Apriori-inspired join step for mining decision-independent rules with DeciClareMiner

Technical Report

Steven Mertens

Department of Business Informatics and Operations Management
Faculty of Economics and Business Administration
Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium
{steven.mertens}@ugent.be

Please do not refer to this technical report in your academic articles. Instead, you can refer to Mertens S., Gailly F., Poels G.: Discovering Healthcare Processes with DeciClare (2017)
See PUBLINK

The join step

Rules with a logical expression of size k are joined to create rules with a logical expression of size k+1 in the iteration steps of the algorithm. However, the elements of the logical expressions can be connected with either the inclusive disjunction or the logical conjunction operator. Therefore the influence of joining these logical expressions using either operator on the number of violations and the support level has been investigated for each template to apply the leveraged property.

The relation constraint templates have two possible logical expressions that can be joined: joining the logical expressions on the condition sides of relation constraints and joining the logical expressions on the consequence sides. **Table 1** and **Table 2** present the lower and upper bound for the number of conforming (#C) and violating (#V) occurrences of the child rule relative to the respective values for the parent rules for each type of join operation. For example, consider rules Response(AtLeast(A, 1), AtLeast(B, 1)) and Response(AtLeast(A, 1), AtLeast(C, 1)) for an event log consisting of four times trace 'AB' and two times 'AC'. The former rule has 4 confirming and 2 violating occurrences, while the latter has 2 confirming and 4 violating occurrences. **Table 2** states that the rule resulting from the OR-join operation on the consequence side, Response(AtLeast(A, 1), AtLeast(B or C, 1)), will have at least 4 (=max[4, 2]) and at most 6 (=min[4+2, 6]) confirming occurrences and at least 0 (=max[6-6, 0]) and at most 2 (=min[4, 2]) violating occurrences. Actually evaluating the rule results in 6 confirming occurrences and no violations.

A distinction is also made between the positive (i.e., 'at least …' consequence) and negative (i.e., 'at most 0' consequence) versions of the relation templates, as this allows for more specific bounds.

	'At least 1' consequence expression		***'At most 0' consequence expression***	
	OR-join	***AND-join***	***OR-join***	***AND-join***
Max #C	sum(#C)	min(#C)	sum(#C)	min(#C)
Min #C	max(#C)	0	max(#C)	0
Max #V	sum(#V)	min(#V)	sum(#V)	min(#V)
Min #V	max(#V)	0	max(#V)	0

Table 1. The bounds for join operation of the condition side of the relation templates

	'At least 1' consequence expression		*'At most 0' consequence expression*	
	OR-join	***AND-join***	***OR-join***	***AND-join***
Max #C	min(sum(#C), #Occurrences)	min(#C)	min(#C)	min(sum(#C), #Occurrences)
Min #C	max(#C)	max(sum(#C) - #Occurrences, 0)	max(sum(#C) - #Occurrences, 0)	max(#C)
Max #V	min(#V)	min(sum(#V), #Occurrences)	min(sum(#V), #Occurrences)	min(#V)
Min #V	max(sum(#V) - #Occurrences, 0)	max(#V)	max(#V)	max(sum(#V) - #Occurrences, 0)

Table 2. The bounds for join operation of the consequence side of the relation templates

Based on **Table 1** and **Table 2**, optimizations can be formulated for the join operations for each type of relation rule. At the condition side, it is clear that the OR-join will never result in a minimal rule as the number of violations of the child rule is at least equal to the highest number of violations of its parent rules. So only the AND-join operation will be used when joining condition sides. Additionally, these bounds make it possible to know beforehand if the resulting rule can have a support level higher than the threshold. The maximum support level of the child rule is the sum of the maximal number of conforming occurrences and the maximal number of violating occurrences. In this case, this is equal to the sum of the minimal number of conforming occurrences and the minimal number of violating occurrences of the parent rules. A similar conclusion can be drawn at the consequence side, however, the join operation to use now depends on whether or not the constraint in question is based on a positive or a negative relation template. Positive constraints will only use the OR-join, while negative constraints will only use the AND-join. The bounds for the consequence side do not allow for any conclusions to be drawn about the support level of the child rule.

Additionally, the iteration step will be executed twice for relation templates, because the join operations are defined for the condition and consequence side separately. The first time it will be executed with the input described in the initialization step. This will only grow the size of the condition side logical expression of the constraints. The resulting output is a set of all the minimal rules that occur at least once with a consequence side logical expression of length one. This output is then again used as input for the algorithm, but this time the consequence side logical expression is the one that is allowed to grow.

The proposed algorithm only supports two of the resource templates, namely the AtLeastUsage1 and AtMostUsage0 templates, as the others are difficult to mine from traditional event logs. For these two templates, the bounds for the number of confirming and violating occurrences of this join operation are the same as for the positive relation templates in **Table 1**.

References

1. Mertens, S., Gailly, F., Poels, G.: Enhancing declarative process models with DMN decision logic. Enterprise, Business-Process, and Information Systems Modelling. LNBIP, vol. 214. pp. 151–165. Springer (2015).
2. Pesic, M.: Constraint-based workflow management systems: shifting control to users, (2008).
3. Westergaard, M., Maggi, F.M.: Looking Into the Future about Timed Declarative Process Models. OTM 2012. pp. 250–267 (2012).
4. Jiménez-Ramírez, A., Barba, I.: Generating multi-objective optimized business process enactment plans. Advanced Information Systems Engineering. LNCS, vol. 7908. pp. 99–115. Springer (2013).
5. OMG: Decision Model and Notation (DMN1). (2014).