Feature Selection via Quantum Annealers for Ranking and Classification Tasks*

Discussion Paper

Maurizio Ferrari Dacrema[1], Fabio Moroni[1], Riccardo Nembrini[1,2], Nicola Ferro[3], Guglielmo Faggioli[3] and Paolo Cremonesi[1]

[1]Politecnico di Milano, Milano, Italy
[2]ContentWise, Milano, Italy
[3]University of Padova, Padova, Italy

Abstract
Feature selection is a common step in many ranking, classification, or prediction tasks and serves many purposes. By removing redundant or noisy features, the accuracy of ranking or classification can be improved and the computational cost of the subsequent learning steps can be reduced. However, feature selection can be itself a computationally expensive process. While for decades confined to theoretical algorithmic papers, quantum computing is now becoming a viable tool to tackle realistic problems, in particular special-purpose solvers based on the Quantum Annealing paradigm. This paper aims to explore the feasibility of using currently available quantum computing architectures to solve some quadratic feature selection algorithms for both ranking and classification. Our experimental analysis shows that the effectiveness obtained with quantum computing hardware is comparable to that of classical solvers, indicating that quantum computers are now reliable enough to tackle interesting problems.

1. Introduction

Information Retrieval (IR) is concerned with delivering relevant information to people, according to their information needs, context, and profile, in the most effective and efficient way possible. Central to this goal are *ranking* and *classification*, often exploited in conjunction. Machine learning approaches have been widely investigated for this purpose. These methods however suffer from the known feature selection problem. As the data becomes more rich and complex, identifying the relevant features may require to evaluate an exponentially increasing number of cases which rapidly becomes prohibitively resource intensive. The feature selection problem is mitigated by deep learning and, more generally, neural approaches that have gained popularity in recent years. Despite these methods being extremely versatile and generally able to provide good overall effectiveness, it is known their performance is not always stable and may vary a lot across topics, for example the performance may improve for half of the topics while degrade for the other half [2]. A further disadvantage is that these neural approaches are very demanding

* The full paper has been originally presented at SIGIR 2022 [1]

IIR 2021 – 12th Italian Information Retrieval Workshop, June 29th – July 1st, 2022, Milano, Italy

in terms of computing resources and require enormous amounts of data which leads to larger and larger models that are not free from risks, as pointed out by Bender et al. [3]. In this paper we take a step back and wonder ourselves if it is possible to make the feature selection problem more “affordable” in order to make more appealing the use of “traditional” machine learning approaches for ranking and classification. To this end, we investigate the feasibility of and how to apply current generation quantum computing technologies to improve feature selection. To the best of our knowledge very little work has been done to asses the effectiveness and efficiency of such technologies to tackle feature selection problems, especially for both ranking and classification. In this paper, we show how to formulate the feature selection problem as a Quadratic Unconstrained Binary Optimization (QUBO) problem which can be solved using Quantum Annealing (QA) and show that a quantum computer is able to more efficiently solve the feature selection problem, for both ranking and classification, with an effectiveness comparable to classical solvers. The results show that quantum computing approaches have become a viable option for the feature selection problem in IR and that they are worthy of further investigation.

2. Methodology and Experimental Pipeline

Quadratic models for feature weighting and selection have been studied for several years [4, 5]. In this work, we focus only on *feature selection*, since it is a more difficult NP-hard problem. In particular, we take into consideration three approaches that model the feature selection problem with the Quadratic Unconstrained Binary Optimization (QUBO) formulation: Mutual Information QUBO (MIQUBO) based on the mutual information maximization [6, 7, 8]; QUBO-Correlation that maximizes the Pearson correlation between selected features and the target variable while minimizing the correlation between features [9]; and QUBO-Boosting that is based on multiple single features Support Vector Classifier (SVC) [10][1]. The QUBO formulation is rather flexible and such problems can be solved with several techniques. Among them, in this work we consider the following: *Simulated Annealing (SA)* [11], *Tabu Search (TS)* [12], and *Steepest Descent (SD)*. Besides traditional approaches, QUBO problems are particularly well suited to be solved via Quantum Annealing (QA). QA is a binary optimization technique performed via a special-purpose physical device – a Quantum Annealer or Quantum Processing Unit (QPU) – capable of escaping local minima thanks to a quantum mechanical phenomenon called tunneling. In this work, the D-Wave Advantage QPU is used, which has 5600 qubit and a topology called Pegasus, in which every qubit is connected to 15 others [13]. Because of the sparse connections between qubits, a QUBO problem has to be first adapted to the QPU topology via *minor embedding* [14]. The embedded problem is served to the QPU on the cloud via APIs and low-energy solutions are sampled by repeating the QA process, which by default has a duration of $20\mu s$. If a QUBO problem cannot fit on the QPU, a hybrid quantum-classical approach can be used, to decompose the problem in smaller ones that can be solved directly on the QPU. This can be done with the D-Wave Leap[2] Hybrid cloud service.

