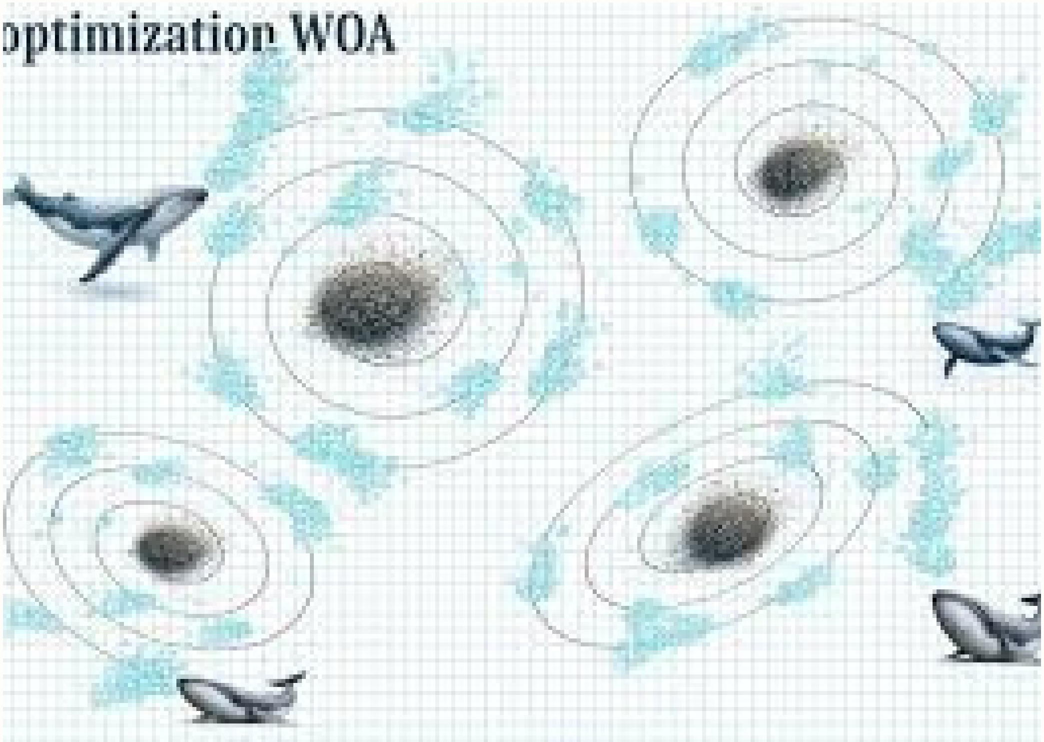


FIND FRQUENT ITEMSET IN A GIVEN DATASET USING WOA AND APRIORI ALGORITHM

presented by :

HARBANE AMEL

LAIDI ZAKARIA

Summary

Abstract

Frequent itemset mining is a data mining technique. It is commonly applied to transactional databases, where each transaction contains of a set of items. The goal of this method consists of identifying subsets of items that frequently co-occur in these transactions. This task is considered as NP-Hard problem. In fact, exact Algorithms of this method such as Apriori Algorithm have salability issues . to find the most frequent itemset (s) in a given dataset and for a given Minsup, using inspired version of the Whale Optimization Algorithm (WOA) metaheuristic.

Key words:

Frequent itemset , transactional databases, co-occur, NP-Hard problem, Apriori Algorithm , Minsup, Whale Optimization Algorithm , meta heuristic.

Introduction

Meta-heuristic optimization algorithms are becoming more and more popular in engineering applications because they: (i) rely on rather simple concepts and are easy to implement; (ii)don’t require gradient information; (iii) can bypass local optima; (iv) can be utilized in a wide range of problems covering different disciplines.

Problematic

The task is to find the most frequent itemset (s) in a given dataset and for a given Minsup using WOA meta heuristic so we have to reformulate it as an optimization problem (Solution representation, Fitness function and constraints definition)

Whale Optimisation Algorithm inspiration

WOA is a novel nature-inspired meta-heuristic optimization algorithm,which mimics the social behavior of humpback whales. The algorithm is in-spired by the bubble-net hunting strategy.

Quantum-inspired metaheuristics

Quantum-inspired metaheuristics emerged by combining the quantum mechanics principles with the metaheuristic algorithms concepts. These algorithms extend the diversity of the population, which is a primary key to proper global search and is guaranteed using the quantum bits’ probabilistic representation
The source of inspiration for most quantum-inspired metaheuristics are the Genetic and Evolutionary algorithms, followed by swarm-based algorithms, and applications range from image processing to computer networks and even multidisciplinary fields such as flight control and structural design. The promising results of quantum-inspired metaheuristics give hope that more conventional algorithms can be combined with quantum mechanics principles in the future to tackle optimization problems in numerous disciplines

