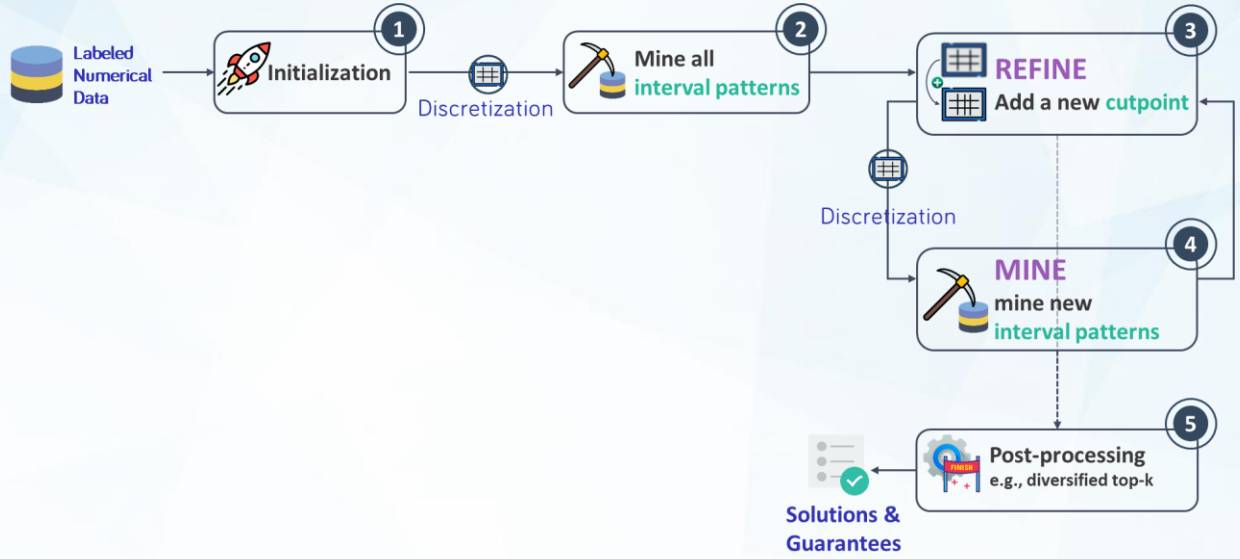
$$(|pattern| - |core(pattern)|)/2 \cdot 10 = \textbf{\textit{nearest distance bound}}(pattern)$$

- Specificity bound = 0.35

Second guarantee

Refine&Mine

Step-by-step



REFINE&MINE: Step-by-step (1)

1 Initialization

Compute an initial discretization [10,11]. e.g. Entropy based discretization, frequency based discretization...

[10] U.M. Fayyad, K.B. Irani. Multi-Interval discretization of continuous-valued attributes for classification learnings. In IJCAI 1993.
[11] Y. Yang, G.I. Webb, X., Wu: Descretization methods. In DMKD 2010.

REFINE&MINE: Step-by-step (2)

2 Mine

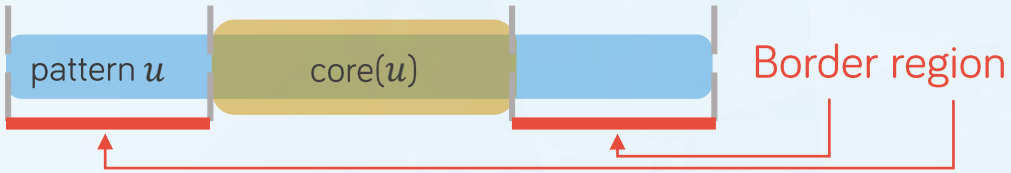
- Compute all interval patterns using MinIntChange algorithm [6] considering only the input-discretization selected points
- Compute the first guarantees.

[6] M. Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli. Revisiting Numerical Pattern Mining with Formal Concept Analysis. In IJCAI 2011.

REFINE&MINE: Step-by-step (3)

3 Refine

- Add one cutpoint in one dimension generating a new discretization, this can be done by:
  - Adding a random cut-point on a random dimension.
  - Adding a cut-point belonging to the border region of the pattern yielding the highest accuracy bound.



