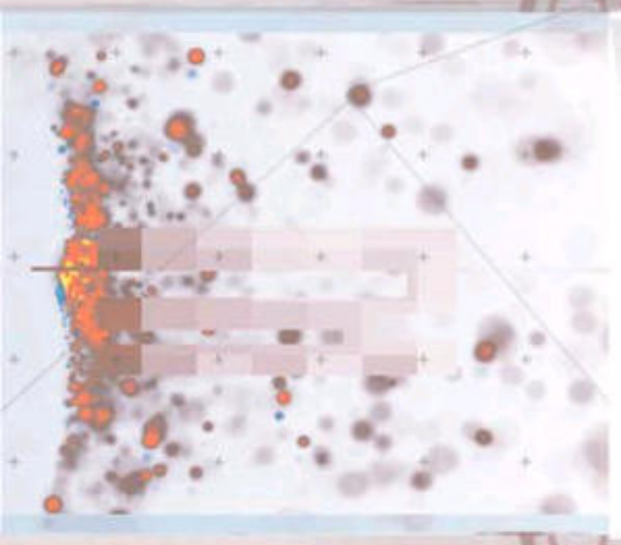

Compressed Representation: Closed Patterns and Max-Patterns

Challenge: There Are Too Many Frequent Patterns!

- A long pattern contains a combinatorial number of sub-patterns
- How many frequent itemsets does the following TDB_1 contain?
  - $TDB_{1:}$ T_1: $\{a_1, ..., a_{50}\}$; T_2: $\{a_1, ..., a_{100}\}$
  - Assuming (absolute) *minsup* = 1
  - Let's have a try

    1-itemsets: $\{a_1\}$: 2, $\{a_2\}$: 2, ..., $\{a_{50}\}$: 2, $\{a_{51}\}$: 1, ..., $\{a_{100}\}$: 1,

    2-itemsets: $\{a_1, a_2\}$: 2, ..., $\{a_1, a_{50}\}$: 2, $\{a_1, a_{51}\}$: 1 ..., ..., $\{a_{99}, a_{100}\}$: 1,

    ..., ..., ..., ...

    99-itemsets: $\{a_1, a_2, ..., a_{99}\}$: 1, ..., $\{a_2, a_3, ..., a_{100}\}$: 1

    100-itemset: $\{a_1, a_2, ..., a_{100}\}$: 1
  - In total: $\binom{100}{1} + \binom{100}{2} + ... + \binom{100}{100} = 2^{100} - 1$ sub-patterns!

A too huge set for any computer to compute or store!

Expressing Patterns in Compressed Form: Closed Patterns

- How to handle such a challenge?
- Solution 1: **Closed patterns**: A pattern (itemset) X is closed if X is *frequent*, and there exists *no super-pattern* Y ⊃ X, *with the same support* as X
  - Let Transaction DB TDB_1: T_1: $\{a_1, ..., a_{50}\}$; T_2: $\{a_1, ..., a_{100}\}$
  - Suppose *minsup* = 1. How many closed patterns does TDB_1 contain?
    - Two: P_1: "$\{a_1, ..., a_{50}\}$: 2"; P_2: "$\{a_1, ..., a_{100}\}$: 1" ← 所有的都被包含了.
- Closed pattern is a lossless compression of frequent patterns
  - Reduces the # of patterns but does not lose the support information!
  - You will still be able to say: "$\{a_2, ..., a_{40}\}$: 2", "$\{a_5, a_{51}\}$: 1"

比如存在$\{a_1 ... a_{49}\}$一定为2
存在$\{a_1 ... a_{51}\}$一定为1

Expressing Patterns in Compressed Form: Max-Patterns

- Solution 2: **Max-patterns**: A pattern X is a max-pattern if X is frequent and there exists no frequent super-pattern Y ⊃ X
- Difference from close-patterns?
  - Do not care the real support of the sub-patterns of a max-pattern
  - Let Transaction DB TDB_1: T_1: $\{a_1, ..., a_{50}\}$; T_2: $\{a_1, ..., a_{100}\}$
  - Suppose *minsup* = 1. How many max-patterns does TDB_1 contain?
    - One: P: "$\{a_1, ..., a_{100}\}$: 1"
- Max-pattern is a lossy compression!
  - We only know $\{a_1, ..., a_{40}\}$ is frequent
  - But we do not know the real support of $\{a_1, ..., a_{40}\}$, ..., any more!
- Thus in many applications, mining close-patterns is more desirable than mining max-patterns

Recommended Readings

- R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", in Proc. of SIGMOD'93
- R. J. Bayardo, "Efficiently mining long patterns from databases", in Proc. of SIGMOD'98
- N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Discovering frequent closed itemsets for association rules", in Proc. of ICDT'99
- J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent Pattern Mining: Current Status and Future Directions", Data Mining and Knowledge Discovery, 15(1): 55-86, 2007

close patterns：关注子 support，需要 support 判断.

max-patterns：不需要关注子 support，可以不判.