[1]For more details on these QUBO formulations, we refer to the original work [1]
[2]D-Wave Leap - https://cloud.dwavesys.com/leap/

Table 1
Classification accuracy for all QUBO feature selection methods. Superscript ^{T} indicates statistical difference with TS. Results are missing for the QPU when the problem required more qubits than the available ones.

		All features		QPU		Hybrid		SA		SD		TS	
Method	Dataset	F	Acc.	N	Acc.	N	Acc.	N	Acc.	N	Acc.	N	Acc.
QUBO Correlation	spambase	57	0.9631	47	0.9573	47	0.9594	51	0.9602	50	0.9587	56	**0.9660**
	nomao	118	0.9654	93	**0.9659**	104	0.9649	110	0.9654	111	0.9650	118	0.9654
	isolet	617	**0.9432**		-	565	0.9402	565	0.9406	556	0.9402	617	0.9406
	gisette	5000	0.9676		-	4536	0.9471^T	4536	0.9486	4536	0.9462	2560	**0.9681**
QUBO Boosting	spambase	57	**0.9631**	37	0.9558	35	0.9529	37	0.9573	36	0.9566	53	0.9587
	nomao	118	0.9654	64	0.9655	79	0.9645	79	0.9641	76	0.9650	112	**0.9667**
	isolet	617	**0.9432**		-	467	0.9380	495	0.9415	495	0.9406	604	0.9380
	gisette	5000	0.9676		-	1750	0.9729	2079	**0.9752**	1002	0.9733	2558	0.9690
MIQUBO	spambase	57	0.9631	52	0.9638	53	0.9616	52	0.9645	54	0.9638	57	**0.9660**
	nomao	118	0.9654	96	**0.9667**	109	0.9656	118	0.9656	113	0.9661	118	0.9656
	isolet	617	0.9432		-	501	**0.9436**	617	0.9389	600	0.9410	617	0.9427
	gisette	5000	0.9676		-	1076	0.9743	1382	**0.9771**	2705	0.9710	2598	0.9686

Experimental Pipeline The effectiveness of the QPU is assessed on two different tasks, classification and ranking[3]. Given the usually high number of features, feature selection is important and largely used in classification, in order to identify the most useful subset of features for the specific task. For the classification experiments, we considered 9 datasets from OpenML [15], ranging from 34 to 5000 features. We report here results only for 4 of them: `spambase`, `nomao`, `isolet`, and `gisette`. The algorithm used for the classification task is Random Forest. Concerning the ranking task, to evaluate different feature selection strategies in a traditional IR setting, we consider the Learning to Rank (LtR) task. Following previous literature on LtR, we adopt three LETOR datasets (i.e., `OHSUMED`, `MQ2007`, and `MQ2008`) to assess the performance of different feature selectors. We use LambdaMART [16] as ranking algorithm.

3. Results and Discussion

The results obtained by the QUBO solvers are compared and displayed in Table 1 (Classification Task) and Table 2 (Ranking Task). To determine statistically significant differences between features selection strategies applied to the classification task, we employ the McNemar's test with significance level $\alpha = 0.05$ and Bonferroni correction following the procedure described by Japkowicz and Shah [17], while, for the ranking task, we applied ANalysis Of VAriance (ANOVA) as employed by Banks et al. [18]. Concerning the effectiveness of the solvers, the first observation that can be made is that there is no single QUBO solver that is superior to the others, rather, the solver that is able to achieve the best result is different depending on the dataset. Across all experiments TS is able to reach the best result 11 times, SA and QPU 8 times, Hybrid 7 times and SD 6 times. This is likely due to the peculiarities of each dataset, task and, to some extent, the stochastic nature of some solvers. Some differences instead emerge by comparing across tasks, in particular no feature selection approach is able to improve the effectiveness on dataset `MQ2008` and the Hybrid solver is slightly less effective when applied to the Ranking task. The behavior of the various QUBO solvers remains instead consistent across the feature

[3]Source code available at: https://github.com/qcpolimi/SIGIR22_QuantumFeatureSelection.git

Table 2
NDCG@10 and computation time on the Ranking task for all QUBO feature selection methods.