Requested work

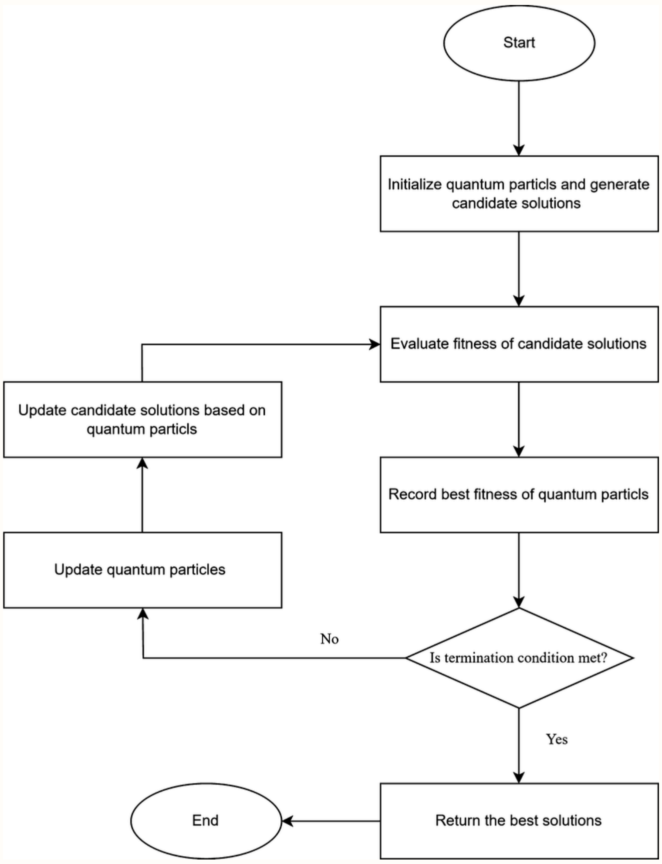
1. Formulate the problem as an optimization problem

- Each solution is represented as a binary vector where: 1 indicates the inclusion of an itemset. 0 indicates the exclusion of an itemset. Example: A binary vector [1, 0, 1, 1] corresponds to including the 1st, 3rd, and 4th itemsets.
- $f(x) = \sum_{i \in S(x)} \text{Support}(i)$
- Constraint : Support(i) > minsup

2.Choosing a quantum inspired version of the WOA metaheuristic

- The chosen Quantum-Inspired Whale Optimization Algorithm (QWOA) integrates quantum principles, enhancing solution search capabilities through:Quantum Superposition: Represents multiple solutions simultaneously.Quantum Rotation Gates: Probabilistically updates solution states to balance exploration and exploitation.Measurement Operation: Collapses quantum states into binary solutions for fitness evaluation.

3.The Quantum inspired WOA metaheuristic to solve this problem



Pseudo Algorithm for Whale Optimization Algorithm (WOA) for Frequent Itemset Mining

```
Initialize whale population (binary vectors)
Initialize best position and best fitness

For each iteration t = 1 to MaxT:
  For each whale i in the population:
    Evaluate the fitness (support) of the current itemset
    Update the best position if the current itemset has higher support
  End for

  For each whale i in the population:
    Update whale position based on best position using spiral or shrinking circle mechanism
  End for

  If best fitness > min_support:
    Stop algorithm
  End if
End for
```

Return the best itemset found with support greater than or equal to min_support

Pseudo Algorithm for Apriori Algorithm for Frequent Itemset Mining

Input
- D: Transaction database
- min_sup: Minimum support threshold

Output
- L: Frequent itemsets that satisfy min_sup

Variables
- Ck: Candidate itemset of size k
- Lk: Frequent itemset of size k
- k: Itemset size counter

Main Algorithm

```
1. L1 = {find frequent 1-itemsets}
2. for (k = 2; Lk-1 ≠ ∅; k++) do
3.    Ck = apriori_gen(Lk-1)    // Generate candidates
4.    for each transaction t ∈ D do
5.       Ct = subset(Ck, t)     // Candidates contained in t
6.       for each candidate c ∈ Ct do
7.          c.count++
8.    end for
9.    Lk = {c ∈ Ck | c.count ≥ min_sup}
10. end for
11. return L = ∪k Lk
```

Apriori_gen Function (Candidate Generation)

Function apriori_gen(Lk-1)
1. Insert into Ck
2. Select p.item1, p.item2, ..., p.itemk-1, q.itemk-1
3. From Lk-1 p, Lk-1 q
4. Where p.item1 = q.item1, ..., p.itemk-2 = q.itemk-2, p.itemk-

4.Implementing the proposed solutions using python.

5.Adjust the parameters of the proposed solution to obtain the best solution comparing to theresults generated by Apriori Algorithm as base line.

For the comparison between Apriori and WOA algorithms, you can focus on these key points:

1. Performance Metrics

- Execution time differences
- Memory usage comparison
- Pattern quality (support and size of itemsets)
- Scalability with dataset size

2. Algorithmic Characteristics

- Apriori:
  * Complete (finds all patterns)
  * Deterministic results
  * Higher memory requirements
  * Bottom-up approach

- WOA:

  * Heuristic approach
  * Stochastic results
  * Lower memory requirements
  * Population-based search

3. Practical Considerations:

- Apriori works better for:
  * Small to medium datasets
  * When completeness is required
  * Dense datasets
  * When exact results are needed

- WOA works better for:

  * Large datasets
  * When approximate solutions are acceptable
  * When quick results are needed
  * Sparse datasets

4. Implementation Complexity

- Parameter tuning requirements
- Code maintenance
- Ease of modification

Conclusion

This project allowed for the comparison of the effectiveness of two approaches for discovering frequent itemsets. The Apriori algorithm remains efficient for small and well-structured datasets, but the WOA algorithm, although more complex, could offer advantages in broader and more diverse contexts where exploration of the search space is crucial.

References

Hakemi, S., Houshmand, M., KheirKhah, E. et al. A review of recent advances in quantum-inspired metaheuristics. Evol. Intel. 17, 627–642 (2024). https://doi.org/10.1007/s12065-022-00783-2