FACULTY OF INFORMATICS AND STATISTICS

Lukas Sykora, Tomas Kliegr
Department of Information and Knowledge Engineering

Action Rule Mining

Prague University of Economics and Business

What Is an Action Rule?

IF Antecedent **THEN** Consequent

$$r = [\overbrace{\overbrace{(Sex : \underline{f})}^{\text{Value of Stable attribute}} \wedge \overbrace{(Embarked : \underline{S} \rightarrow \underline{C}) \wedge (Pclass : \underline{3} \rightarrow \underline{1})}^{\text{Changes in Flexible attributes}}}^{\text{Antecedent}}] \Rightarrow \overbrace{\overbrace{[Surv. : \underline{0} \rightarrow \underline{1}]}^{\text{Change in Decision}}}^{\text{Consequent}},$$

$$\textit{with support } 5\%,\ \textit{confidence } 59\% \textit{ and uplift } 5.7\%$$

Undesired Part ($r_{undesired}$) - The initial condition under which the undesired target value occurs.
Desired Part ($r_{desired}$) - The condition after applying the change, under which the desired target value occurs.

Key Concepts in Frequent Itemset Mining

- **Item** - A single attribute-value pair, e.g., Age=Young, Class=2nd.
- **Itemset** - A set of items (attribute-value pairs), e.g., {Sex=Female, Age=Young, Class=2nd}.
- **Antecedent** - The if part of a rule; the conditions that need to be met.
  E.g., Sex=Female $\wedge$ Age=Young $\wedge$ Class=2nd.
- **Consequent** - The then part of a rule; the predicted outcome, e.g., Survived=Yes.
- **Support** - Number of records in the dataset that contain a given itemset. Example: Support of {Sex=Female, Age=Young} is the number of passengers matching both.
- **Confidence** - Probability that the rule's consequent is true when the antecedent holds.
  For rule $A \Rightarrow B$: confidence = support($A \wedge B$) / support(A)

Action rule mining builds on definitions from frequent itemset mining

- **Support of action** rule is min(support($r_{undesired}$), support($r_{desired}$))
- **Confidence of action** rule is confidence($r_{undesired}$) × confidence($r_{desired}$)
- **Uplift of action rule** is a new metric (more details later)

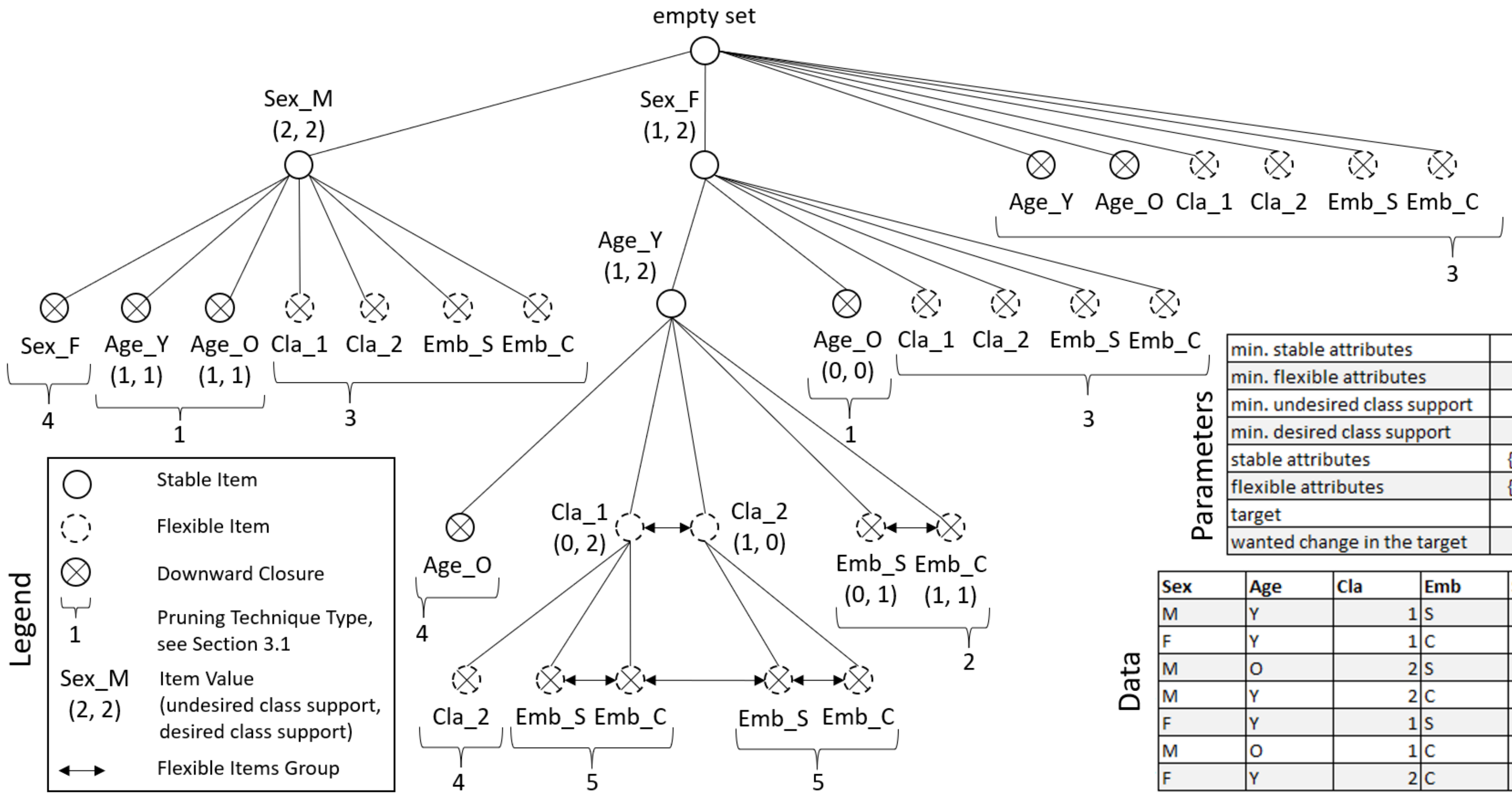
Current State – Algorithms

- Rule-based approach
  - Ras, Z. W., & Wieczorkowska, A. (2000). Action-rules: How to increase profit of a company.
  - Raś, Z. W., Wyrzykowska, E., & Wasyluk, H. (2007). Aras: Action rules discovery based on agglomerative strategy.
  - Ras, Z. W., & Tsay, L.-S. (2003). Discovering extended action-rules (system dear).
- Object-based approach
  - He, Z., Xu, X., Deng, S., & Ma, R. (2005). Mining action rules from scratch.
  - Hashemi, S., & Shamsinejad, P. (2021). Ga2rm: A ga-based action rule mining method.

Current State – Software

- LISp-Miner
  - Based on the General Unary Hypotheses Automaton (GUHA).
    [Rauch Jan, Š. M. (2014). Lisp-miner, GUHA a data mining. Oeconomica]
- SCARI
  - Separate and Conquer Algorithm for Action Rules and Recommendations Induction.
    [Sikora, M., Matyszok, P., & Wróbel, Ł. (2022). Scari: Separate and conquer algorithm for action rules and recommendations induction. Information Sciences, 607, 849–868.]

