Frequent Itemset Mining using QUBO

Jonas Nüßlein

LMU Munich jonas.nuesslein@ifi.lmu.de

Abstract. In this paper we propose a R-step approximation to solve frequent itemset mining on quantum hardware like quantum annealing or QAOA. The idea is to search for the set of items where the minimal 2-item frequency is maximal. This can be represented as a maximum clique problem.

Keywords: Frequent Itemset Mining · QUBO · Quantum Annealing · Ising

1 Introduction

Frequent Itemset Mining (FIM) is a common problem in data mining [4]. The goal of FIM is to find a set of objects that frequently occur together. An example is the analysis of customer behavior [14]: let a set of transactions be given, where a transaction is a multi-set of objects (the objects bought by one person). The goal is then to find a set of objects of size $k$, which were frequently bought together.

It has been shown that FIM is NP-hard [11] and therefore difficult to solve on classical computers. In this paper we present a simple method of formulating FIM as a Quadratic Unconstrained Binary Optimization (QUBO) [2] which can be solved on quantum hardware for example using Quantum Annealing or QAOA [16].

2 Background

2.1 Quadratic Unconstrained Binary Optimization

Given a symmetric $(n \times n)$-matrix $Q$ and a binary vector $x$ of length $n$, a QUBO [2] is a function of the form:

$$H(x, Q) = \sum_{i=1}^{n} \sum_{j=i}^{n} x_i \ x_j \ Q_{ij}$$

(1)

The optimization task is to find a binary vector $x$ which is as close to the optimum $x^* = \text{argmin}_x H(x, Q)$ as possible. This we want to delegate to the machine. Our task, on the other hand, is to specify a function which maps a FIM problem instance $P$ (a database of transactions) to a QUBO matrix $Q$ in such a way that the solution $p$ (i.e. the frequent itemsets) for the problem instance $P$ can be derived from the solution vector $x^*$. 
2.2 Maximum Clique

Maximum clique is the problem of finding the biggest set $C \subseteq V$ of vertices in a graph $G = (V, E)$ such that all vertices in $C$ are pairwise connected. This problem is known to be NP-complete [1].

3 FIM as QUBO

We are looking for a frequent itemset of size $N$. The general idea is as follows: we model the database as a graph, where the vertices represent the objects and the edge weights $w_{ij}$ are the empirical probabilities that the two connected vertices $i$ and $j$ are in one transaction. We then conduct a R-step approximation: we set the threshold to $\tau = 0.5$ then we remove all edges from a graph $G$ with edge weight lower than $\tau$. For the resulting graph (which is now not necessarily fully connected) we calculate the maximum clique. The clique does not necessarily has size $N$. The next step is to adjust the threshold $\tau$ and repeat the process. If the size of the found maximum clique is lower then $N$ then we reduce $\tau$, if the clique size was larger then $N$ then we increase $\tau$. So the optimization problem we are solving can be formulated as:

$$\max_{I} \min_{i,j \in I} P(i \cup j)$$  \hspace{1cm} (2)

$I$ is the frequent itemset we are searching and $P(i \cup j)$ is the empirical probability of objects $i$ and $j$ occurring in the same transaction. The whole algorithm is presented in Algorithm 1.

4 Experiments

As a little proof of concept experiment we created 24000 completely random transactions of size 22 out of a basic quantity of 250 items. Additionally, we added the following transaction 1000-times to the database $[1,2,3,\ldots,20]$. So, in total the database consisted of 25,000 transactions. We then computed the most frequent 20-item set (which was clearly $[1,2,3,\ldots,20]$).

MFIO returned the correct solution in 34 seconds. For comparison we computed the most frequent 20-set also with Bomo. Bomo is based on FP-Growth [12], but it only searches for the top-N (in our case the top-1) k-itemsets. So, it doesn’t need a threshold [13]. We used a fast C implementation of Bomo [15]. Bomo returned the correct result after 102s. Also, the memory usage of MFIO was significantly more efficient (measured was the memory needed for the internal data structures):

