

Supplementary Material

Proof of Lemma 1

Proof. Consider a partial assignment x and let $j \in x^-$ s.t. $\text{cl}_{inc}(x^+, j, M')$ holds, meaning that $j \in \text{clos}_{M'}(x^+)$. As x^+ cannot be extended to a closed itemset without adding j (j being in x^-), the result follows. \square

Proof of Proposition 2

Proof. Let x be a consistent partial assignment and i be a free item.

- 1) Assume that $\text{cl}_{inc}(x^+, i, M')$ holds. It follows that $i \in \text{clos}_{M'}(x^+)$. From Lemma 1, x^+ cannot be extended to a closed itemset w.r.t. M' with $i \in x^-$. Thus $\{0\} \notin \text{dom}(x_i)$.
- 2) (**proof by contradiction**) Suppose $j \in x^-$ s.t. $\text{cl}_{inc}(x^+ \cup \{i\}, j, M')$, which means that $j \in \text{clos}_{M'}(x^+ \cup \{i\})$. Since $j \in x^-$, which means that $j \notin \text{clos}_{M'}(x^+)$, it follows that $j \notin \text{clos}_{M'}(x^+ \cup \{i\})$. \square

Algorithm 1 ADEQUATECLOSURE

In: \mathcal{D} : dataset, M' : set of measures
InOut: x : vector of Boolean variables

```

1: if  $\exists i \in x^- : \text{cl}_{inc}(x^+, i, M')$  then return false
2: for  $i \in x^*$  do
3:   if  $\text{cl}_{inc}(x^+, i, M')$  then  $\text{dom}(x_i) \leftarrow \text{dom}(x_i) \setminus \{0\}$ 
4: end for
5: for  $j \in x^-$  do
6:   for  $i \in x^*$  do
7:     if  $\text{cl}_{inc}(x^+ \cup \{i\}, j, M')$  then  $\text{dom}(x_i) \leftarrow \text{dom}(x_i) \setminus \{1\}$ 
8:   end for
9: end for
10: return true
```

Algorithm 1 shows the propagator for ADEQUATECLOSURE. It takes as input the transactional dataset \mathcal{D} , the item variables x , and the set of measures M' . It starts by checking the consistency of the current partial assignment (Lemma 1), that is, if x^+ cannot be extended to a closed itemset w.r.t. M' without adding i ($i \in x^-$) (line 1), if so, the constraint is violated and a fail is returned. Algorithm 1 must then check items $i \in x^*$ that are required to extend x^+ to a closed pattern w.r.t. M' . Thus, we remove 0 from each free item variable $i \in x^*$ such that $\text{cl}_{inc}(x^+, i, M')$ holds by using rule (\mathbf{R}_1). Second, using rule (\mathbf{R}_2), we remove 1 from each free item variable $i \in x^*$ such that $x^+ \cup \{i\}$ cannot be extended

to a closed pattern w.r.t. M' without adding the absent item $j \in x^-$ (lines 5-9).

Computing $m(x^+)$ depends on the type of m . For instance, computing $\text{sup}(x^+)$ requires a reversible BitSet to store the cover of x^+ . Other measures such as min or max only require a reversible integer.

Proposition (Consistency, time complexity). *Let \mathcal{D} be a transaction database of n items and m transactions and a set of measures M' , which contains c sup-based measures. Algorithm 1 enforces domain consistency in time $\mathcal{O}(n^2mc)$.*

Proof of the domain consistency of Algorithm 1

Proof. (**proof by contradiction**) Let i be an item, M' a set of preserving measures and $v \in \text{dom}(x_i)$ such that x_i is not domain consistent:

- if $v = 1$, then it means that x^+ can't be extended to a closed pattern w.r.t. M' . Such case is not possible since rule 2 is applied.
- if $v = 0$, then it means that x^+ can't be extended to a closed pattern w.r.t. M' without adding i . This will never happen because rule 1 is applied. \square

Proof of the time complexity of Algorithm 1

Proof. Let n be the number of items and m the number of transactions in the database. Computing $m(x^+)$ requires at most $\mathcal{O}(nm)$ for each sup-based measures (such as sup or sup_\vee) and $\mathcal{O}(n)$ for each attribute-based measures (such as min or max). Checking rule 1 can be done in $\mathcal{O}(nm)$ for sup-based measures and $\mathcal{O}(n)$ for attribute measures. Finally, checking rule 2 requires $\mathcal{O}(n^2m)$ for sup-based measures and $\mathcal{O}(n^2)$ for attribute measures. Therefore, if c is the number of sup-based measures in M' , the worst case complexity is $\mathcal{O}(n^2mc)$. \square