Contribution: Action-Apriori Algorithm

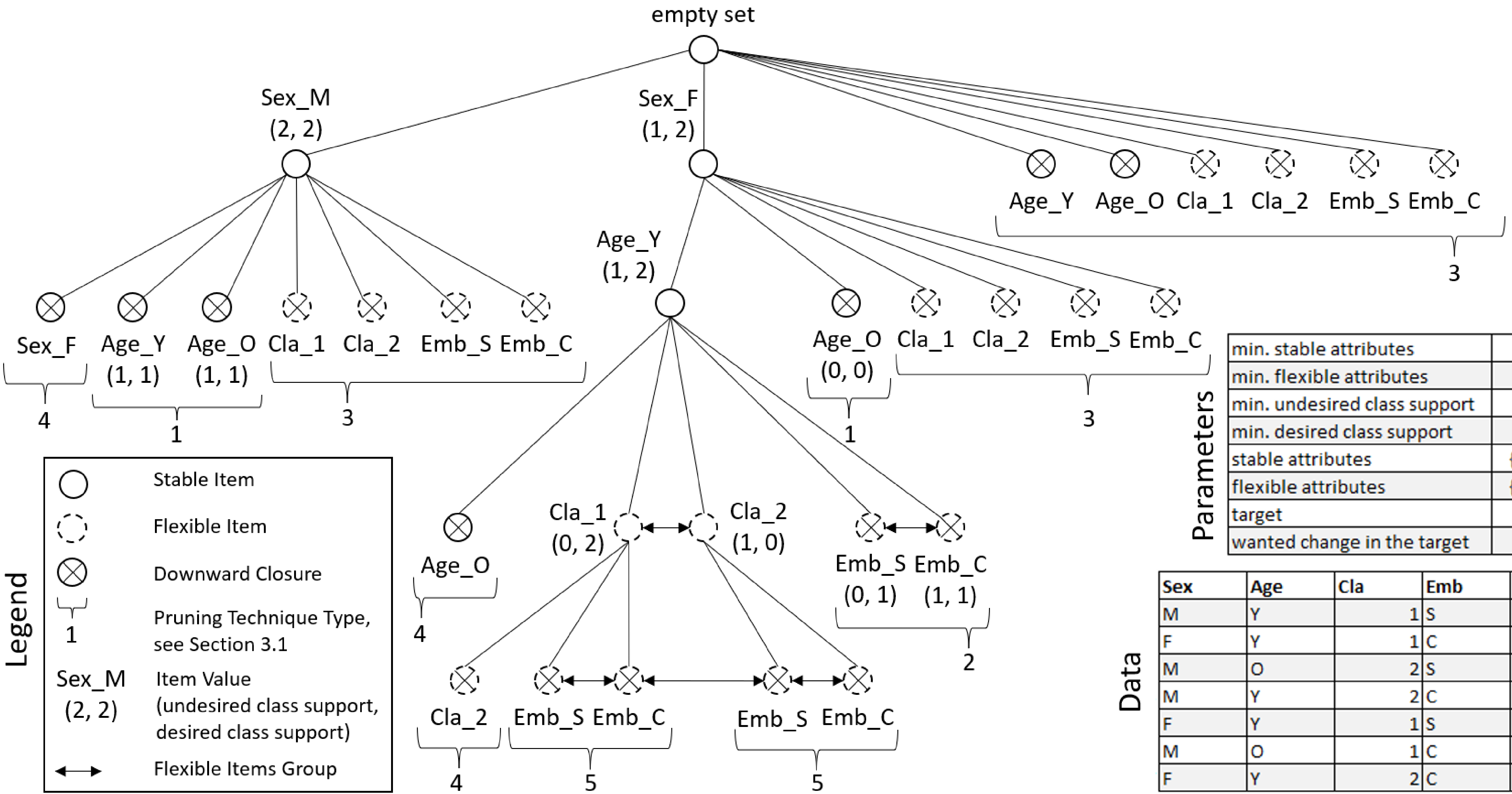


Parameters

min. stable attributes	2
min. flexible attributes	1
min. undesired class support	1
min. desired class support	2
stable attributes	{Sex, Age}
flexible attributes	{Cla, Emb}
target	Sur
wanted change in the target	0 -> 1

Data

Sex	Age	Cla	Emb	Sur
M	Y	1	S	1
F	Y	1	C	1
M	O	2	S	0
M	Y	2	C	0
F	Y	1	S	1
M	O	1	C	1
F	Y	2	C	0



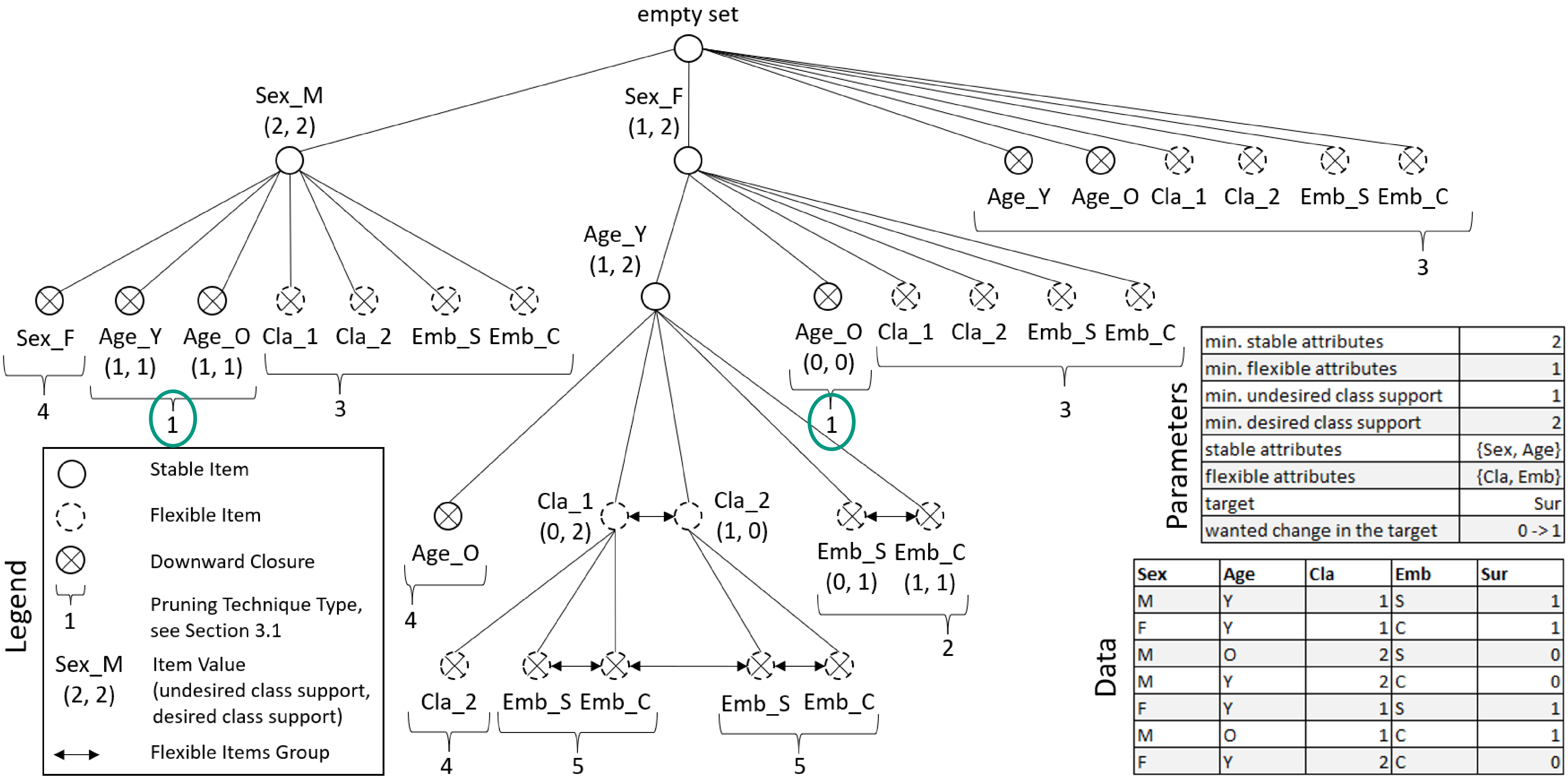
Parameters

min. stable attributes	2
min. flexible attributes	1
min. undesired class support	1
min. desired class support	2
stable attributes	{Sex, Age}
flexible attributes	{Cla, Emb}
target	Sur
wanted change in the target	0 -> 1

