Parameter-free association rule mining with *yacaree*

José L Balcázar

Departamento de Matemáticas, Estadística y Computación
Universidad de Cantabria, Santander, Spain
joseluis.balcazar@unican.es

1 Introduction

Association rule mining is one of the most central aspects of data mining. There are many implementations of association miners: Borgelt (2003); Witten and Frank (2005); Zaki and Hsiao (2005). The problems faced in association rule mining are twofold. First, the quantity of candidate itemsets potentially leading to rules grows exponentially with the often already large universe of items. The introduction of a support parameter was a key advance that allowed for the design of efficient frequent set miners and for the computation of association rules in large datasets: there, exploration is limited to those itemsets that appear "often enough" as subsets of the transactions, that is, their relative frequency exceeds a certain ratio of the transactions (Agrawal et al. (1996)). Then, the second problem is that, often, the set of rules provided as output is huge, specially if we consider that its purpose is to be read by a human.

Many studies of notions of redundancy exist that limit the output to "irredundant" rules, according to several existing notions of redundancy (e.g. Zaki (2004); Kryszkiewicz (2001)); but even taking redundancies into account, the results are, in many cases, unsatisfactory: high implication thresholds leave out many interesting rules, whereas lower ones let pass far too many rules to be manually inspected. Many alternative quality measures exist for association rules: Geng and Hamilton (2006); Lenca et al. (2008); Tan et al. (2004).

Our system *yacaree* (Yet Another Closure-based Association Rule Experimentation Environment) processes transactional datasets, each transaction being an itemset, and obtains irredundant rules $X \to Y$, where both X and Y can be arbitrary disjoint itemsets; yet, it does not require the user to select the value of any threshold parameter. As in most current proposals, we mine only frequent closed itemsets, and apply known redundancy filters. Our current closure mining algorithm is a simplified variant of ChARM (see Zaki and Hsiao (2005)), rather close to a depth-first search. As in some of the associators of Weka, we mine closures in order of decreasing support; see Witten and Frank (2005). However, our algorithm is very different: it requires no "delta" parameter for stepwise support threshold decrease as Weka's "Apriori" does, and does not relate support to confidence as "predictive Apriori" does (Scheffer (2005), also present in Weka). Instead, we self-adjust the internal effective support bound on the basis of technological limitations: it starts at an almost trivial level, and grows, if necessary, as the monitorization of the mining process reveals that the memory consumption surpasses internal thresholds. The closures are passed on to a "border" algorithm which computes the lattice structure, and then irredundant rules are extracted.