Clustering Classical Data with Quantum k-Means[⋆]

Alessandro Poggiali[1,*,†], Alessandro Berti[2,†], Anna Bernasconi[2,†], Gianna M. Del Corso[2,†] and Riccardo Guidotti[2,†]

[1] *University of Piemonte Orientale, Italy*

[2] *University of Pisa, Italy*

Abstract

In the last years, we have witnessed an unstoppable growth of data created, captured, copied, and consumed globally by more and more devices. The demand for such an increasing amount of information to be processed led to research towards higher computational power systems and specialized algorithms. Among them, quantum computing is a promising paradigm based on quantum theory for performing fast computations. Quantum algorithms are expected to surpass their classical counterparts in terms of computational complexity for certain kinds of tasks, and machine learning is one of them, so the subfield of Quantum Machine Learning is one of the most promising. In this work, we design a hybrid quantum algorithm for k-Means. The main idea of our algorithm is to compute in a quantum way the distance between pairs of records in the input dataset. We show that our quantum algorithm could be, in principle, more efficient than the classical k-Means, yet obtain comparable clustering results.

Keywords

Quantum Machine Learning, Clustering, Data Mining

1. Introduction

Quantum Machine Learning (QML) is the branch of Quantum Computing (QC) that attempts to adapt classical data mining and machine learning algorithms, or their expensive subroutines, to run on a potential quantum computer. Indeed, the expectation is that such machines will be commonly available for applications in the near future. Many QML algorithms have been recently studied [1]. However, the translation of classical algorithms into their quantum counterpart named "quantization" is not trivial and hides many difficulties. In this work, we focus on the quantization of k-Means [2], which is one of the most famous algorithms used for clustering. The k-Means clustering algorithm is an unsupervised learning algorithm, and its goal is to find natural groups of elements in a data set. In particular, the elements inside a group are more similar to the central element of the group than to the central elements of other groups, according to a specific distance measure. Building a quantum version of this algorithm means creating a quantum circuit that takes classical data as input and exploits quantum gates to perform the computation, satisfying all the quantum mechanical constraints. Nowadays quantum computers are in the *noisy intermediate-scale quantum (NISQ) era*[3]. This means that today, quantum computers are prone to noises that generate errors in computation. This is

Proceedings of the 23rd Italian Conference on Theoretical Computer Science, Rome, Italy, September 7-9, 2022

[*] Corresponding author.

[†] These authors contributed equally.

known as the decoherence problem. Typically, the deeper the circuit, the more prone the output is to errors that make it unreliable.

For this reason, this work presents a hybrid k-Means that exploits a quantum subroutine to boost the distance computation between each record and each centroid while the overall algorithm remains classical. This solution permits the mitigation of the decoherence problem in this NISQ era. In our analysis, we use the classical δ-k-Means algorithm to provide a measure of error for our hybrid k-Means that we call q-k-Means algorithm. The experiments show that our quantum algorithm could be, in principle, more efficient than its classical counterpart yet obtain comparable clustering results.

The rest of the paper organizes as follows. Section 2 provides the related works, Section 3 sets up the stage by defining the notations, Section 4 describes the Quantum k-Means, Section 5 reports the experimental results, and eventually, Section 6 summarizes contributions.

2. Related Works

In general, the problem of quantum clustering can be addressed in several ways. Some studies were inspired by quantum theory. For instance, the classical clustering method proposed in [4] is based on physical intuition derived from quantum mechanics. In [5], the authors perform clustering by exploiting a well-known reduction from clustering to the Maximum-Cut problem that is then solved using a quantum algorithm for approximate combinatorial optimization. In [6], the authors present an unsupervised quantum learning algorithm for k-Means clustering based on adiabatic quantum computing [7].

The general idea for quantizing classical clustering algorithms is to substitute the most expensive parts in the algorithm with more efficient quantum subroutines. For instance, in [8] the fidelity distance measure is used for distance computation between each pair of states in the dataset. The fidelity is efficiently estimated with a quantum circuit containing only a few gates (two Hadamard gates and a Control-Swap). In this way, the algorithm can perform clustering directly on quantum states. However, the paper lacks a discussion on how the algorithm could deal with classical data.