Data

Sex	Age	Cla	Emb	Sur
M	Y	1	S	1
F	Y	1	C	1
M	O	2	S	0
M	Y	2	C	0
F	Y	1	S	1
M	O	1	C	1
F	Y	2	C	0

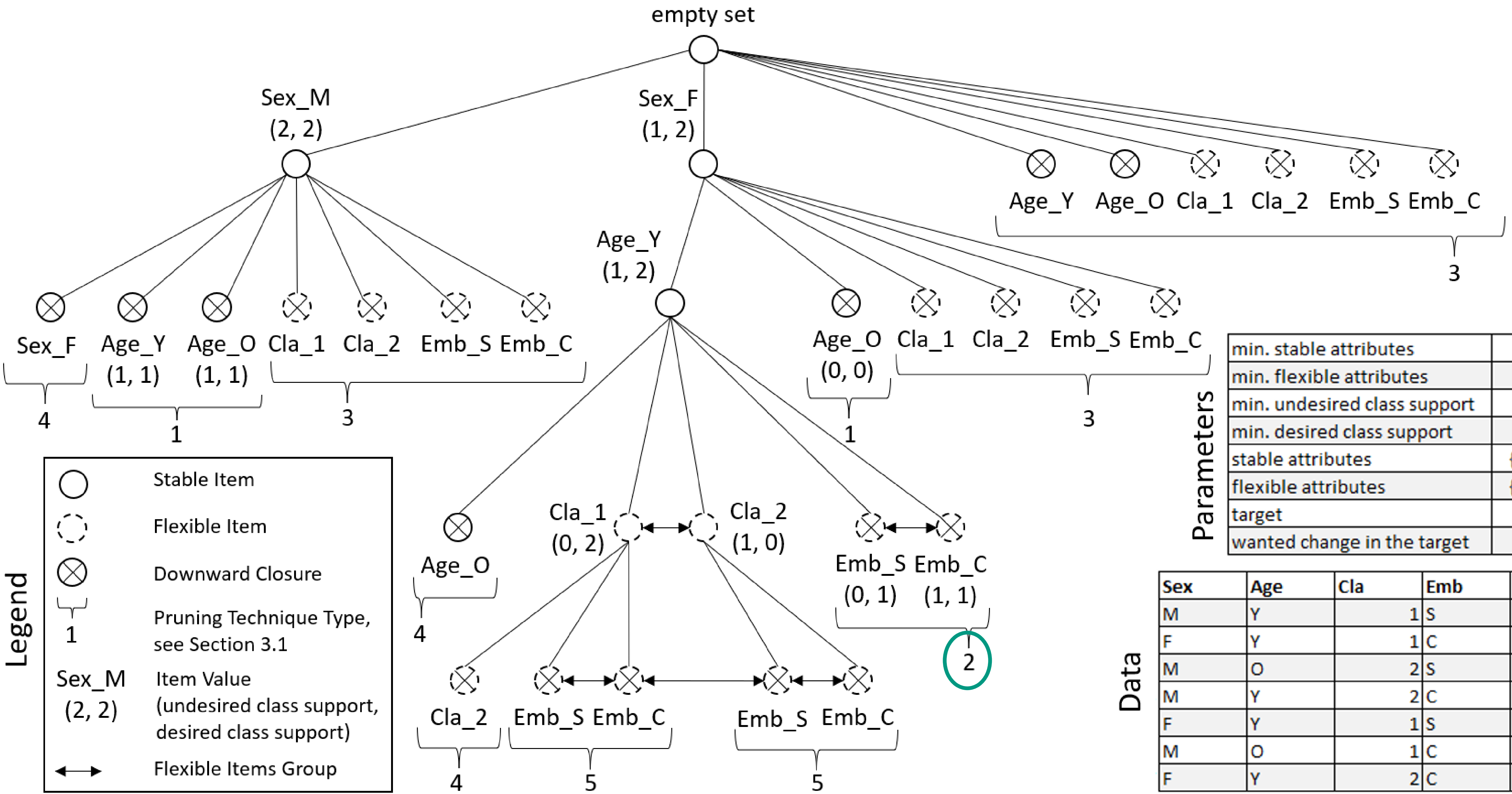
1) Nodes derived from stable attributes must have sufficient support for both desired and undesired classes in the consequent.



min. stable attributes	2
min. flexible attributes	1
min. undesired class support	1
min. desired class support	2
stable attributes	{Sex, Age}
flexible attributes	{Cla, Emb}
target	Sur
wanted change in the target	0 -> 1

Sex	Age	Cla	Emb	Sur
M	Y	1	S	1
F	Y	1	C	1
M	O	2	S	0
M	Y	2	C	0
F	Y	1	S	1
M	O	1	C	1
F	Y	2	C	0

2) Groups of nodes representing flexible attributes must have sufficient support for both desired and undesired classes.



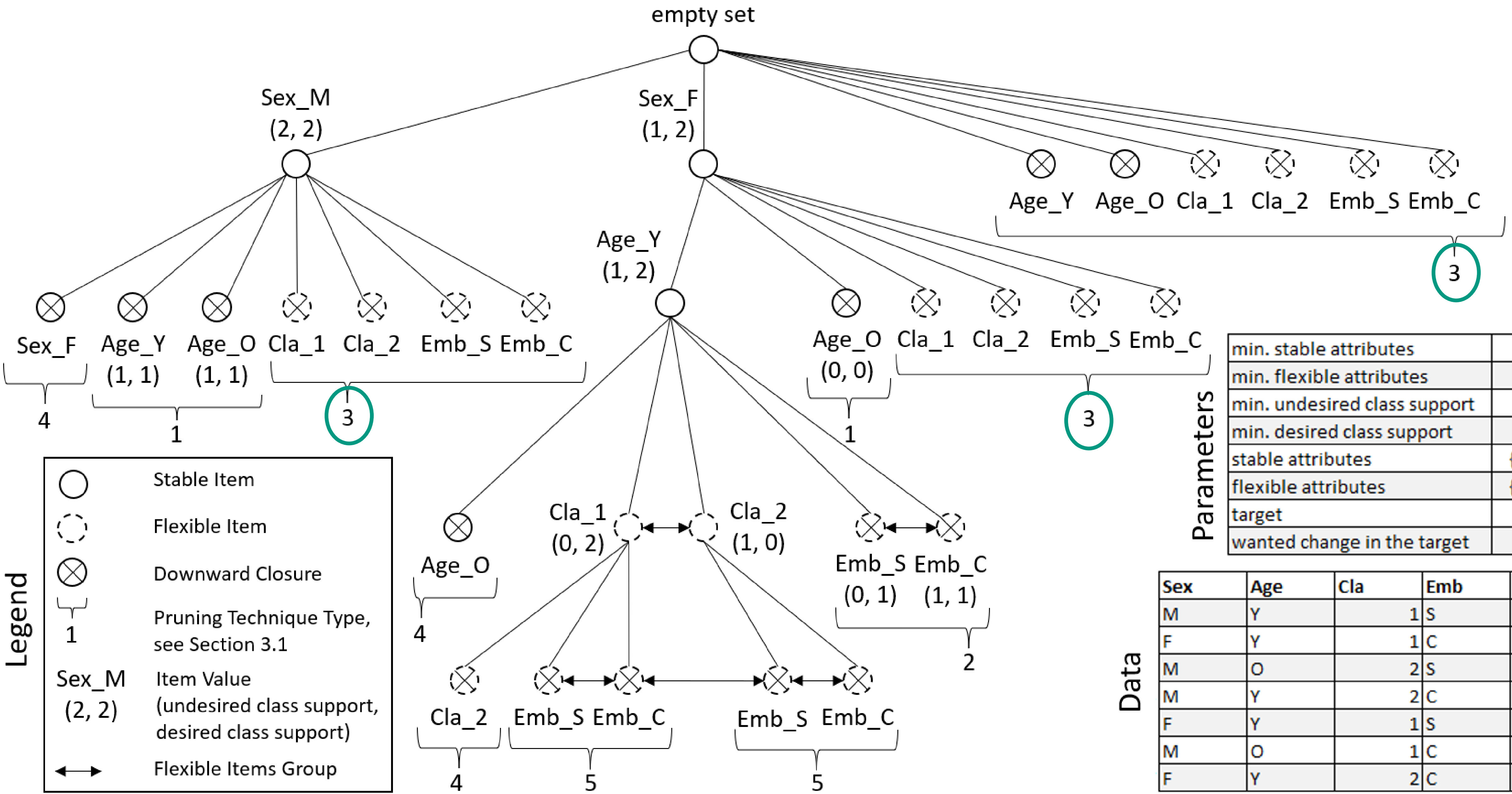
Parameters

min. stable attributes	2
min. flexible attributes	1
min. undesired class support	1
min. desired class support	2
stable attributes	{Sex, Age}
flexible attributes	{Cla, Emb}
target	Sur
wanted change in the target	0 -> 1

