1. **Answer:**

Aumann and Lindell have pen -pictured the ways and means of information retrieval on how a certain quantitative rule can be discovered systematically and efficiently in large data sets with quantitative attributes in their article ‘A Statistical Theory for Quantitative Association Rules’. The database considered in the paper is called *Determinants of Wages from the 1985 Current Population Survey in the United States* (the database may be found at http://lib.stat.cmu.edu/datasets). The database contains 534 transactions and 11 attributes (7 categorical and 4 quantitative).

Aumann and Lindell designed an efficient algorithm for finding quantitative association rules **based on two types of rules:**

1. Rules X→ (Tx) where both X and J contain a single quantitative attribute only.
2. Rules X →(Tx) where X (only categorical attributes) and J (only quantitative attributes). There is no limit on the number of attributes in X or J. The algorithm is correct for any measure M.

5Algorithm Motivation

Let i and j be a pair of quantitative attributes. If we sort the database by attribute i, then any above or below average continuous region of values in j is a rule (provided it passes the necessary statistical test). This is because attribute i is sorted, and therefore any continuous region is a range. However, we must also ensure that the rule is irreducible and maximal. The algorithm is based on the following simple idea: if the regions [a, b] and [b, c] are both above or below average, then so too is the region [a, c]. It is completely symmetrical to search for above or below average regions, therefore Aumann and Lindell refer only to above average. “Above average” mean above the overall mean plus *mindif.*

The Window Procedure.

The following data-driven procedure, called “Window”, accepts as input an array of values and the average of the values in the array (*plus mindif*). The input array is the array of values of attribute j, sorted by i. Then Aumann and Lindell execute a single pass to find all rules from i to j. The procedure works with two windows or regions: A and B. A is an irreducible above average region (this remains invariant throughout). B is an adjacent region that may be joined to A if A B will also be irreducible.

To begin, Aumann and Lindell initialize A to the first above-average value in j (this is clearly irreducible) and B is empty.

Given A and B, Aumann and Lindell added the next value in j to B. There are three possibilities at this stage:

(1.) The average of the B region is above-average: Aumann and Lindell join B to A (emptying B). A is still irreducible: separating AU B into A and B or into and B obviously leaves two above average regions. Separating A U B into two halves AU and also leaves two above-average regions because if A U is not above-average, step (2) below would already have happened, and if B2 is not above-average then BI must be above average and this step would have happened earlier.

The average of the region including A and B together is not above-average: The A region is a potential rule (and not the B region). Aumann and Lindell continue by emptying B and initializing A to be the first above-average value after the region of the potential rule. No rule can contain AUB at the beginning, as the rule would then be reducible. Furthermore, A certainly has maximal support by definition for if expand A, either the region will not be above-average or it will be reducible.

If neither of the above is true, then Aumann and Lindell simply continue by adding the next value in j to B.

Window is shown in figure below. Aumann and Lindell ran Window twice in order to find both above and below average rules. Upon finding a potential rule, Aumann and Lindell execute a Ztest to determine whether or not to accept the rule or not. If yes, Window is call recursively with the input array as the j values supporting the accepted rule. The input average is the average of the rule (plus/minus mindif appropriately). This recursive call finds all sub-rules of the rule and so on. If the rule is not accepted, Aumann and Lindell simply continue searching for rules in the following regions.

# Complexity Analysis

For a given pair of attributes, the time taken for n transactions is O(nlogn) for the sort, plus the complexity of the Window algorithm. The complexity of Window is clearly upper-bound by O(n) times the number of levels of rules (i.e. the number of recursive calls). Since the number of levels is expected to be low (as experience has shown), Aumann and Lindell effectively maintain linear complexity. Note that minimum support has no effect on the running time, enabling them to find rules with very low support. For k quantitative attributes, the time taken to find all rules of this type is therefore O (k . n log n +. n).

Window Procedure for finding “Numerical → Numerical” Rules

|  |
| --- |
| Input: an array Array and a number a = average + mindif  **Window** (*Array, a*)  current ← index of the beginning of Array  While (current < end-of-Away)  {  current ← next above average value Initialize parameters for A and B regions  While (AVERAGE(A, B)≥ a) // Weighted average  {  Update B to include Array[current] current ← current + 1 if (AVG(B) > a)  Join A to B and empty B }  If the values in region A pass a Z-test:  Add the appropriate rule to the set of results  Call Window(Array[A region], AVG(A) ± mindif) current ← first index after the A region } |

# Finding rules from Categorical to Numerical attributes

In this section, Aumann and Lindell wrote an algorithm as below Algorithm Outline. The algorithm has three distinct stages:

1. Find all frequent sets of categorical items only, using known algorithms such as Apriori (see [2]).
2. For all quantitative attributes, calculate the distribution measure (mean/variance) for each frequent set, using the hash-tree data structure. One pass over the database is sufficient.
3. Find all non-contained rules and sub-rules. For every frequent set X and quantitative attribute e, it remains to check if X →, and X → are basic rules or sub-rules or neither. Aumann and Lindell do this by traversing a lattice of the frequent sets while keeping track of containment relations between sets and the sub-rule hierarchy. We note that the ideas in sections 3.1 and 3.2 may be combined in order to find rules with profiles containing many categorical and a single numerical attribute. For a given frequent set X, we run Window on TX. Aumann and Lindell may run Window in parallel on each frequent set and efficiently achieve the desired result.

**Apriori and Apriori Tid** is for discovering all significant association rules between items in a large database of transactions were extensively discussed in the paper ‘Fast Algorithms for Mining Association Rules’ by Rakesh Agrawal and Ramakrishnan Srikant\*. I used these algorithms in solving assignment 4 problems.

**2. Solution:**

Given that,

The price of each item is nonnegative.

(1). Containing items whose sum of the prices is less than $150.

Here we are interested to find a set of items X such all items x€X

This is anti-monotonic constraint, as here

X,

, as prices are nonnegative.

(2). This is monotonic constraint, as if an itemset satisfies this constraint, any of their superset also satisfies (prices being non-negative), for example

If for set X: 200

Then X, 200

(3). Where the average item for all item is between $100 and $500,

Here the average price is neither monotonic nor anti-monotonic.

It is convertible constraint and can be converted by ordering the items.