As alternative approaches, full quantum routines for clustering have been proposed. In [9] two subroutines based on Grover's search algorithm [10] are used to accelerate classical clustering methods. For the k-Means quantization, the authors use the subroutines in this way. First, the total distance of each state to all other states of one cluster is computed with the help of an oracle that calculates the distance between two quantum states. Then, another routine finds the smallest value of this distance function in order to select a quantum state as the new centroid for the cluster. Unfortunately, this approach cannot be used in practice because the oracle is not described in detail.

Finally, in [11] it is proposed a quantum version of k-Means (called q-Means) that provides an exponential speedup in the number of records of the dataset compared to the classical version. Moreover, q-Means returns explicit classical descriptions of the final centroids. Although q-Means looks promising from a practical point of view, the paper discussion is strictly theoretical. The experiments are performed using a classical algorithm (δ-k-Means) simulating q-Means, instead of a real quantum version.

Algorithm 1: δ-k-Means

```
Input: D - input data, k - number of clusters
Output: $L$ - records to clusters assignment, C - centroids
$C \leftarrow initCentroids(D, k)$ ;                              // centroid initialization
while convergence is not achieved do
    for $\vec{r} \in D$ do                                              // for each record
        $\vec{c} \leftarrow argmin(d(\vec{r}, \vec{c}_j)) \, \forall \vec{c}_j \in C$ ;                  // find nearest centroid
        $L_\delta(\vec{r}) \leftarrow \{\vec{c}_p : |d^2(\vec{r}, \vec{c}) - d^2(\vec{r}, \vec{c}_p)| \leq \delta\}$ ;      // find possible labels
        $c_j \leftarrow rand(L_\delta(\vec{r}))$ ;                          // pick a random centroid
        $C_j \leftarrow C_j \cup \{\vec{r}\}$ ;                              // assign $\vec{r}$ to cluster $C_j$
        $L(\vec{r}) \leftarrow j$ ;                                      // assign label $j$ to $\vec{r}$
    for $j \in [1, k]$ do                                           // for each centroid
        $\vec{c}_j \leftarrow \frac{1}{|C_j|} \sum_{\vec{r} \in C_j} \vec{r}$ ;                       // update cluster center $\vec{c}_j$
return $L, C$ ;                                        // return assignments and centroids
```

Different from the literature, our work concentrates on practical problems arising when implementing a quantum clustering algorithm, with particular attention to the encoding of classical data.

3. Setting the stage

We keep our paper self-contained by summarizing the key concepts necessary to comprehend our work. Given a dataset $D = \{\vec{r}_1, \ldots, \vec{r}_M\}$ of M records where every record $\vec{r}_i$ is a N-dimensional vector of numerical value, the goal of clustering is to assign each record to one out of k different clusters $\{C_1, \ldots, C_k\}$, represented by centroids $\{\vec{c}_1, \ldots, \vec{c}_k\}$ respectively, so that similar records share the same assignment. The k-Means algorithm is one of the most famous clusterings algorithms [12]. After randomly[1] choosing k initial centroids, the algorithm consists of two repeated steps until a certain convergence condition is met. The first step is the *cluster assignment* step: every element in the dataset has to be assigned to its closest centroid according to a specific distance measure. As distance, we consider the *Eucledian distance* that is defined as $d\left(\vec{p}, \vec{q}\right) = \sqrt{\sum_{i=1}^{N} \left(p_i - q_i\right)^2}$, where p and q are two N-values vectors. The second step is the *centroids update* step, where for every cluster we recompute, the new cluster center to be used as a centroid for the next iteration. In this work, we concentrate on the cluster assignment step whose time complexity is $\mathcal{O}(kMN)$ where k is the number of centroids, M is the number of records, and N their dimension.

As already mentioned in Section 2, a quantum version of k-Means, called q-Means, is proposed in [11]. The authors evaluate q-Means by means of δ-k-Means, a "quantum-approximation" of k-Means that simulates the quantum calculus that q-Means is supposed to do. More precisely, δ-k-Means simulates the classical k-Means algorithm as performed in a quantum environment.

[1] We adopt the clever initialization proposed in [13].