- MFIO: 122KB (for the 31.125 2-sets)
- Bomo: 83MB
Algorithm 1 FIM as QUBO

Require:
1: \( F_{ij} \) \( \triangleright \) Empirical probability of items \( i \) and \( j \) occurring in the same transaction
2: \( K \) \( \triangleright \) Number of objects
3: \( \tau = 0.5 \) \( \triangleright \) Threshold
4: \( \text{solution} = [] \)
5: \( N \)
6: \( R \) \( \triangleright \) number of iterations
7: for \( i = 1 \) to \( R \) do
8: \( Q = [] \)
9: for \( q_1 = 0 \) to \( K - 1 \) do
10: for \( q_2 = 0 \) to \( K - 1 \) do
11: if \( q_1 == q_2 \) then
12: \( Q[q_1][q_2] = -1 \)
13: else if \( F_{q_1 q_2} < \tau \) then
14: \( Q[q_1][q_2] = K + 1 \)
15: end if
16: end for
17: end for
18: \( \text{answer} = \text{solve}(Q) \) \( \triangleright \) Solve QUBO with quantum hardware
19: if size of clique in \( \text{answer} \) \( \geq N \) then
20: \( \text{solution} = \text{getSelectedVertices}(\text{answer}) \)
21: \( \tau = \tau + 2^{-(i+1)} \)
22: else
23: \( \tau = \tau - 2^{-(i+1)} \)
24: end if
25: end for

As a little real-world example we were able to find the most frequent 4-, 5-, 6- and 7-set letters in English words. So, the basic quantity was the alphabet and the words (subsets of the alphabet) were the transactions. The database consisted of 370,000 English words [6]. The computed sets were: \{a, e, i, n\}, \{a, e, i, n, t\}, \{a, e, i, n, o, s, t\}. All experiments were conducted on D-Wave 2000Q.

5 Related Work

FP-Growth [12] is the most common approach for frequent itemset mining. Bomo [15] is a variant of FPGrowth which only searches for k-itemsets. The most related work to this paper is [5]. The authors already propose to use maximum clique to find frequent itemsets. However they use a user-defined threshold. We extend this work by proposing a R-step approximation and solving the maximum clique problems on quantum hardware.
In [7] a quantum algorithm for finding association rules is presented. In contrast to that we are searching for the most frequent itemset of length $N$ with QUBO.

In [9] and [10] QUBO and Ising formulations for maximum clique are presented. Our approach builds on these formulations.

References

1. Karp, R.M., 1972. Reducibility among combinatorial problems. In Complexity of computer computations (pp. 85-103). Springer, Boston, MA.
2. Lewis, M. and Glover, F., 2017. Quadratic unconstrained binary optimization problem preprocessing: Theory and empirical analysis. Networks, 70(2), pp.79-97.
3. Nüüslein, J., Roch, C., Gabor, T., Linnhoff-Popien, C. and Feld, S., 2022. Black Box Optimization Using QUBO and the Cross Entropy Method. arXiv preprint arXiv:2206.12510.
4. Grahne, G. and Zhu, J., 2005. Fast algorithms for frequent itemset mining using fp-trees. IEEE transactions on knowledge and data engineering, 17(10), pp.1347-1362.
5. Nadi, F., Hormozi, S.G., Foroozandeh, A. and Shahraki, M.H.N., 2014, October. A new method for mining maximal frequent itemsets based on graph theory. In 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 183-188). IEEE.
6. https://github.com/dwyl/english-words/blob/master/words_alpha.txt
7. Yu, C.H., Gao, F., Wang, Q.L. and Wen, Q.Y., 2016. Quantum algorithm for association rules mining. Physical Review A, 94(4), p.042311.
8. Nüüslein, J., Gabor, T., Linnhoff-Popien, C. and Feld, S., 2022. Algorithmic QUBO Formulations for k-SAT and Hamiltonian Cycles. arXiv preprint [arXiv:2204.13539]
9. Chapuis, G., Djidjev, H., Hahn, G. and Rizk, G., 2019. Finding maximum cliques on the d-wave quantum annealer. Journal of Signal Processing Systems, 91(3), pp.363-377.
10. Lucas, A., 2014. Ising formulations of many NP problems. Frontiers in physics, p.5.
11. Yang, G., 2004, August. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 344-353).
12. Borgelt, C., 2005, August. An Implementation of the FP-growth Algorithm. In Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations (pp. 1-5).
13. Cheung, Y.L. and Fu, A.W.C., 2004. Mining frequent itemsets without support threshold: with and without item constraints. IEEE Transactions on Knowledge and Data Engineering, 16(9), pp.1052-1069.
14. Nüüslein, J., Illium, S., Müller, R., Gabor, T. and Linnhoff-Popien, C., 2022. Case-Based Inverse Reinforcement Learning Using Temporal Coherence. In International Conference on Case-Based Reasoning (pp. 304-317). Springer, Cham.
15. https://apprsv.cse.cuhk.edu.hk/kdl/program.html
16. Farhi, E., Goldstone, J. and Gutmann, S., 2014. A quantum approximate optimization algorithm. arXiv preprint [arXiv:1411.4028]