REFINE&MINE: Step-by-step (4)

4 Mine

- Mine new interval patterns using [6]

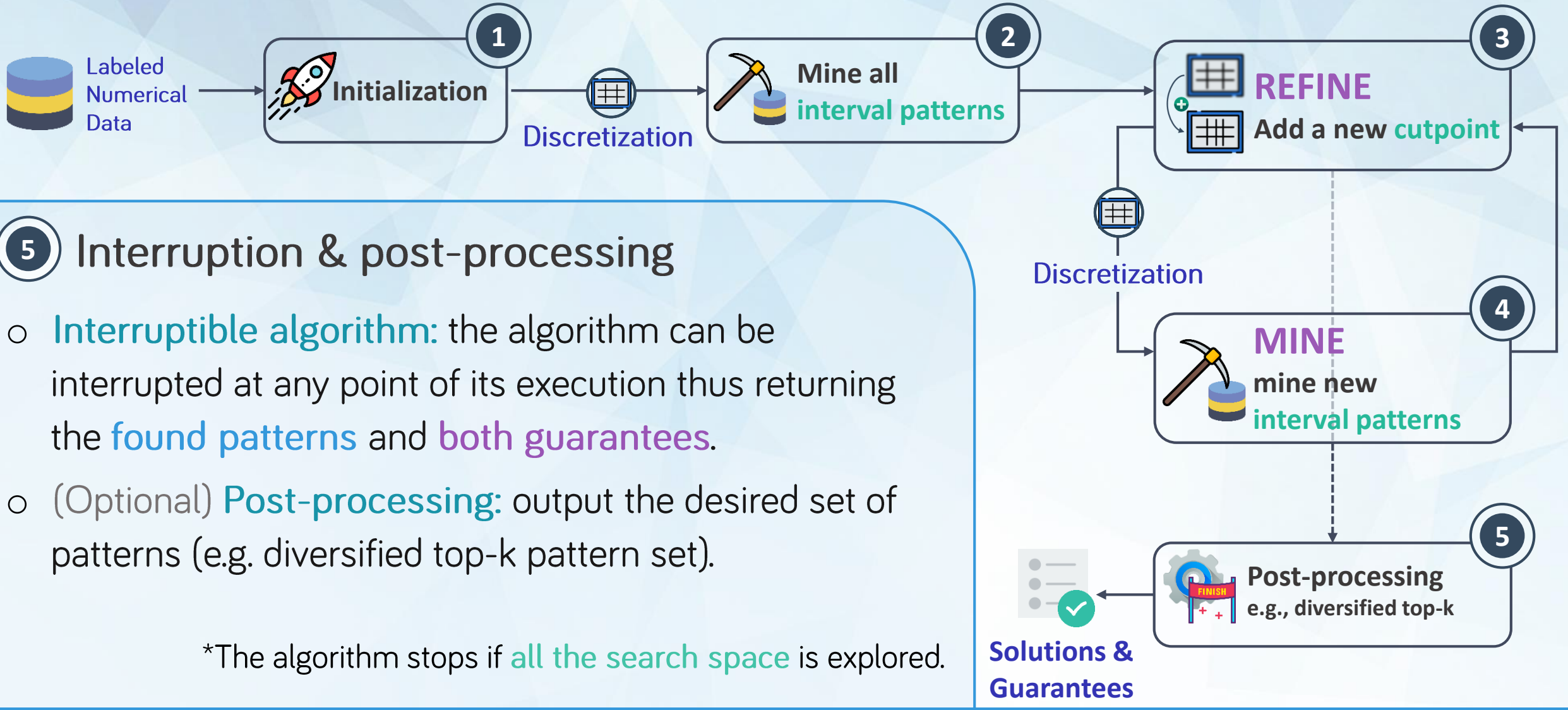
  by only visiting those whose left or right bound on the split dimension is the new selected cut-point.

- Update both guarantees

  An index structure is used to update quickly the guarantees

[6] M. Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli. Revisiting Numerical Pattern Mining with Formal Concept Analysis. In IJCAI 2011.

REFINE&MINE: Step-by-step (5)