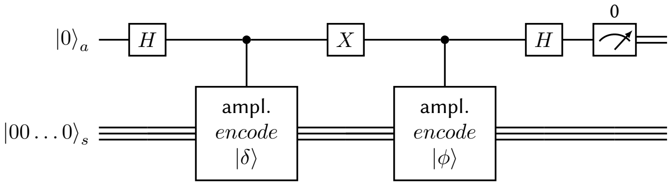
$|0\rangle_a$
$|00\dots0\rangle_s$
H
X
H
0
ampl. *encode* $|\delta\rangle$
ampl. *encode* $|\phi\rangle$

Figure 1: Quantum circuit for the Euclidean distance

Since a quantum algorithm can introduce errors due to *decoherence* and *noise* present in quantum machines, δ-k-Means simulates such errors by introducing some noise in both steps of k-Means. However, as discussed in Section 4, in the quantum version of k-Means we propose in this paper, we keep the second step (i.e., centroids update) classical. For this reason, we will consider in the experiments a slightly different version of δ-k-Means where we introduce the noise δ only in the cluster assignment step.

In Algorithm 1, we show the pseudocode of the updated version of the considered δ-k-Means. Let $\vec{c}$ be the centroid closest to the point $\vec{r}$. Then, δ-k-Means defines the set of possible labels $L_\delta(\vec{r})$ for $\vec{r}$ as follows:

$$L_\delta(\vec{r}) = \{\vec{c}_p : \left| d^2(\vec{c}, \vec{r}) - d^2(\vec{c}_p, \vec{r}) \right| \leq \delta\}.$$

When $\delta = 0$, δ-k-Means is equivalent to the standard k-Means since no uncertainty is included and $L_\delta(\vec{r})$ contains only the closest centroid. On the other hand, a high value of δ allows δ-k-Means to include in $L_\delta(\vec{r})$ not very close centroids, bringing more noise in the entire procedure. Indeed, the assignment rule selects randomly a cluster label from the set $L_\delta(\vec{r})$ (see line 5 in Algorithm 1). In [11] it is proven that if the data are "well-cluserable" (see [11] for the detailed definition) and the centroids are well separated, the δ-k-Means algorithm succeeds assigning the right cluster to most of the points for a suitable value of δ depending on the separation of the centroids.

4. Quantum k-Means

Classical information can be encoded in different ways into a quantum state. In [14], the authors revisit several data encoding strategies and quantum distance algorithms. The process of encoding input numerical features into the amplitude of a quantum system is called *amplitude encoding* [15]. Amplitude encoding allows the design of quantum circuits that compute distances between quantum states.

One distance measure commonly used in ML is the Euclidean distance [12]. We show here a general circuit for computing a quantum Euclidean distance $d(\delta, \phi)$ between two general quantum states $|\delta\rangle$, $|\phi\rangle$ encoded in a register s via amplitude encoding [16]. To compute this distance, we use an additional ancilla qubit a entangled with the two states $|\delta\rangle$ and $|\phi\rangle$. This can be accomplished by first applying an Hadamard gate on the ancilla a, and then by loading in the register s the two states $|\psi\rangle$ and $|\delta\rangle$ conditioned on the ancilla. Then, the initial state $|0\rangle_a |00\cdots0\rangle_s$ evolves in $\frac{1}{\sqrt{2}}(|0\rangle_a |\delta\rangle_s + |1\rangle_a |\phi\rangle_s)$. Eventually, we apply a Hadamard gate

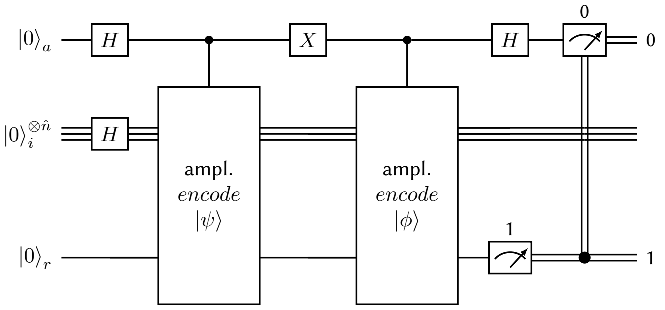
$|0\rangle_a$
$|0\rangle_i^{\otimes \hat{n}}$
$|0\rangle_r$
H
H
X
H
ampl. *encode* $|\psi\rangle$
ampl. *encode* $|\phi\rangle$
0
1
0
1

