COL 761

DATA MINING

Assignment – 1
Transactional Data Compression

TEAM MEMBERS:

1. Banoth Chethan Naik	-	2020CS10333
2. Perugu Krishna Chaitanya Yadav	-	2020EE10521
3. Devarakonda Ronith Kumar	-	2020EE10486

Transactional Data Compression

Imagine you have a set of data sequences, and you want to find recurring patterns while making the data more compact. This is where the FP-Growth algorithm comes into play.

Creating the FP-Tree:

In this initial step, the algorithm constructs a specialized tree structure known as the FP-Tree from the input data sequences. This tree helps identify frequent itemsets, which are combinations of items that appear together often in the sequences.

Discovering Frequent Itemsets:

From the FP-Tree, the algorithm extracts the frequent itemsets – these are sets of items that occur above a specified threshold frequency. These itemsets represent recurring patterns within the data.

Symbolic Substitution:

Rather than retaining the actual frequent itemsets, the algorithm substitutes them with unique symbols or codes. These symbols serve as placeholders for the frequent itemsets in the compressed data representation.We replaced with negative integers.

Generating the Compressed Data:

By using the symbol substitutions, the algorithm creates a compressed version of the original data. Each sequence is now represented using the generated symbols, which effectively reduces the size of the data.

Creating a Symbol-Key Mapping:

To ensure the reversibility of the process, the algorithm generates a mapping between the symbols and the original frequent itemsets. This mapping is stored at the end of the compressed data as a reference.

Decompressing the Data:

Mapping Symbols to Itemsets:

The decompression process begins by interpreting the compressed data. The symbol-key mapping is used to understand the relationships between the symbols and the original frequent itemsets.

Restoring Original Sequences:

As the algorithm traverses the compressed data, it replaces the symbols with the corresponding frequent itemsets using the mapping. This step effectively restores the original data sequences.

In summary, the FP-Growth algorithm facilitates data compression by substituting frequent itemsets with symbols, leading to a more compact representation. With the help of a symbol-key mapping, the original data can be efficiently reconstructed by reversing the symbol substitutions.