Method	Dataset	All features		QPU		Hybrid		SA		SD		TS	
		F	NDCG	N	NDCG	N	NDCG	N	NDCG	N	NDCG	N	NDCG
QUBO Correlation	OHSUMED	45	0.388	35	0.371	44	0.366	44	**0.390**	44	0.387	26	0.333
	MQ2007	46	0.472	43	0.473	39	0.472	39	0.474	28	0.473	24	**0.476**
	MQ2008	46	**0.489**	4	0.483	21	0.468	21	0.466	19	0.485	25	0.461
QUBO Boosting	OHSUMED	45	0.388	9	0.346	32	0.363	19	0.338	40	**0.400**	23	0.388
	MQ2007	46	0.472	43	**0.476**	37	0.464	42	0.463	36	0.464	35	0.466
	MQ2008	46	**0.489**	8	0.460	8	0.474	20	0.485	39	0.476	34	0.485
MIQUBO	OHSUMED	45	0.388	11	0.369	17	0.375	17	**0.394**	6	0.376	4	0.388
	MQ2007	46	0.472	25	**0.480**	34	0.472	34	0.469	34	0.472	2	0.472
	MQ2008	46	**0.489**	1	0.474	18	0.479	18	0.479	18	0.479	18	**0.489**

selection methods, although different QUBO heuristics result in different overall effectiveness. For example, MIQUBO appears to produce better results compared to the others. Looking at the QPU solver, its effectiveness is very close, if not almost identical, to that of the other solvers, sometimes resulting in the best selection of features. In very few cases the solution obtained with the QPU is worse compared to the other solvers, but within 5% of the best one. This result indicates that the QPU is indeed a reliable solver that can be used to tackle real problems across different datasets, heuristics and tasks. Note however that the largest problem that could be solved directly on the QPU had 124 features. Although the QPU has more than 5000 qubits, the QUBO matrix resulting from the feature selection problem is fully-connected and therefore its structure is difficult to fit on the limited connectivity structure of the QPU, therefore the problem size that can fit the hardware is greatly reduced. For larger problems it is still possible to use the Hybrid QPU-Classical approach, which was used for datasets of up to 5000 features but is able to tackle even larger problems. With respect to the computational time needed to solve the QUBO problems with different solvers, some patterns can be denoted. In general, SD is faster than QPU, SA and TS. It is also faster than Hybrid on datasets with only hundreds of features, but it scales worse, thus Hybrid comes out as the fastest solver for bigger datasets. The QPU is almost always faster than TS and SA, but slower than the latter for the MIQUBO model. Finally, most of the time currently required to solve a fully-connected QUBO problem with a QPU can be eliminated by offering pre-built embeddings and by reducing the network delay and queuing time required to access the quantum computer on the cloud.

Discussion and Future Works Overall, this work has shown that Quadratic Unconstrained Binary Optimization (QUBO) and Quantum Annealing (QA) are viable options for improving feature selection for both classification and ranking. QA is an emerging technology that has now evolved to the point where it can be used to tackle real problems and is easily accessible for researchers. There is indeed much room for improvement, which could lead to more competitive results. It would be worth if we, as a community, undertake a systematic exploration of these promising research directions, by developing a formulation suitable for applying quantum computing approaches to other relevant tasks. It would be extremely valuable if IR large scale evaluation campaigns take a lead and promote the organization of shared activities for exploring the application of quantum computing to IR, NLP, and RecSys in a comparable and shared way.

References

[1] M. Ferrari Dacrema, F. Moroni, R. Nembrini, N. Ferro, G. Faggioli, P. Cremonesi, Towards feature selection for ranking and classification exploiting quantum annealers, in: SIGIR 2022, 2022. URL: https://doi.org/10.1145/3477495.3531755. doi:10.1145/3477495.3531755.