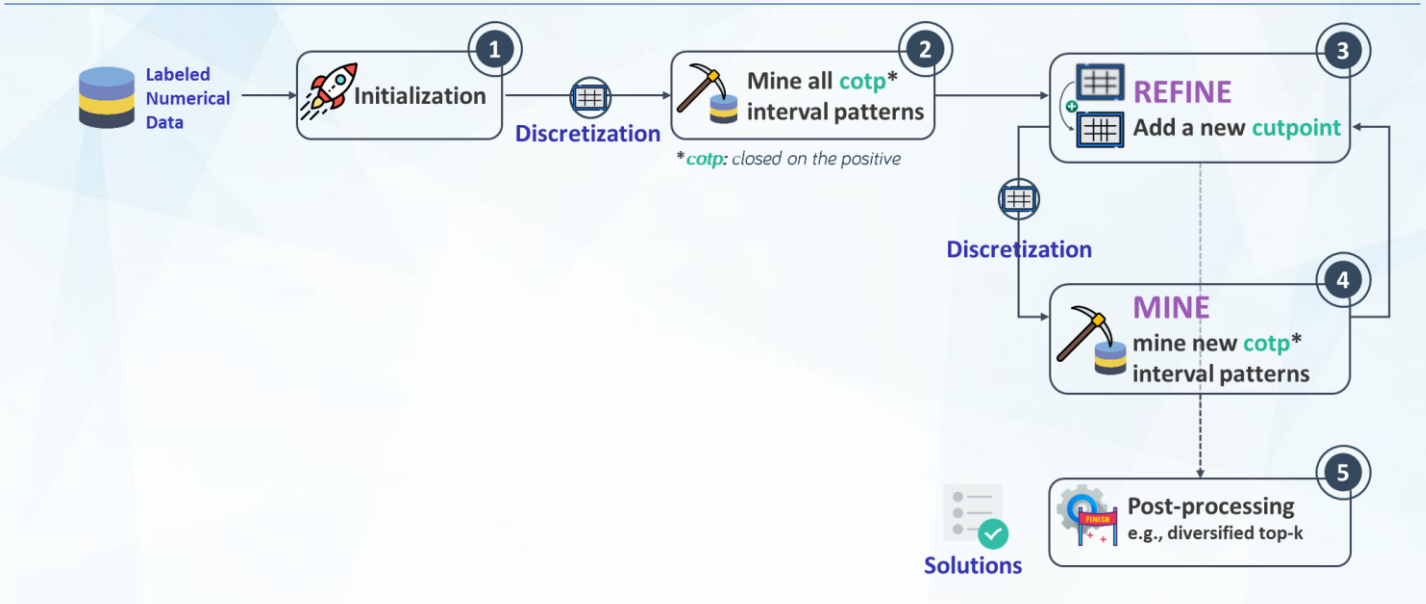
5 Interruption & post-processing

- Interruptible algorithm: the algorithm can be interrupted at any point of its execution thus returning the found patterns and both guarantees.
- (Optional) Post-processing: output the desired set of patterns (e.g. diversified top-k pattern set).

*The algorithm stops if all the search space is explored.

Refine&Mine

Step-by-step



REFINE&MINE: Step-by-step (1)

1 Initialization

- Compute relevant cuts [10] (cuts separating two classes).

- Compute an initial discretization [11].

[10] U.M. Fayyad, K.B. Irani. Multi-Interval discretization of continuous-valued attributes for classification learnings. In IJCAI 1993.
[11] Y. Yang, G.I. Webb, X., Wu: Descretization methods. In DMKD 2010.

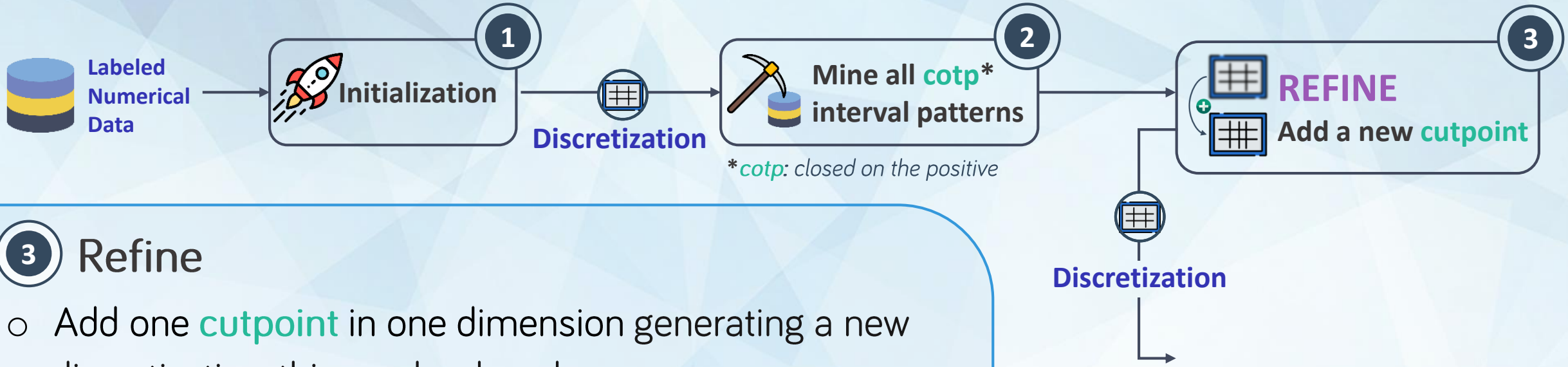
REFINE&MINE: Step-by-step (2)

2 Mine

- Compute all closed-on-the-positive (cotp [12]) patterns using MinIntChange algorithm [6].
- Compute the first guarantees.

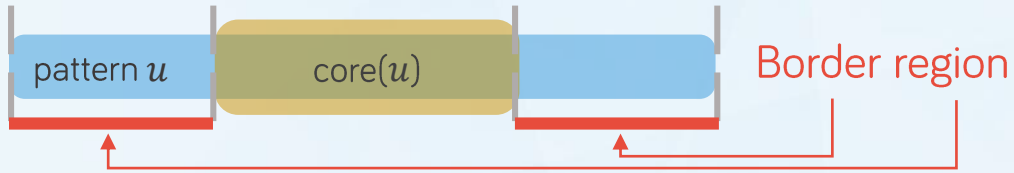
[6] M. Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli. Revisiting Numerical Pattern Mining with Formal Concept Analysis. In IJCAI 2011.
[12] C.G.. Garriga, P. Kralj, N. Lavrac. Closed sets for labeled data. In JMLR 2008.