Proof of Proposition 3

Proof. We have $P = X \cup Y$, X is a generator. Since $Y' = Y \cup \text{clos}^-(P)$, we have $X \cup Y' = \text{clos}_{\text{sup}}(P)$. Thus, r' is an MNR. By definition, $\text{sup}(P) = \text{sup}(\text{clos}_{\text{sup}}(P))$ and thus $\text{sup}(r) = \text{sup}(r')$. As $\text{conf}(r) = \frac{\text{sup}(r)}{\text{sup}(X)}$ and $\text{conf}(r') = \frac{\text{sup}(r')}{\text{sup}(X)}$, we get $\text{conf}(r) = \text{conf}(r')$. \square

CP model for computing MNRs (solving step)

$$\text{SKY-MNR}_{\mathcal{D}}(x, y, z) \equiv \left\{ \begin{array}{ll} \forall s \in \text{Sky} : & \\ \text{sky}_s \Leftrightarrow \left(\bigwedge_{i \in s} z_i \right) \wedge \left(\bigwedge_{i \notin \text{clos}_{sup}(s)} \neg z_i \right) \wedge \left(\bigwedge y_i \right) & (1) \\ \sum_{s \in \text{Sky}} \text{sky}_s \geq 1 & (2) \\ \forall i \in \mathcal{I} : \neg x_i \vee \neg y_i & (3) \\ \sum_{i \in \mathcal{I}} x_i \times \sum_{i \in \mathcal{I}} y_i \geq 1 & (4) \\ \forall i \in \mathcal{I} : z_i \Leftrightarrow x_i \vee y_i & (5) \\ \text{GENERATOR}_{\mathcal{D}}(x) & (6) \end{array} \right.$$

(a) Variables. There are two groups of decision variables:

- *rule variables*: x, y and z are three vectors of Boolean variables, where x_i, y_i and z_i represent the presence of item $i \in \mathcal{I}$ in the antecedent x , the consequence y and in the rule as a whole z .

- *skypattern variables*: sky is a vector of Boolean variables, where sky_s holds iff the MNR is related to the skypattern s .

(b) Constraints. Our model involves the following set of constraints:

- **Cons.(1)** is a channeling constraint ensuring that the rule associated to skypattern s satisfies $X \cup Y = \text{clos}_{sup}(s)$ and $\text{clos}^-(s) \subseteq Y$.

- **Cons.(2)** associates a rule to at least one skypattern.

- **Cons.(3)** ensures that an item cannot appear in the antecedent and in the consequence of the rule simultaneously.

- **Cons.(4)** ensures that the antecedent and the consequence of the rule are not empty.

- **Cons.(5)** is a channeling constraint between z and x, y .

- **Cons.(6)** ensures that x is a generator.

Rules quality for SKY4MNR vs CP4MNR

Figures 1 and 2 plot the values of support (noted freq) vs. confidence and confidence vs. lift for both approaches for different datasets.