Data

Sex	Age	Cla	Emb	Sur
M	Y	1	S	1
F	Y	1	C	1
M	O	2	S	0
M	Y	2	C	0
F	Y	1	S	1
M	O	1	C	1
F	Y	2	C	0

3) The required minimum number of stable or flexible attributes must be met.



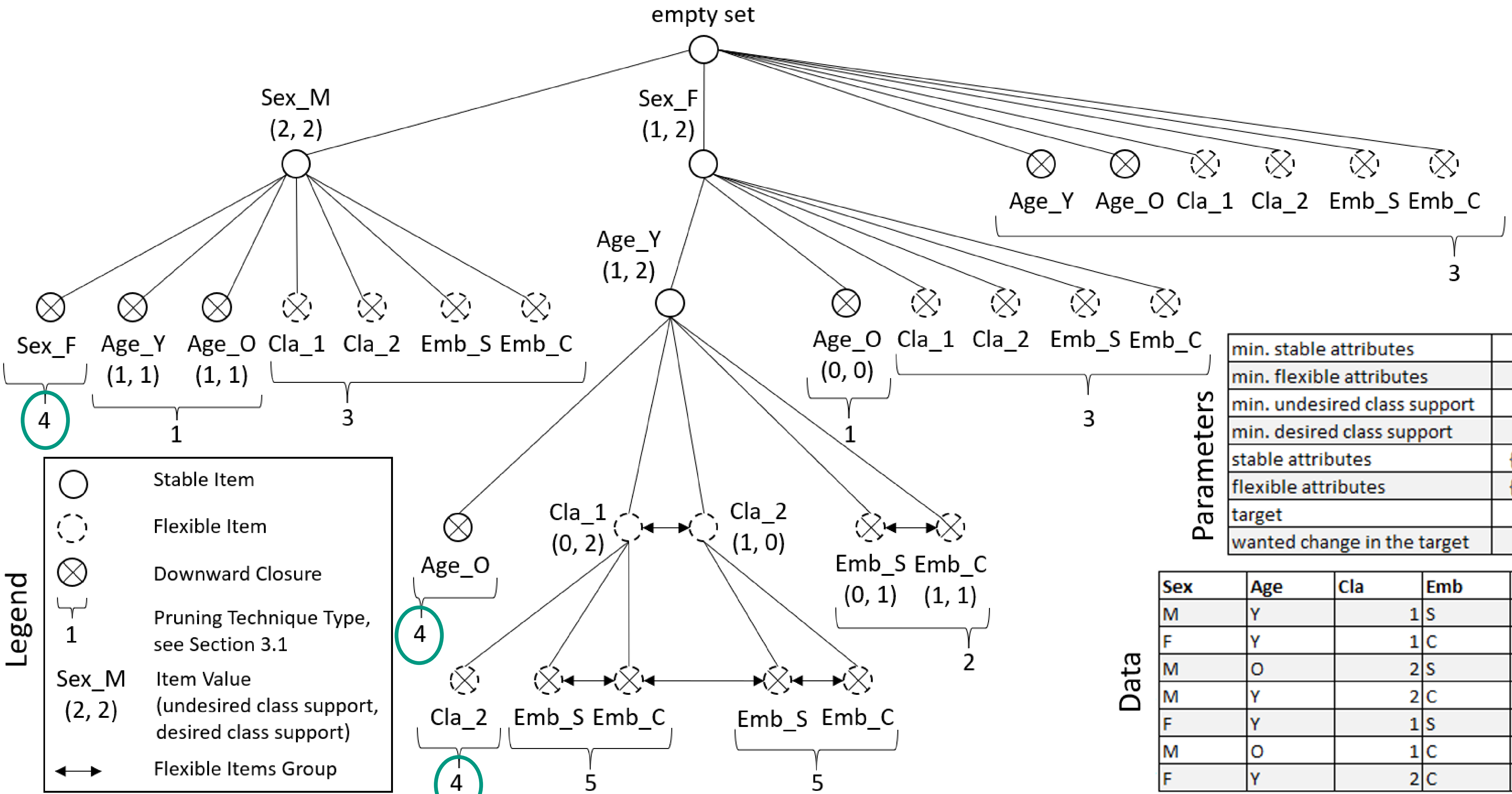
Parameters

min. stable attributes	2
min. flexible attributes	1
min. undesired class support	1
min. desired class support	2
stable attributes	{Sex, Age}
flexible attributes	{Cla, Emb}
target	Sur
wanted change in the target	0 -> 1

Data

Sex	Age	Cla	Emb	Sur
M	Y	1	S	1
F	Y	1	C	1
M	O	2	S	0
M	Y	2	C	0
F	Y	1	S	1
M	O	1	C	1
F	Y	2	C	0

4) Each path in the trie can contain at most one item derived from the same attribute.