REFINE&MINE: Step-by-step (3)

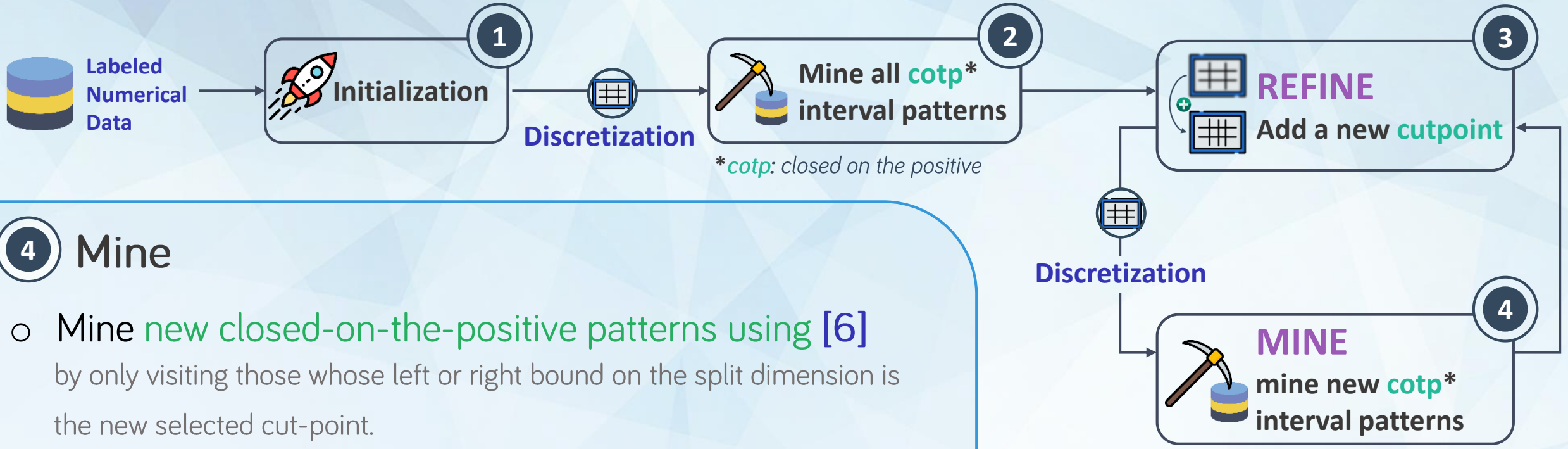


3 Refine

- Add one cutpoint in one dimension generating a new discretization, this can be done by:
  - Adding a random cut-point on a random dimension or,
  - adding a cut-point belonging to the border region of the pattern yielding the highest accuracy bound.



REFINE&MINE: Step-by-step (4)