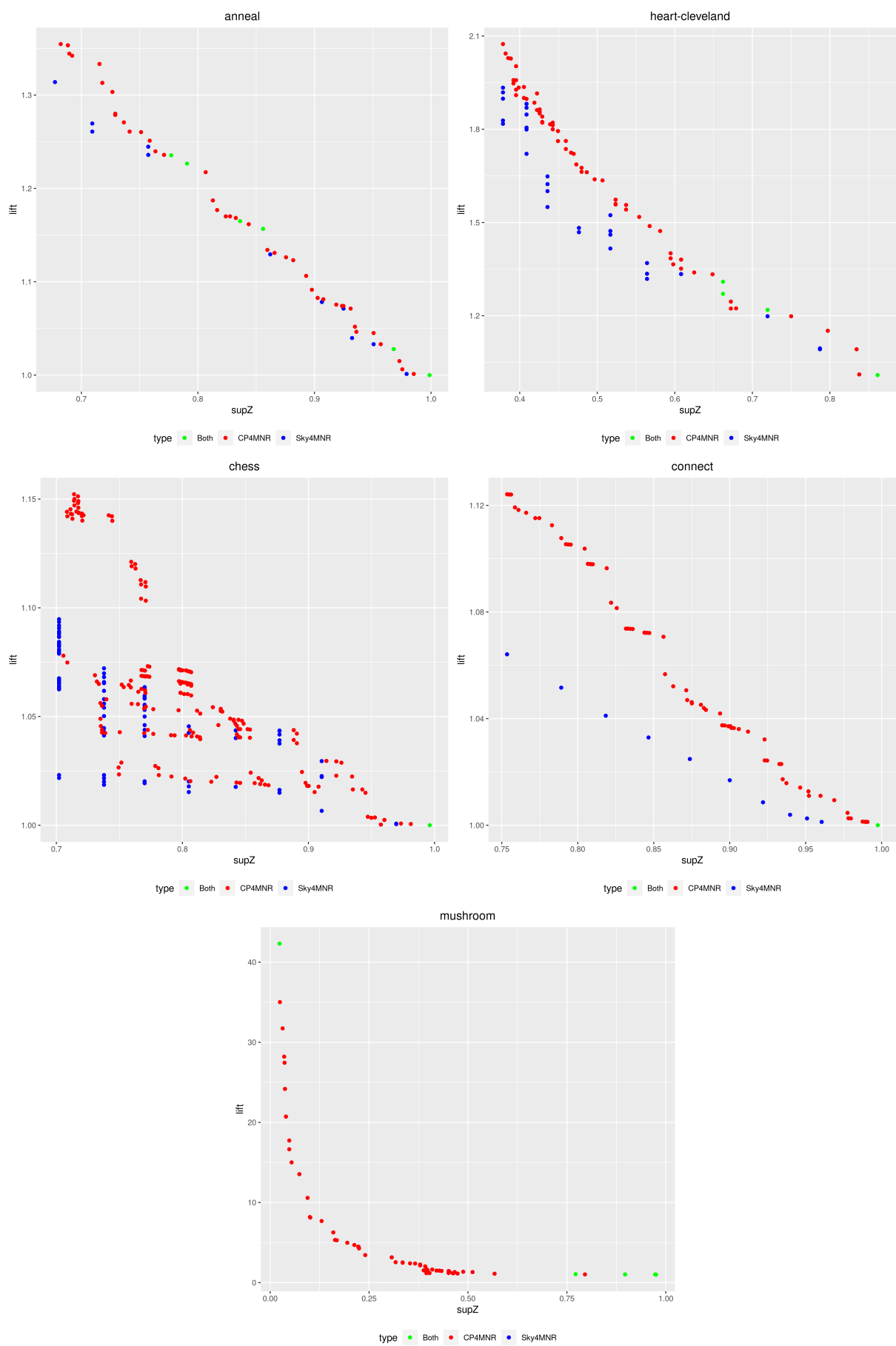


Figure 1: Qualitative analysis of MNRs (all datasets): support vs. lift.

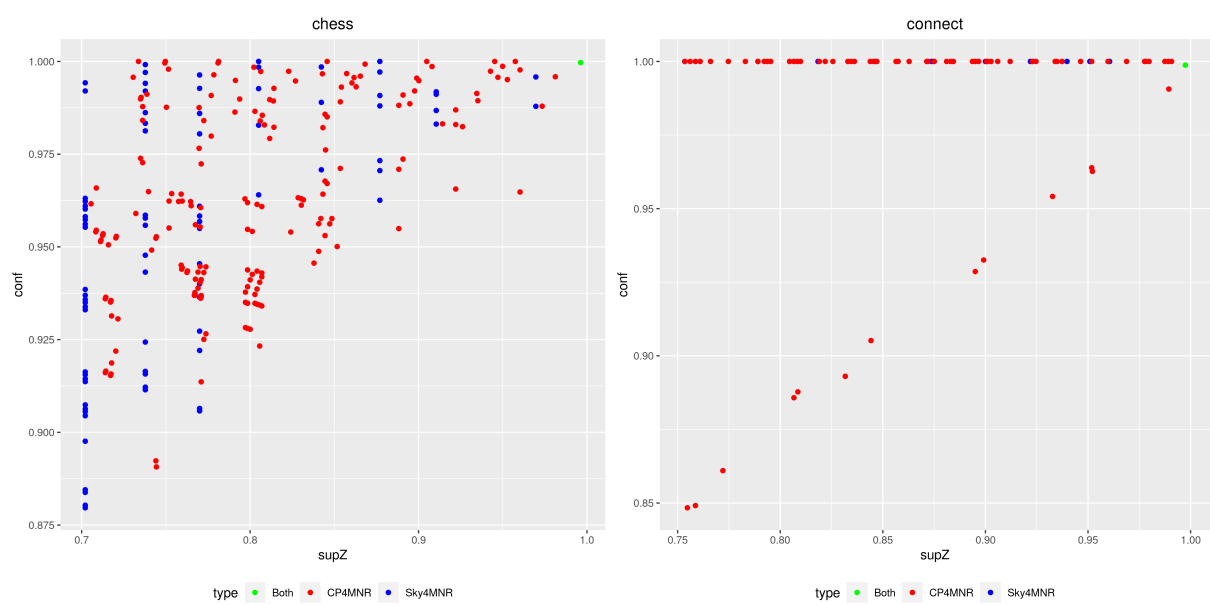


Figure 2: Qualitative analysis of MNRs (Chess and Connect datasets): support vs. confidence