Figure 2: QC: quantum Euclidean distance with FF-QRAM.

on ancilla a. The corresponding state becomes: $\frac{1}{2}\big(|0\rangle_a(|\delta\rangle_s + |\phi\rangle_s) + |1\rangle_a(|\delta\rangle_s - |\phi\rangle_s)\big)$. The probability of measuring the ancilla in the state $|0\rangle_a$ is given by $p_a = \frac{1}{4}\|\delta + \phi\|_2^2$ which corresponds to $p_a = 1 - \frac{1}{4}\|\delta - \phi\|_2^2 = 1 - \frac{1}{4}d(\delta, \phi)^2$, since δ and ϕ are unit vectors. The overall circuit computing the Euclidean distance between two states δ and ϕ is illustrated in Figure 1.

In the rest of this section, we propose our quantum version of k-Means. In particular, what we want to *"make quantum"* is the computation of the distance between two records. This is similar to what some previous works have proposed [8], but here we give a practical implementation of the entire algorithm.

In order to efficiently load classical data in a suitable quantum state, we employ the FF-QRAM algorithm [17]. In particular, we deal with N-feature vectors, and we use the FF-QRAM algorithm to amplitude encode each vector.

Once data have been loaded, we build a quantum circuit that computes the distance between a single pair of vectors. In the k-Means context, we compute distances between every record and every centroid in order to assign a cluster label to every record in the dataset. This can be accomplished by using the quantum circuit shown in Figure 1 that computes the Euclidean distance between two quantum states generated by the amplitude encoding technique.

Figure 2 illustrates the whole quantum circuit (QC). QC can compute the Euclidean distance simultaneously between the features of two vectors. The $\hat{n}$-qubit register $|i\rangle$ is the index register for the N features of each vector. It consists of $\hat{n} = \lceil \log_2 N \rceil$ qubits which control the rotation of a qubit $|r\rangle$.

To get a proper estimate of the Euclidean distance, we need to repeat the execution of this circuit a certain number of times t. In this way, we can estimate the Euclidean distance as $d(\psi, \phi) = \sqrt{4 - 4\left(\frac{\#|0\rangle_a}{t'}\right)}$, where $\#\,|0\rangle_a$ is the number of times the ancilla qubit is measured in the state $|0\rangle$. Actually, also the FF-QRAM encoding procedure requires a post-selection on the qubit $|r\rangle$ (see [17, 14] for more details). Thus, $\#\,|0\rangle_a$ is the number of times the outcome for ancilla is 0 after having post-selected the result where qubit $|r\rangle$ was in 1. Notice that, here, $t' < t$ is not the total number of repetitions, but the number of times the post-selection on $|r\rangle$ is successful.

Algorithm 2 describes the pseudocode of the proposed procedure, q-k-Means, for cluster computation. We adopt k-Means++ [13] as centroid initialization strategy (line 1). Then, we

```
Algorithm 2: q-$k$-Means
  Input: $D$ - input data, $k$ - number of clusters, $t$ - number of quantum shots
  Output: $L$ - records to clusters assignment, C - centroids
1  $C \leftarrow initCentroids(D, k)$ ;                          // centroid initialization
2  while centroids do not change do
3      for $\vec{r} \in D$ do                                    // for each record
4          $v \leftarrow \infty$;                                // init distance
5          for $j \in [1, k]$ do                                 // for each centroid
6              $\# |0\rangle_a \leftarrow$ QC($t$,$\vec{r}$,$\vec{c}_j$) ;     // quantum circuit executed $t$ times
7              $d \leftarrow \sqrt{4 - 4\left(\frac{\#|0\rangle_a}{t'}\right)}$ ;   // Euclidean distance estimation
8              if $d \leq v$ then                                // if closer to current centroid
9                  $v \leftarrow d$;                             // update current distance
10                 $C_j \leftarrow C_j \cup \{\vec{r}\}$ ;       // assign $\vec{r}$ to cluster $C_j$
11                 $L(\vec{r}) \leftarrow j$ ;                   // assign label $j$ to $\vec{r}$
12     for $j \in [1, k]$ do                                     // for each centroid
13         $\vec{c}_j \leftarrow \frac{1}{|C_j|} \sum_{\vec{r} \in C_j} \vec{r}$ ;   // update cluster center $\vec{c}_j$
14 return $L, C$ ;                                               // return assignments and centroids
```