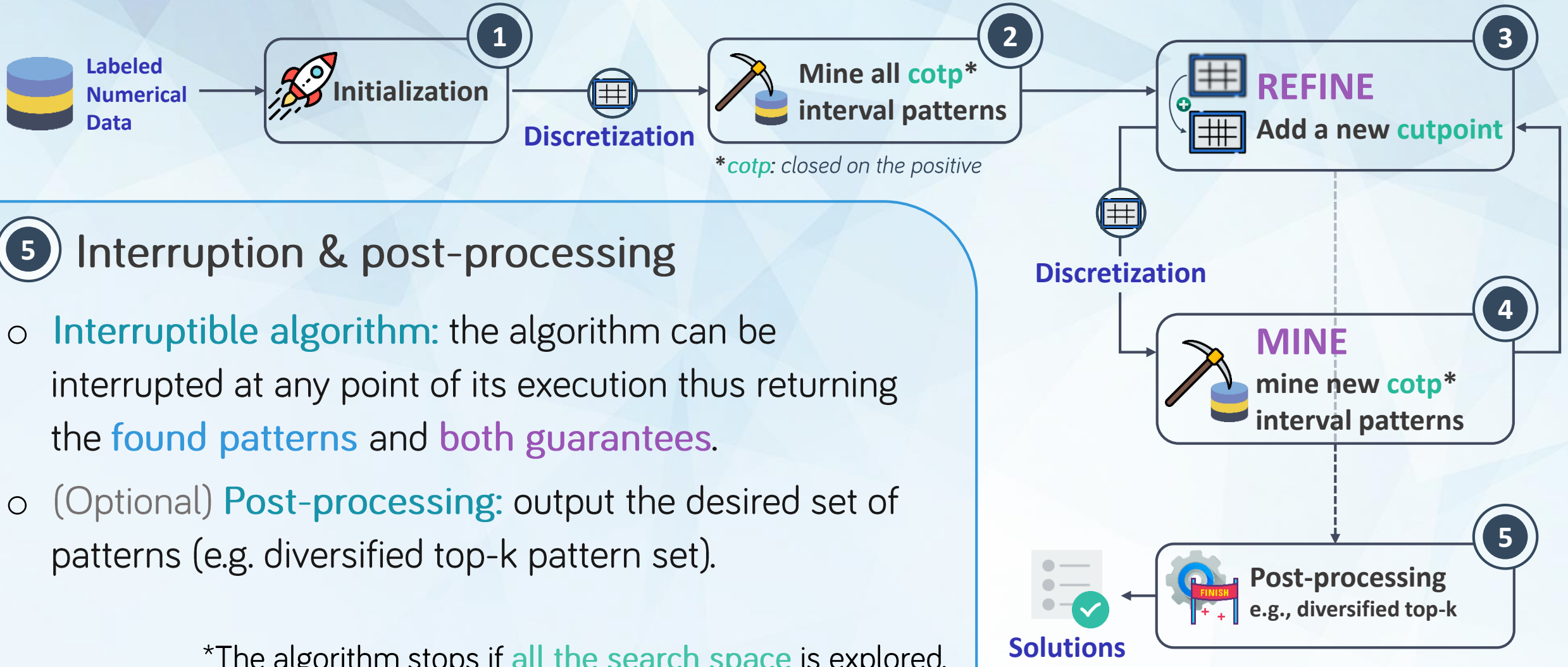


4 Mine

- Mine new closed-on-the-positive patterns using [6]
  by only visiting those whose left or right bound on the split dimension is the new selected cut-point.
- Update both guarantees
  An index structure is used to update quickly the guarantees

[6] M. Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli. Revisiting Numerical Pattern Mining with Formal Concept Analysis. In IJCAI 2011.

REFINE&MINE: Step-by-step (5)



5 Interruption & post-processing

- Interruptible algorithm: the algorithm can be interrupted at any point of its execution thus returning the found patterns and both guarantees.
- (Optional) Post-processing: output the desired set of patterns (e.g. diversified top-k pattern set).

*The algorithm stops if all the search space is explored.

EMPIRICAL EVALUATION

Experiment Plan

Accuracy Guarantee

A bound on the difference of quality between the best quality pattern in D and the best found pattern in S.

Specificity Guarantee

A bound on the distance between the set of all patterns D and the set of already found patterns S.

Comparison with MCTS4DM

MCTS4DM is the closest approach to Refine&Mine in terms of the underlying paradigm, i.e. Anytime Algorithms.

Experiment Plan

Accuracy Guarantee

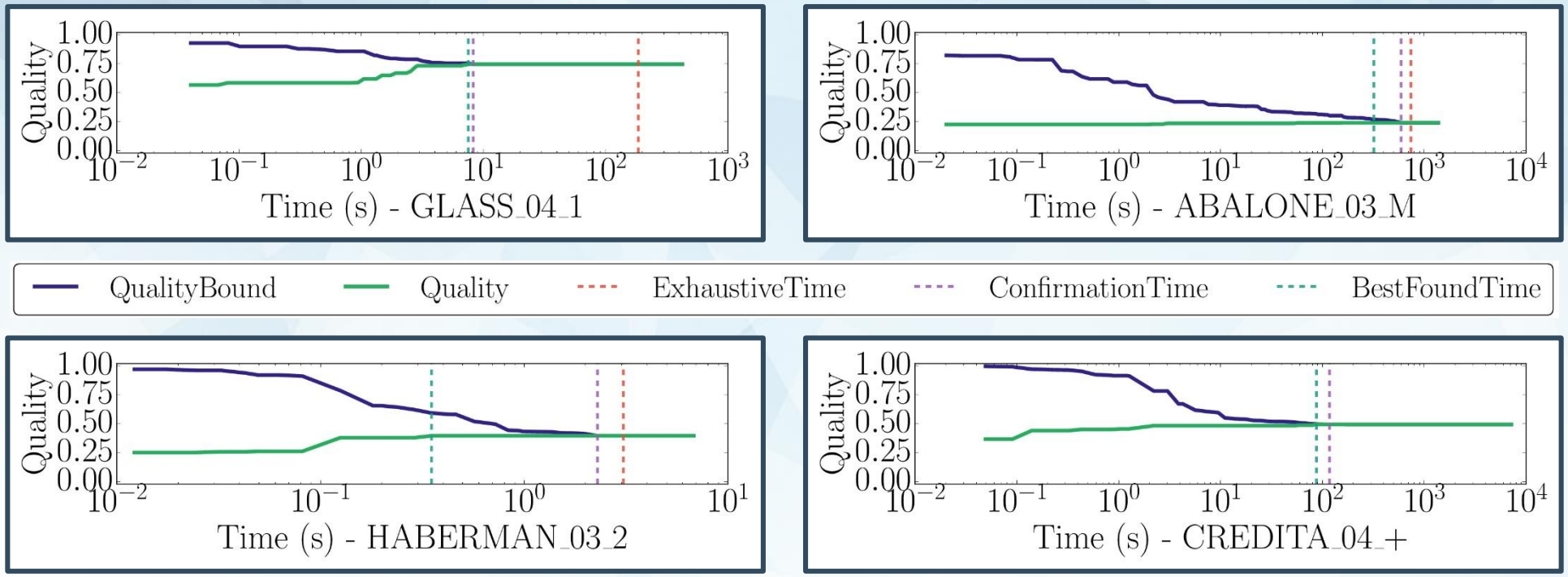
A bound on the difference of quality between the best quality pattern in D and the best found pattern in S.