[2] S. Marchesin, A. Purpura, G. Silvello, Focal elements of neural information retrieval models. An outlook through a reproducibility study, Information Processing & Management 57 (2020) 102109:1–102109:29.

[3] E. M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?, in: M. C. Elish, W. Isaac, R. Zemel, L. Irani, S. Kannan, M. Mitchell (Eds.), Proc. ACM Conference on Fairness, Accountability, and Transparency (FAccT 2021), ACM Press, New York, USA, 2021, pp. 610–623.

[4] I. Rodríguez-Luján, R. Huerta, C. Elkan, C. S. Cruz, Quadratic programming feature selection, J. Mach. Learn. Res. 11 (2010) 1491–1516. URL: http://portal.acm.org/citation.cfm?id=1859900.

[5] A. Katrutsa, V. V. Strijov, Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria, Expert Syst. Appl. 76 (2017) 1–11. URL: https://doi.org/10.1016/j.eswa.2017.01.048. doi:10.1016/j.eswa.2017.01.048.

[6] Z. Zeng, H. Zhang, R. Zhang, C. Yin, A novel feature selection method considering feature interaction, Pattern Recognit. 48 (2015) 2656–2666. URL: https://doi.org/10.1016/j.patcog.2015.02.025. doi:10.1016/j.patcog.2015.02.025.

[7] H. Peng, F. Long, C. H. Q. Ding, Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1226–1238. URL: https://doi.org/10.1109/TPAMI.2005.159. doi:10.1109/TPAMI.2005.159.

[8] K. Torkkola, Feature extraction by non-parametric mutual information maximization, J. Mach. Learn. Res. 3 (2003) 1415–1438. URL: http://jmlr.org/papers/v3/torkkola03a.html.

[9] A. J. Ferreira, M. A. T. Figueiredo, Efficient feature selection filters for high-dimensional data, Pattern Recognit. Lett. 33 (2012) 1794–1804. URL: https://doi.org/10.1016/j.patrec.2012.05.019. doi:10.1016/j.patrec.2012.05.019.

[10] H. Neven, V. S. Denchev, G. Rose, W. G. Macready, Qboost: Large scale classifier training with adiabatic quantum optimization, in: S. C. H. Hoi, W. L. Buntine (Eds.), Proceedings of the 4th Asian Conference on Machine Learning, ACML 2012, Singapore, Singapore, November 4-6, 2012, volume 25 of *JMLR Proceedings*, JMLR.org, 2012, pp. 333–348. URL: http://proceedings.mlr.press/v25/neven12.html.

[11] S. Kirkpatrick, D. G. Jr., M. P. Vecchi, Optimization by simmulated annealing, Sci. 220 (1983) 671–680.

[12] G. Palubeckis, Multistart tabu search strategies for the unconstrained binary quadratic optimization problem, Ann. Oper. Res. 131 (2004) 259–282. URL: https://doi.org/10.1023/B:ANOR.0000039522.58036.68. doi:10.1023/B:ANOR.0000039522.58036.68.

[13] K. Boothby, P. Bunyk, J. Raymond, A. Roy, Next-generation topology of d-wave quantum processors, arXiv preprint arXiv:2003.00133 (2020).

[14] V. Choi, Minor-embedding in adiabatic quantum computation: I. the parameter set-

ting problem, Quantum Inf. Process. 7 (2008) 193–209. URL: https://doi.org/10.1007/s11128-008-0082-9. doi:10.1007/s11128-008-0082-9.
[15] J. Vanschoren, J. N. van Rijn, B. Bischl, L. Torgo, Openml: Networked science in machine learning, SIGKDD Explorations 15 (2013) 49–60. URL: http://doi.acm.org/10.1145/2641190.2641198. doi:10.1145/2641190.2641198.
[16] C. J. Burges, From ranknet to lambdarank to lambdamart: An overview, Learning 11 (2010) 81.
[17] N. Japkowicz, M. Shah, Evaluating learning algorithms: a classification perspective, Cambridge University Press, 2011.
[18] D. Banks, P. Over, N. Zhang, Blind men and elephants: Six approaches to trec data (1998). URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=151743.