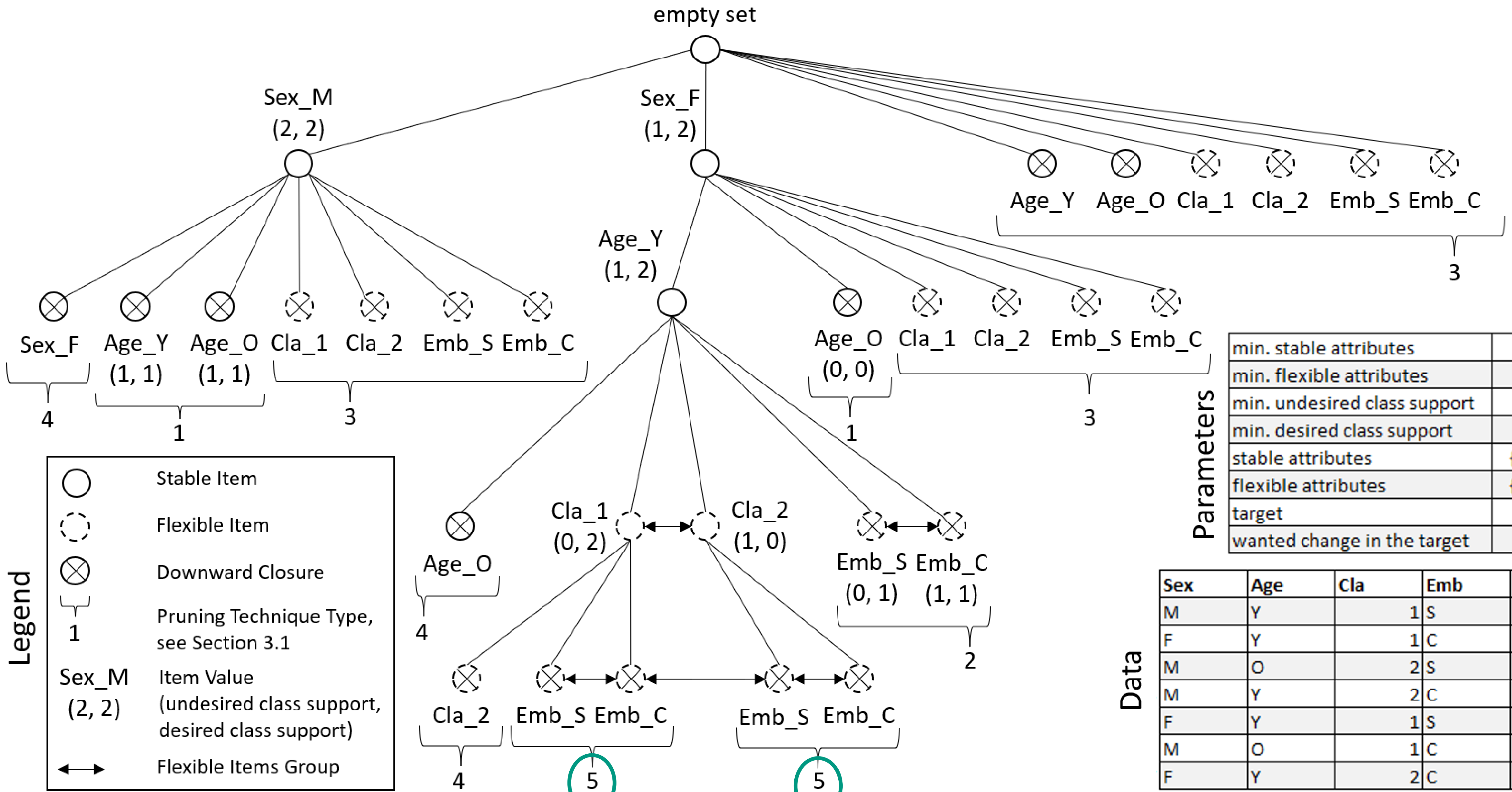
Parameters

min. stable attributes	2
min. flexible attributes	1
min. undesired class support	1
min. desired class support	2
stable attributes	{Sex, Age}
flexible attributes	{Cla, Emb}
target	Sur
wanted change in the target	0 -> 1

Data

Sex	Age	Cla	Emb	Sur
M	Y	1	S	1
F	Y	1	C	1
M	O	2	S	0
M	Y	2	C	0
F	Y	1	S	1
M	O	1	C	1
F	Y	2	C	0

5) The itemset is on the stop list.



Parameters

min. stable attributes	2
min. flexible attributes	1
min. undesired class support	1
min. desired class support	2
stable attributes	{Sex, Age}
flexible attributes	{Cla, Emb}
target	Sur
wanted change in the target	0 -> 1

Data

Sex	Age	Cla	Emb	Sur
M	Y	1	S	1
F	Y	1	C	1
M	O	2	S	0
M	Y	2	C	0
F	Y	1	S	1
M	O	1	C	1
F	Y	2	C	0

Contribution: New measure for action rules

Existing measures for action rules are hard to interpret.

$$uplift = P(\text{decision} \mid \text{treatment}) - P(\text{decision} \mid \text{no treatment})$$ [Radcliffe (2007)]

$$uplift(r_a) = \frac{\left(\overbrace{conf_{r_{desired}}}^{\text{Confidence of desired class for desired rule}} - \overbrace{(1 - conf_{r_{undesired}})}^{\text{Confidence of desired class for undesired rule}}\right) * \overbrace{\frac{suppr_{undesired}}{conf_{r_{undesired}}}}^{\text{Number of transactions that can be changed}}}{\underbrace{|t \in D : t|}_{\text{Number of transactions}}}$$

Uplift quantifies the increase of desired target states in a dataset.

Contribution: New pruning technique for Action Rules

Dominant Action Rules

A dominant action rule is defined as a rule that cannot be extended by additional conditions without reducing its uplift. In cases where an extension of the rule leads to a higher uplift, a new dominant action rule is formed. Inspired by the concept of closed association rules (Pasquier et al., 1999).

Dominant $\mathrm{rigor} = (\mathrm{medium} \to \mathrm{high}) \Rightarrow \mathrm{evaluation} = (\mathrm{bad} \to \mathrm{good})$

with uplift 16.96%.

Not Dominant $\mathrm{rigor} = (\mathrm{medium} \to \mathrm{high}) \wedge \mathrm{grammar} = (1 \to 0) \Rightarrow \mathrm{evaluation} = (\mathrm{bad} \to \mathrm{good})$

with uplift 10.82%.

Contribution: High-Utility Action Rules Mining

$$u(r_{1\to 2}) = u(r_2) - u(r_1) = (ul(\phi_2) - ul(\phi_1)) \times (\mathrm{conf}(r_2) - (1 - \mathrm{conf}(r_1))) + \sum_{n=1}^{k} \left(ul(\alpha_2^{(n)}) - ul(\alpha_1^{(n)}) \right)$$

where $ul(\phi_2)$ represents the utility of the desired class, $ul(\phi_1)$ represents the utility of the undesired class, and $ul(\alpha_1^{(n)})$ and $ul(\alpha_2^{(n)})$ represent the utilities of items derived from flexible attributes before and after the recommended action. The expression $ul(\phi_2) - ul(\phi_1)$ computes the difference in utility between the desired and undesired classes. This value is multiplied by $conf(r_2) - (1 - conf(r_1))$, which represents the change in confidence of the desired class as predicted by r_2 compared to its confidence as predicted by r_1. The final part of the formula, $\sum_{n=1}^{k}(ul(\alpha_2^{(n)}) - ul(\alpha_1^{(n)}))$, captures the total change in utility resulting from the recommended actions.

Item	Utility
Salary: Low	-300
Salary: Medium	-500
Salary: High	-1000
Attrition: False	700
Attrition: True	0