Bottom line

Experiments were performed over 25 UCI Datasets

- 5 out of 25 Datasets: Exhaustive Algorithm achieved a better time in ensuring the discovery of the best pattern.
- 6 out of 25 Datasets: Exhaustive Algorithm was not able to finish within 2 Hours
- Refine&Mine required in average 40% of the time to guarantee the discovery of the best pattern

Experiment Plan: Accuracy Guarantee



Accuracy Guarantee

Bottom line Experiments were performed over 25 UCI Datasets

- 5 out of 25 Datasets: Exhaustive Algorithm achieved a better time in ensuring the discovery of the best pattern.
- 6 out of 25 Datasets: Exhaustive Algorithm was not able to finish within 2 Hours
- Refine&Mine required in average 40% of the time to guarantee the discovery of the best pattern

Experiment Plan

Specificity Guarantee

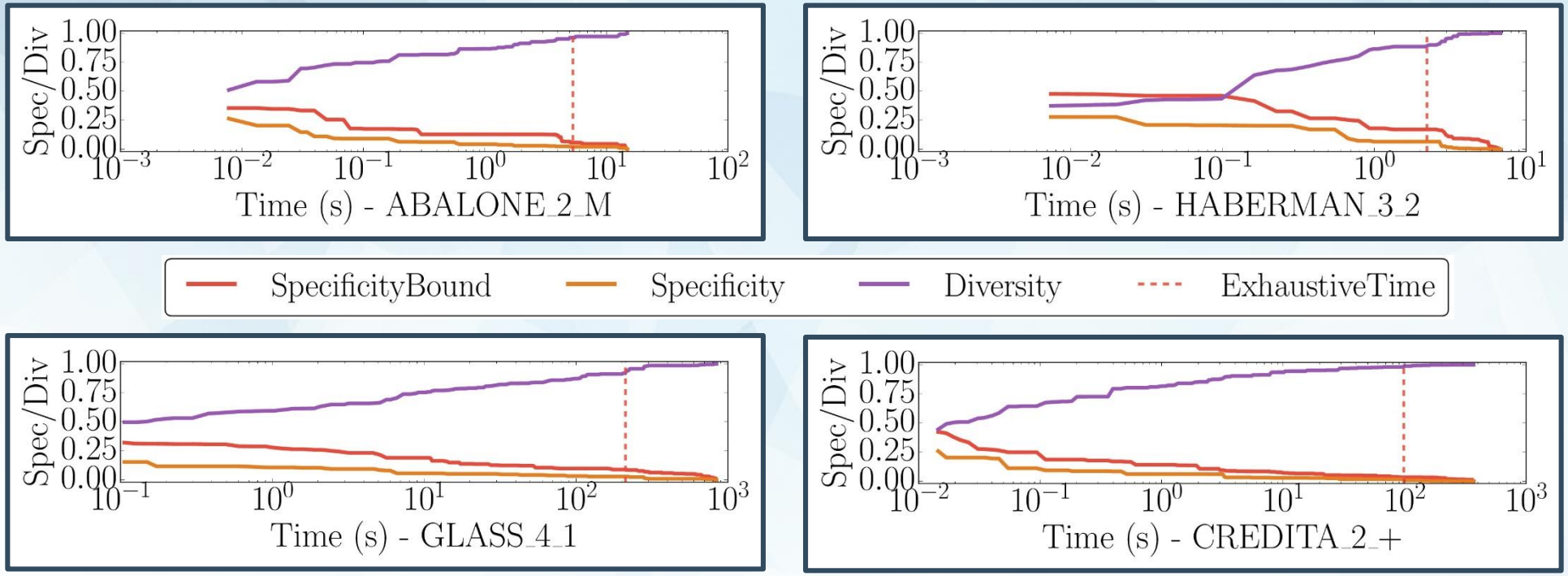
A bound on the distance between the set of all patterns D and the set of already found patterns S.

