Sampling :

Mining on a subset of the given data.

The basic idea of sampling approach is to pick a random sample S of the given data, and then search for frequent itemsets in S instead of D. In this way, we trade off some degree of accuracy against efficiency.

The sample size of S is such that the search for frequent itemsets in S can be done in main memory and so only one scan of the transactions in S is required overall.

Because we are searching for frequent itemsets in S rather than D, it is possible that we will miss some of the global frequent itemsets.

To, decide whether any frequent itemsets have been missed, the concept of negative border is used.

The negative border with respect to a frequent itemset S and set of items I, is the minimal itemsets contained in powerset (I) & not in S. The basic idea is that negative border of a set of frequent itemsets contains the closet itemsets that could also be frequent

for eg:
Consider the set of items $I = \{A, B, C, D, E\}$ &
let combined frequent itemsets of size 1 to 3 be

$S = \{\{A\}, \{B\}, \{C\}, \{D\}, \{AB\}, \{AC\}, \{BC\}, \{AD\}, \{CD\}, \{ABC\}\}$.

The negative border is $\{\{E\}, \{BD\}, \{ACD\}\}$.
The set $\{E\}$ is the only 1-itemset not contained in S, $\{BD\}$ is the only 2-itemset not in S but whose 1-itemsubsets are and $\{ACD\}$ is the only 3-itemset whose 2-itemsets subset are all in S

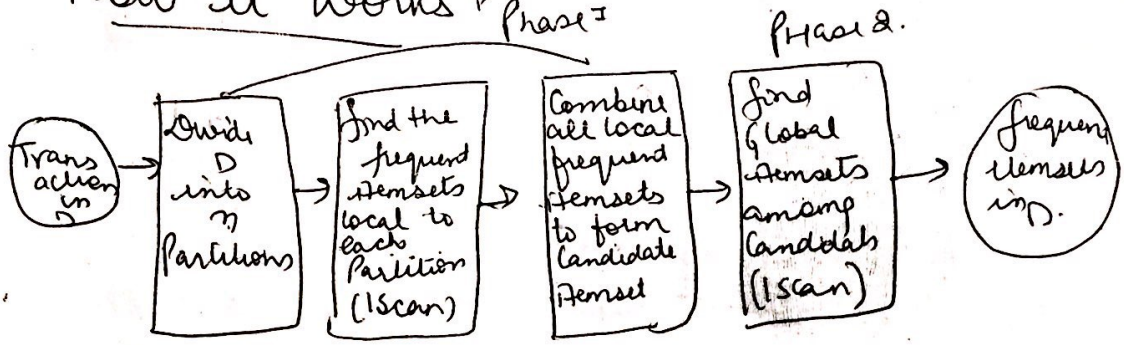
The negative border is important since it is necessary to determine the support for those itemsets in the negative border to ensure that no large itemsets are missed from analyzing the sample data.

If we find that an itemset X is the negative border belongs to the set of all frequent itemsets, then there is potential for a superset of X to also be frequent. If this happens, then a second pass over the database is needed to make sure that all frequent itemsets are found.

Partitioning Algorithm

If we are given a database with a small no of potential large itemsets, say a thousand then the support for all of them can be tested in one scan by using a partitioning technique.

How it works:



At the end of pass one we take the union of all frequent itemsets from each partition. This forms the global candidate frequent itemsets for the entire database.

When these lists are merged, they may contain some FALSE POSITIVES. That is, some of itemsets that are frequent in one partition may not qualify in several other partitions & hence not exceed the minimum support when the original database is considered.