compute the quantum Euclidean distance between each record $\vec{r}$ and each centroid $\vec{c}_j$. In particular, the quantum distance computation is repeated for each of the Mk pairs of vectors, where M is the number of records in the dataset and k is the number of centroids (lines 3-7). Then, the algorithm assigns $\vec{r}$ to the cluster C_j such that the distance between $\vec{r}$ and the centroid $\vec{c}_j$ of the cluster C_j is the minimum (lines 8-10). Eventually, each centroid $\vec{c}_j$ is updated (lines 11-12). The output of *q-k*-Means consists of a final cluster assignment L and the corresponding centroids C.

Our hybrid approach improves the *cluster assignment step* with respect to the classical k-Means by a factor N if we do not consider the QRAM preparation. In particular, the overall complexity of the *cluster assignment step* is $O(Mk)$, where M is the number of records and k is the number of centroids, plus the cost of the QRAM preparation.

5. Experiments

In this section, we assess the effectiveness of the *q-k*-Means algorithm[2]. We intend to evaluate the capability of *q-k*-Means in terms of clustering quality by simulating it on a number of datasets. Since quantum computers currently available are not large enough to test *q-k*-Means, we exploit the QASM_SIMULATOR provided by QISKIT[3].

[2] Python code at: https://github.com/AlessandroPoggiali/Qkmeans-ICTCS

[3] Qiskit library: https://qiskit.org/

For an evaluation of the algorithm as much complete as possible, the parameters that we are going to test follow:

- SHOTS (t): how many times we repeat the execution of a quantum circuit.
- SC_THRESH: it rules the stopping condition of the algorithm. It is defined as the relative tolerance with regards to the Frobenius norm of the difference in the cluster centers of two consecutive iterations to declare convergence.
- MAX_ITERATIONS (MAX_ITE): the maximum number of iterations the algorithm can perform.

If not differently specified, we tested q-k-Means with the following parameters: MAX_ITE: 10, SC_THRESH: 0.0001, and SHOTS: 8192 on all four synthetic datasets.

In order to evaluate the algorithm performance and the cluster quality, our analysis considers the following measures.

- N_ITE (ITE): the actual number of iterations q-k-Means performs to converge.
- AVG_SIMILARITY (SIM): the concept of similarity is defined in terms of how accurately q-k-Means assigns the right centroids to records with respect to the classical assignments. The AVG_SIMILARITY measure is basically the average similarity among all iterations of the algorithm.
- SILHOUETTE (SIL): the Silhouette Coefficient measures how an element is similar to its cluster with respect to the other clusters.
- SSE [18]: it corresponds to the sum of the squared differences between each record and its centroid.
- V_MEASURE (VM) [19]: it measures the correctness of the clustering assignments with respect to ground truth.

Dataset. For the q-k-Means assessment, we consider two groups of datasets: the first group of four **synthetic datasets** and two **real datasets**. The synthetic datasets come from the clustering guide of `scikit-learn`[4]. These datasets show the characteristics of different clustering algorithms on datasets that are "interesting" but still in two dimensions. For these datasets, we also have the ground truth (i.e., the actual clustering we want to obtain) so that we can have an objective evaluation of the algorithm. They include: ANISO, BLOBS, BLOBS2, and MOON (also known as NOISYMOON). While instead, the real datasets we take into consideration are: IRIS, and DIABETES. Regarding the real datasets, we have no ground truth. Thus, we have to choose a suitable number of centroids k. As synthetic datasets, the real datasets are available on `scikit-learn`.

Data Preprocessing. The essential data preprocessing required by q-k-Means consists of two steps: standardization and normalization. The standardization of the dataset has the effect of having zero mean and unit variance among all samples. This is common practice in ML

[4] https://scikit-learn.org/stable.