Contribution: Method for applying action rule mining to unstructured text data I

Feature Discovery → Feature Generation → Action Rules Mining

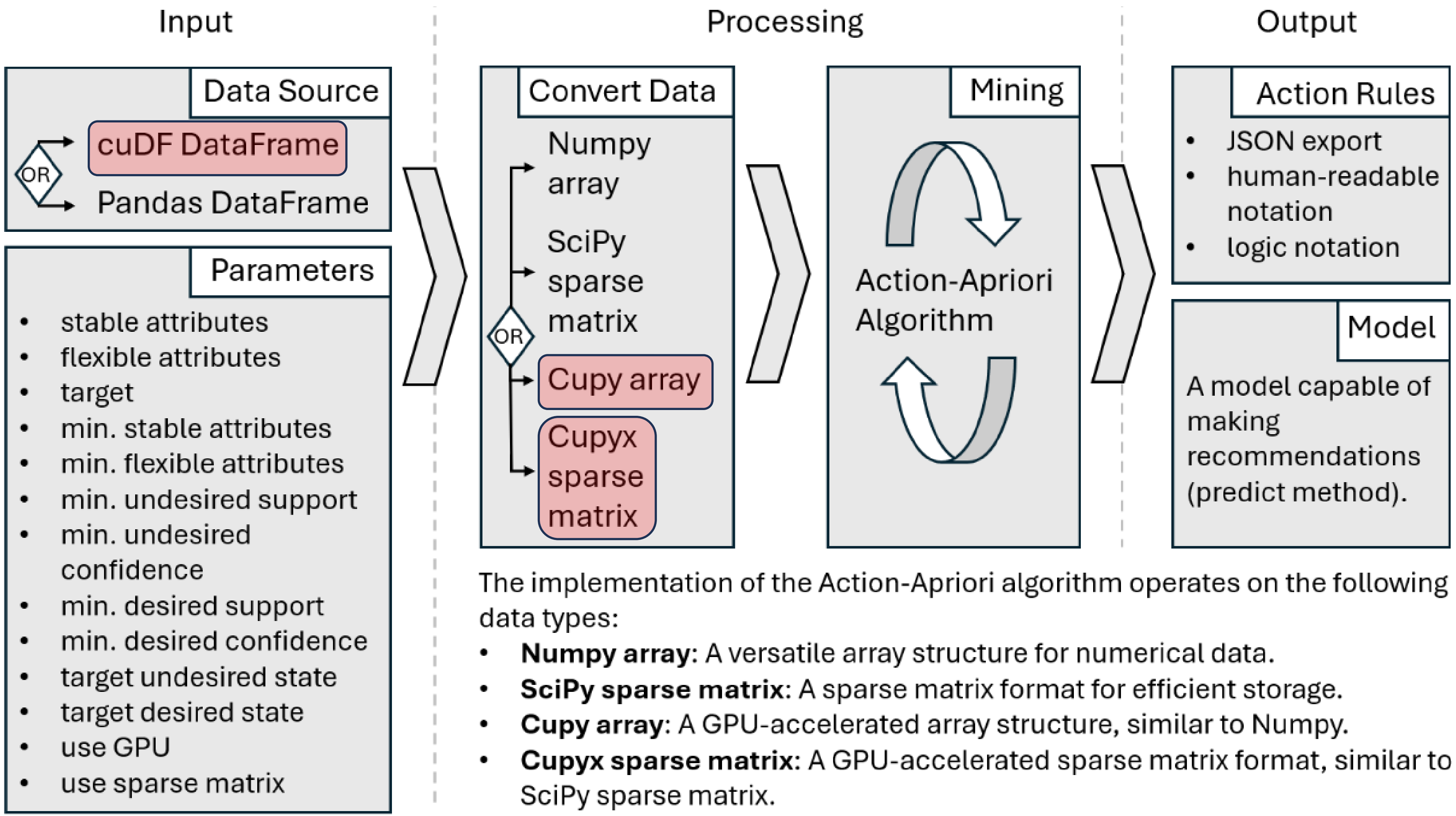
Feature discovery refers to the process of identifying what features should be extracted from the data — essentially discovering the names or types of features that may be relevant for a given task. In this context, an LLM analyzes the dataset and proposes which features to generate.

Feature generation is the process of extracting the values of those features for individual data instances. This also uses LLMs, which are provided with prompts to generate the specific feature values.

Contribution: Method for applying action rule mining to unstructured text data II

Rule	Area	Novelty	Rigor	Grammar	Replicability	Accessibility	Evaluation	Uplift
r_1			med → high		0 → 1		bad → good	16.96%
r_2			med → high				bad → good	16.95%
r_3				1 → 0	0 → 1		bad → good	11.26%
r_4		med → high			0 → 1		bad → good	6.73%
r_5		med → high		1 → 0			bad → good	4.36%
r_6		med → high		1			bad → good	3.96%
r_7		med → high	med → high				avg. → best	4.57%
r_8		med → high		1 → 0			avg. → best	3.58%
r_9		low → high	med → high				avg. → best	2.24%
r_{10}		low → high			0 → 1		avg. → best	1.95%
r_{11}		low → high		1 → 0			avg. → best	1.24%
r_{12}	compu.	med → high	med → high				bad → good	1.98%
r_{13}	elect.		med → high			med	bad → good	1.15%
r_{14}	mechan.	med → high	med → high	1 → 0	0 → 1		bad → good	1.51%
r_{15}	chemic.	med → high	med → high				bad → good	1.88%
r_{16}	mater.	med → high			0 → 1		bad → good	1.54%

Contribution: Use of GPU data structures for action rule mining



The implementation of the Action-Apriori algorithm operates on the following data types:

- **Numpy array:** A versatile array structure for numerical data.
- **SciPy sparse matrix**: A sparse matrix format for efficient storage.
- **Cupy array**: A GPU-accelerated array structure, similar to Numpy.
- **Cupyx sparse matrix**: A GPU-accelerated sparse matrix format, similar to SciPy sparse matrix.

Contribution: Python packages

Criteria / Packages	actionrules-lukassykora	action-rules
Algorithms	Baseline, ARAS, DEAR	Action-Apriori (author's algorithm)
Load classification rules	YES	NO
High-Utility Mining	YES	YES
Command-line interface	NO	YES
GPU Support	NO	YES
Multiple values per attribute	NO	YES

Contribution: User-friendly explanation in natural language

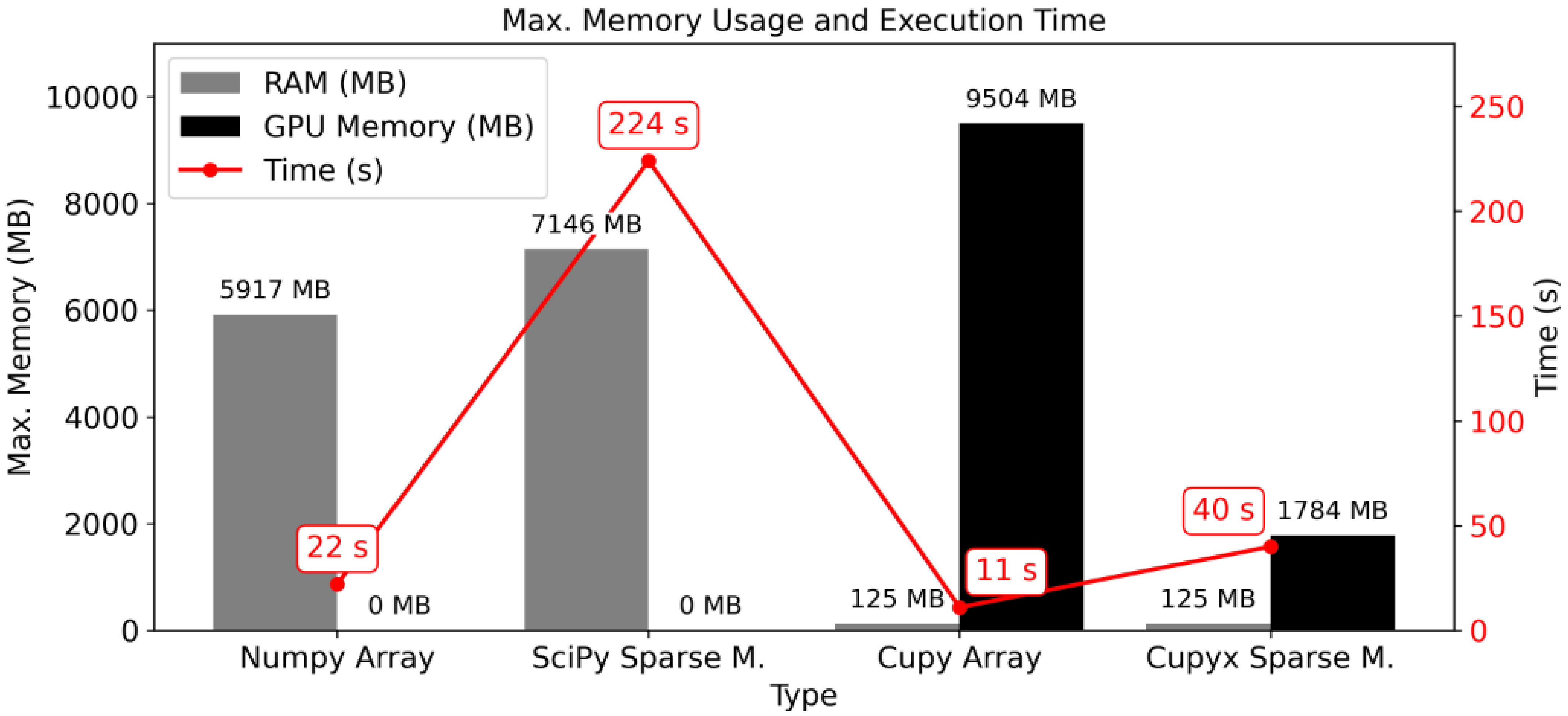
```
r = ar.get_rules().get_ar_notation()
for action_rule in r:
    print(action_rule)
```

$r = [(\text{gender: Female}) \wedge (\text{SeniorCitizen: 1}) \wedge (\text{InternetService}^{*}\text{: Fiber optic}) \wedge (\text{OnlineSecurity: No} \rightarrow \text{Yes}) \wedge (\text{DeviceProtection: No} \rightarrow \text{Yes}) \wedge (\text{StreamingTV: No} \rightarrow \text{Yes})] \Rightarrow [\text{Churn: Yes} \rightarrow \text{No}]$ with support of undesired part: 0.073, confidence of undesired part: 0.73, support of desired part: 0.029, confidence of desired part: 0.8286, and uplift: 0.0079.