Bottom line

Experiments were performed over 15 UCI Datasets

Refine&Mine achieves to find a 80% approximation of the top-k diversified patterns set found by the exhaustive search algorithm in less than 20% of the required time of the exhaustive search algorithm

Experiment Plan: Specificity Guarantee

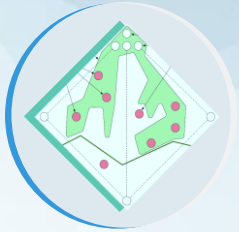


Specificity Guarantee

Bottom line Experiments were performed over 15 UCI Datasets

- Refine&Mine achieves to find a 80% approximation of the top-k diversified patterns set found by the exhaustive search algorithm in less than 20% of the required time of the exhaustive search algorithm

Experiment Plan



Comparison with MCTS4DM

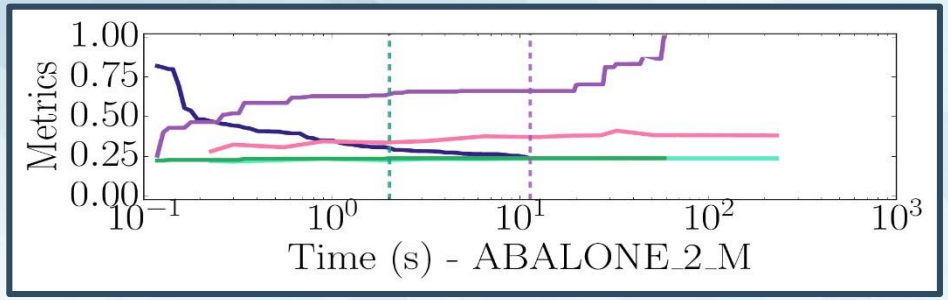
MCTS4DM is the closest approach to Refine&Mine in terms of the underlying paradigm, i.e. Anytime Algorithms.

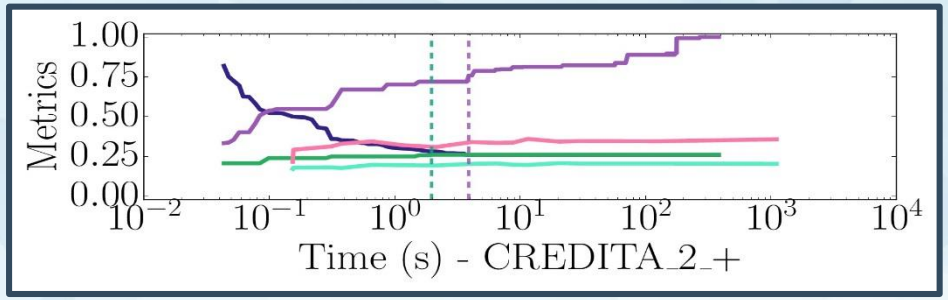
Bottom line

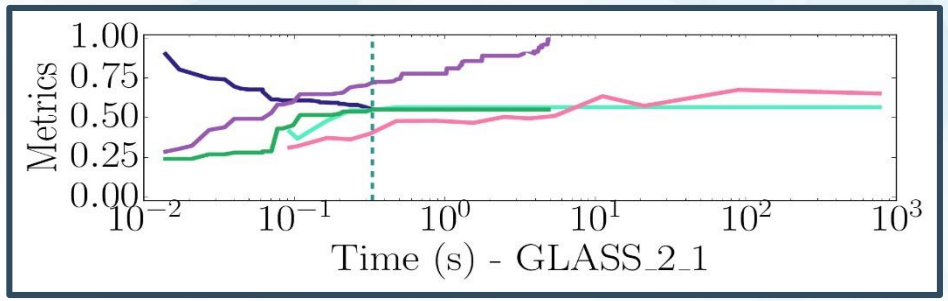
Experiments were performed over 6 UCI Datasets

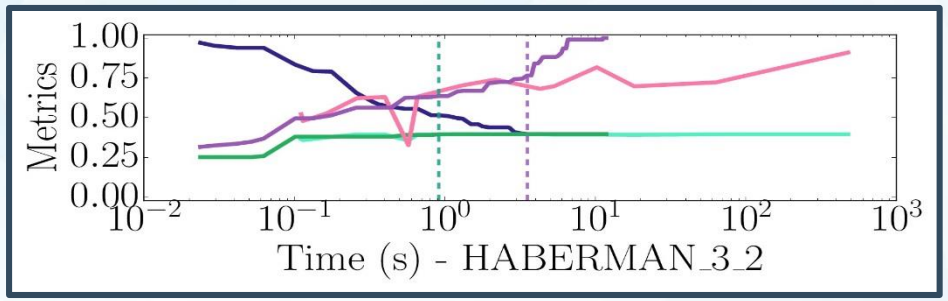
- In all configurations: Refine&Mine ensured a full traversal of the search space before MCTS4DM
- In all configurations: In average, Refine&Mine achieved 25% more diversity (after 10s of execution time) than MCTS4DM