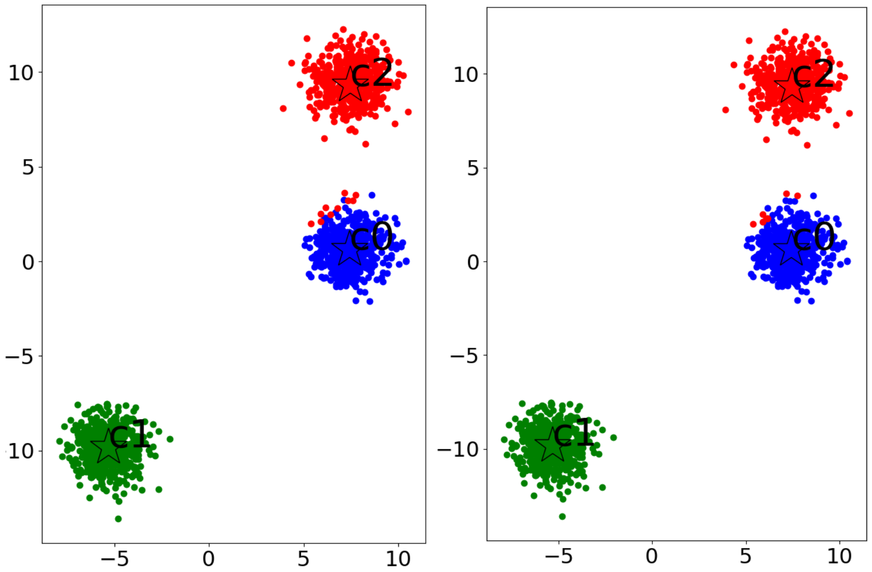

Figure 3: Clustering result with *1-norm* preprocessing left, *inf-norm* right.

Table 1
q-k-Means results on BLOBS2 dataset.

Preprocessing	ITE	SIM	SSE	SIL	VM
1-norm	8	99.62	177.18	0.857	0.970
inf-norm	7	99.65	422.23	0.865	0.983

to compensate for scaling effects and to ensure that the data does not only populate a small subspace of the input space. In fact, input spaces in higher dimensions lead to indistinguishably small distances between data points. The normalization applies row by row and allows us to deal with unit length vectors. While the dataset standardization is always applied, we have two different preprocessing techniques available: the normalization to unit length (*1-norm*) and the *inf-norm* preprocessing, which is equivalent to dividing the entries of the vector by the component of maximum modulus.

Finally, vector values are converted to suitable angles in order to encode them in the FF-QRAM. Note that the ancilla post-selection probability for the quantum Euclidean distance is always around 0.5 [15]. However, we can enhance the FF-QRAM post-selection probability applying the *inf-norm* normalization [20].

Looking at the algorithm assessments, the execution with the *inf-norm* gives comparable results with respect to the *1-norm* (see Table 1). In this test, we executed q-k-Means on BLOBS2 dataset using 1024 shots. The Silhouette score tells us that in the *inf-norm* case, we obtain a better clustering, and also the *v_measure* tells us that the final assignment is closer to the ground truth. Instead, the SSE in the *inf-norm* case is worse than in the *1-norm* case. This depends on the range of input data, and hence it does make sense only when comparing results whereby input data have their range of values comparable. We report in Figure 3 the output in

Table 2
q-k-Means, δ-k-Means, and k-Means results on synthetic datasets.

	q-k-Means					δ-k-Means						k-Means			
	ITE	SIM	SSE	SIL	VM	δ	ITE	SIM	SSE	SIL	VM	ITE	SSE	SIL	VM
ANISO	10	96.6	2569.1	.64	.70	1.4	10	96.77	2279.2	.68	.76	6	2208.7	.72	.85
BLOBS	7	99.6	422.23	.86	.98	1.5	10	99.6	409.94	.86	.98	2	396.85	.87	.99
BLOBS2	10	97.6	2420.5	.70	.67	1.7	10	97.5	2353.4	.70	.67	3	2318.7	.71	.69
MOON	5	99.2	7185.8	.55	.39	4.1	10	99.2	7191.1	.55	.38	2	7152.1	.55	.37

Table 3
Confusion matrix k-Means vs q-k-Means: the True Positive (TP) column report the percentage of sample pairs whereby both clusterings group them together, the others column follow.