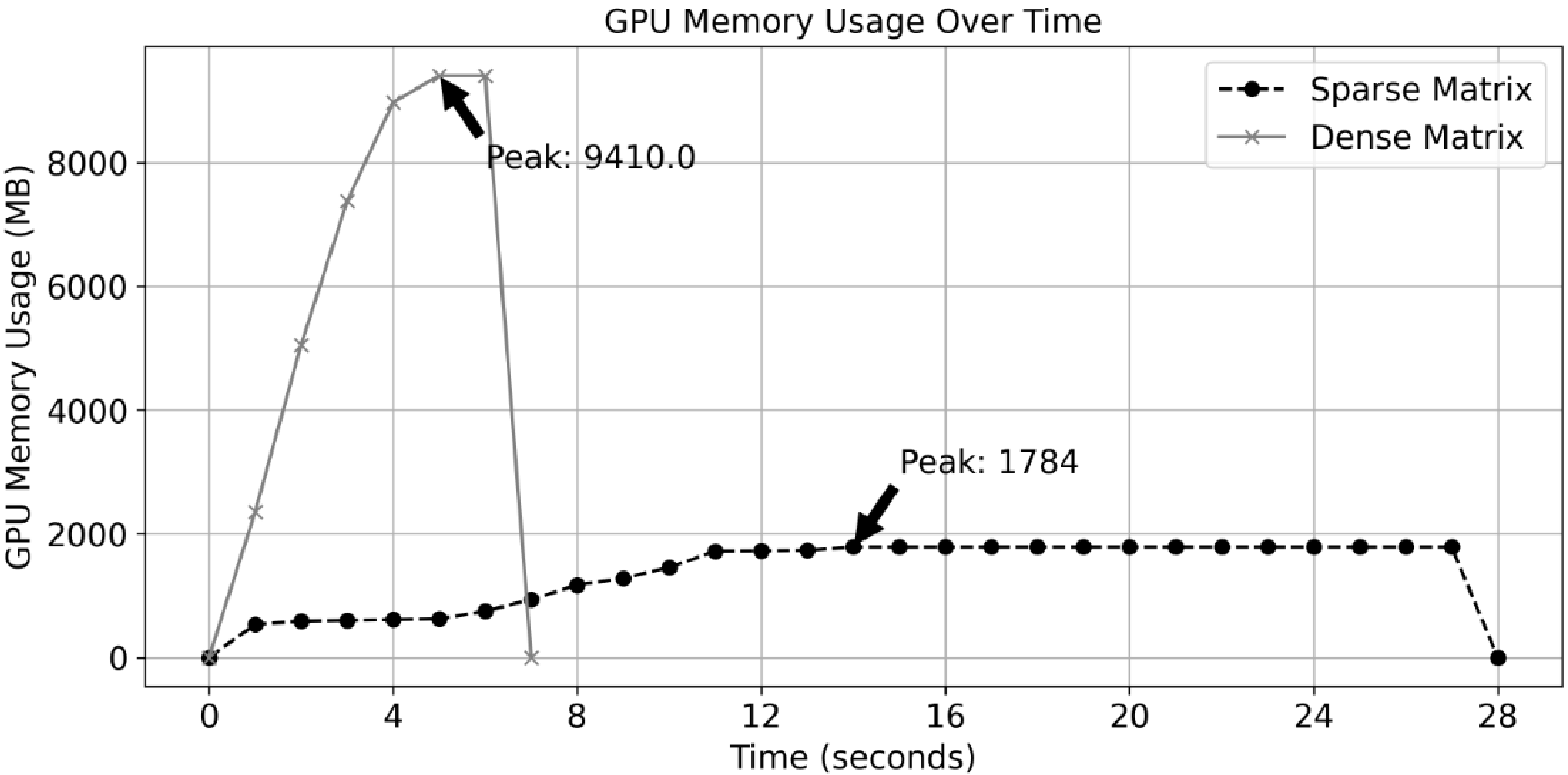
```
r2 = ar.get_rules().get_pretty_ar_notation()
for action_rule in r2:
    print(action_rule)
```

If attribute `gender` is `Female`, attribute `SeniorCitizen` is `1`, attribute (flexible is used as stable) `InternetService` is `Fiber optic`, attribute `OnlineSecurity` value `No` is changed to `Yes`, attribute `DeviceProtection` value `No` is changed to `Yes`, and attribute `StreamingTV` value `No` is changed to `Yes`, then `Churn` value `Yes` is changed to `No` with uplift: `0.007930873613111296`, support of undesired part: `73`, confidence of undesired part: `0.73`, support of desired part: `29`, and confidence of desired part: `0.828571428571428б`.

Improved scalability through the use of sparse matrices and GPU

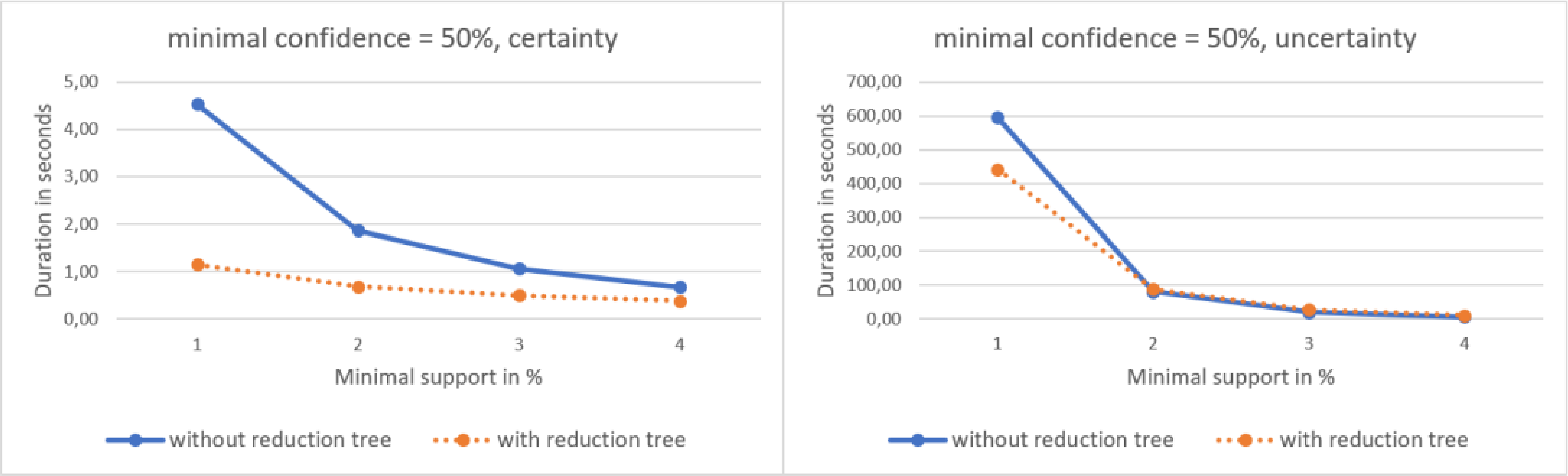


Evaluation: Benchmarking of data structures (GPU vs CPU processing)