Experiment Plan: Comparison with MCTS4DM







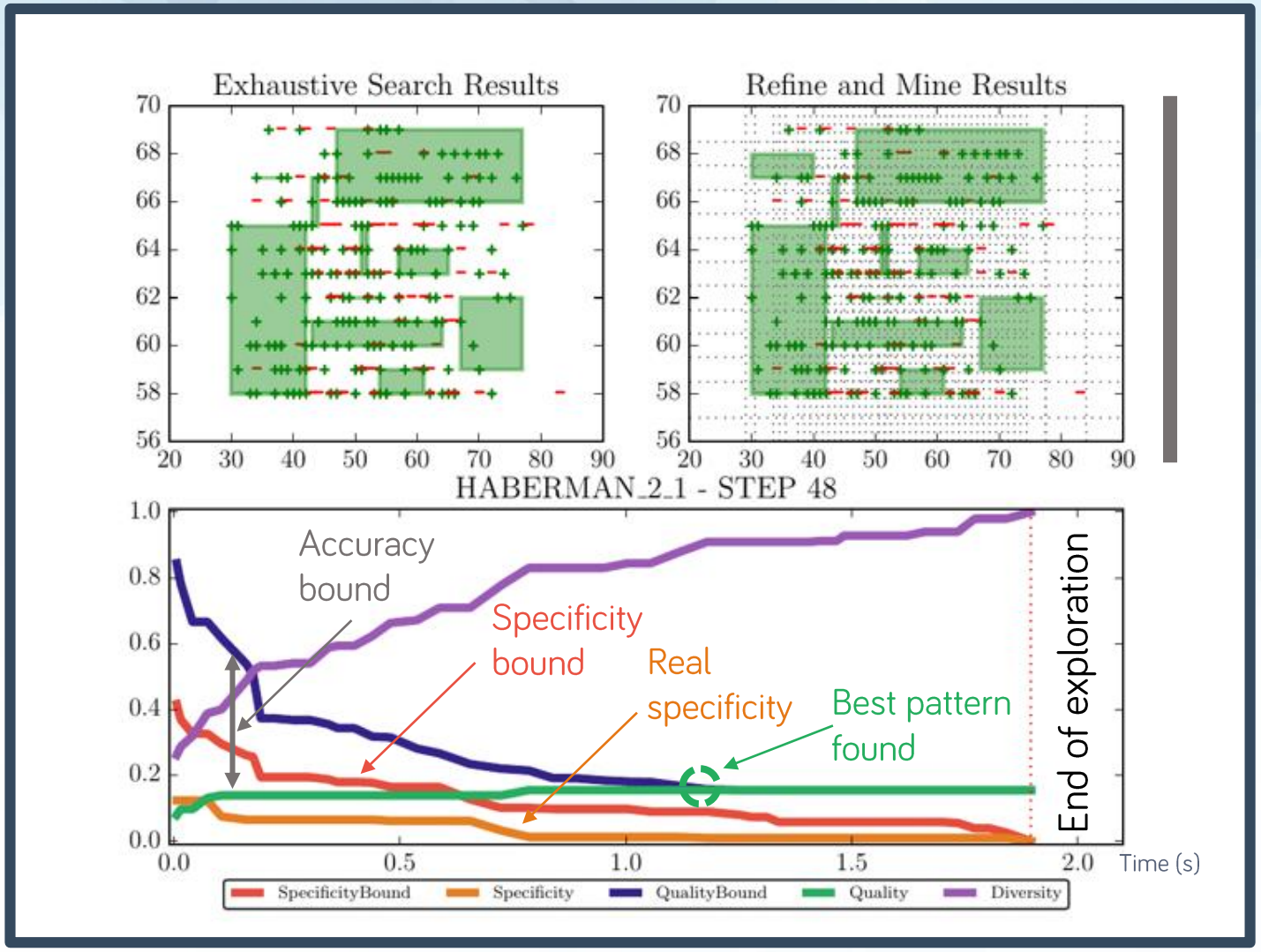


R&M vs. MCTS4DM

Bottom line Experiments were performed over 6 UCI Datasets

- In all configurations: Refine&Mine ensured a full traversal of the search space before MCTS4DM
- In all configurations: In average, Refine&Mine achieved 25% more diversity (after 10s of execution time) than MCTS4DM

REFINE&MINE: Wrap-up



Empirical Evaluation

Experiments were performed over

25 UCI Datasets

- Refine&Mine achieves to find a 80% ground truth in less than 20% of the time required for the exhaustive search algorithm to finish
- Refine&Mine achieved better results both in finding the best pattern and a high diversified results set compared to MCTS4DM [*]

[*] *G. Bosc, J.F. Boulicaut, C. Raïssi, M. Kaytoue. Anytime discovery of a diverse set of patterns with MCTS. In DMKD 2018.*