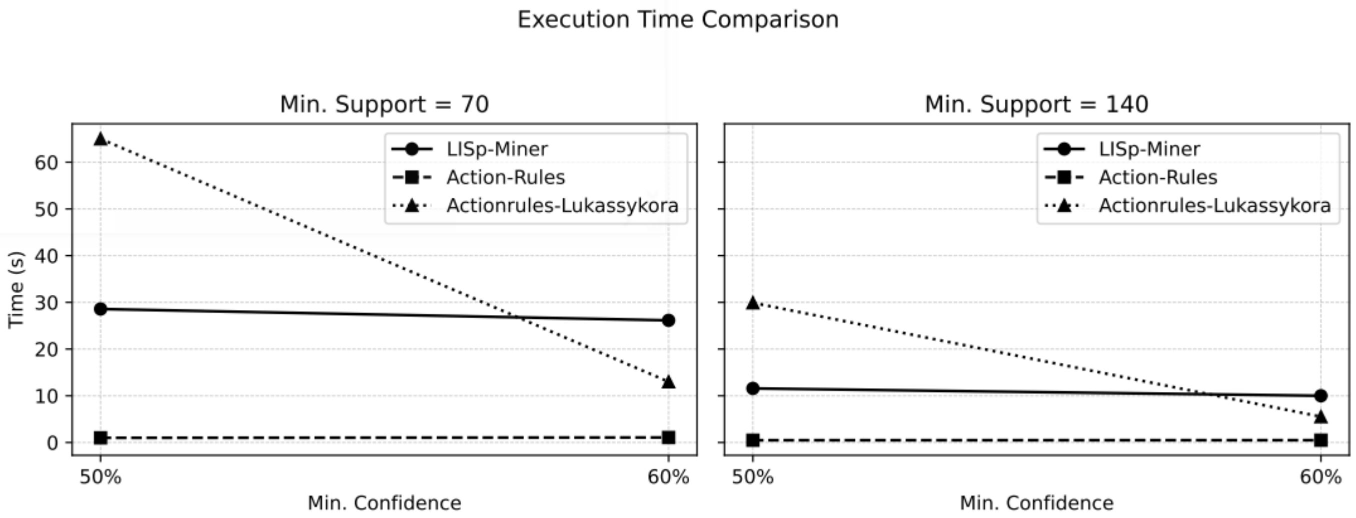
Evaluation: Scalability

The complexity of association rule learning depends on the support threshold.
In this experiment, we observed the impact of varying the minimum support threshold on the runtime of two of the implemented algorithms (Titanic dataset).



Validation of results against LISp-Miner and performance comparison

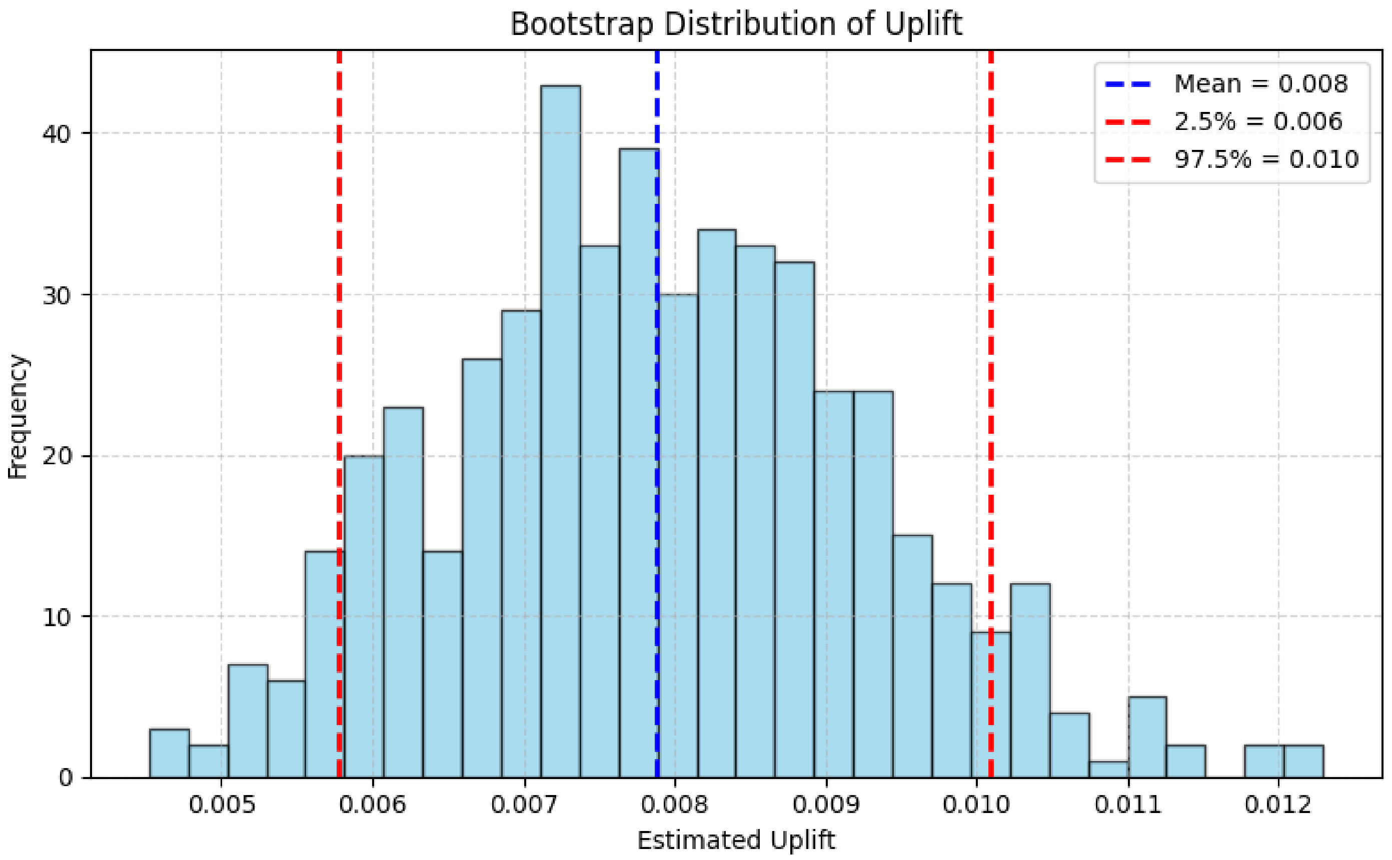
Setting		LISp-Miner		action-rules		actionrules-lukassykora	
Min. Sup	Min. Conf	Time (s)	Number of Rules	Time (s)	Number of Rules	Time (s)	Number of Rules
70	50%	28.571	178	0.996	178	65.000	178
70	60%	26.143	32	1.070	32	13.000	32
140	50%	11.571	70	0.480	70	29.900	70
140	60%	10.000	8	0.491	8	5.520	8



Bootstrap

Action rule and bootstrap distribution:

```
[(gender: Female) ∧ (SeniorCitizen: 1) ∧ (InternetService*: Fiber optic) ∧ (OnlineSecurity: No → Yes) ∧ (DeviceProtection: No → Yes) ∧ (StreamingTV: No → Yes)] ⇒ [Churn: Yes → No], uplift: 0.007930873613111296
```



Bootstrap

- Suitable for non-parametric, rule-based systems
- Simulates model uncertainty by **resampling data**
- Produces an empirical distribution of outputs
- Provides **confidence intervals** for uplift (or other metric)

Links

- https://github.com/lukassykora/action-rules
- https://github.com/lukassykora/actionrules
- https://github.com/vojtech-balek/llm-features/blob/main/notebooks/actions-M17Plus.ipynb

FACULTY OF INFORMATICS AND STATISTICS

Lukas Sykora

lukas.sykora@vse.cz

Thank you for your attention

Prague University of Economics and Business