	TP	FP	FN	TN
ANISO	61.25%	3.78%	4.93%	30.04%
BLOBS	66.58%	0.13%	0.13%	33.16%
BLOBS2	62.35%	1.13%	1.03%	35.49%
MOON	49.17%	0.86%	0.86%	49.11%

the original space of the final clustering assignment. We observe that we get a more accurate clustering in the second case, even though bad-clustered points are still present. Henceforth, we adopt the *inf-norm* as the default preprocessing.

Results on Synthetic Datasets. Table 2 reports the evaluation measures observed for q-k-Means on synthetic datasets. The values show good performance on aniso, blobs, and blobs2. In fact, the measures sil and vm highlight a good clustering output. While, in the moon dataset, in spite of having a high similarity value, the vm returned is low. This probably happens because the moon dataset, as it is generated, is not a well-clusterable dataset. In fact, it has no spherical clusters hence algorithms like k-Means will not be able to identify the right shapes of its clusters. Table 2 reports also the results obtained with δ-k-Means and k-Means. In particular, q-k-Means produces a clustering with error δ when the δ-k-Means similarity is comparable to the q-k-Means similarity. A final consideration concerns the confusion matrices in Table 3, where we report the percentages of *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, and *True Negative (TN)*. We observe that k-Means and q-k-Means typically output the same results on the datasets blobs, blobs2, and moon. In fact, the value $FP + FN$ in these datasets is quite small. This does not hold for the aniso dataset where, instead, $FP + FN = 8.71\%$.

Results on Real Datasets. With real datasets, we have no prior knowledge about the clustering result we should achieve. In other words, the number of clusters k has to be selected properly in order to get good clusterization. A very common heuristic used in clustering to determine the number of clusters in a dataset is the *elbow method* [12]. Our first test aims at performing the elbow method with the k-Means and q-k-Means algorithm separately on the iris dataset. We repeat the execution of both algorithms by varying k from 2 to 8. We obtain the curves in Figure 4. A suitable value of k for both algorithms is 3 because this is the point where the SSE stops decreasing sharply. The comparison between the classical and the quantum

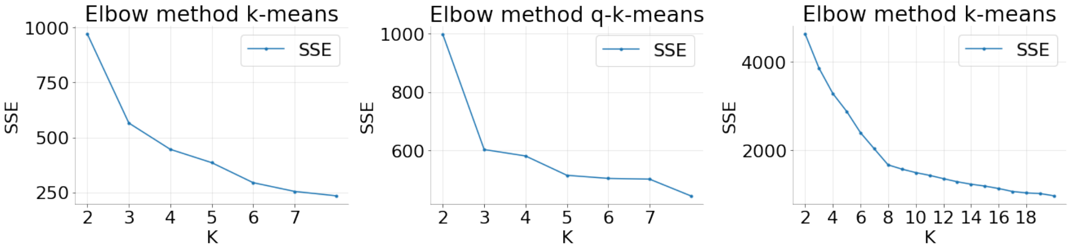

Figure 4: Elbow method: (left) on IRIS dataset with k-Means, (center) on IRIS dataset with q-k-Means, (right) on DIABETES dataset with k-Means.

Table 4
q-k-Means vs k-Means on IRIS and DIABETES.

	IRIS					DIABETES				
	δ	ITE	SIM	SSE	SIL	δ	ITE	SIM	SSE	SIL
k-Means	-	4	-	565.50	0.51	-	7	-	1671.27	0.37
δ-k-Means	3.20	10	95.20	582.25	0.49	1.95	10	87.35	1863.52	0.32
q-k-Means	-	7	95.33	616.83	0.48	-	10	87.83	2147.90	0.20

algorithm with this configuration is reported in Table 4. We repeated the same experiment on the DIABETES dataset. Again, we select the right k using the elbow method with k-Means, and then we compare the result that q-k-Means gives us for the same k. Figure 4 shows that a good value for k is 8 for this dataset. Hence, by executing the q-k-Means algorithm with $k = 8$, we obtain the result in Table 4. The result shows that q-k-Means performs not well compared to δ-k-Means and k-Means according to SIL measure. A possible explanation can be related to the use of PCA to reduce the number of features from 10 to 4, which is necessary for executing the algorithm in a reasonable amount of time, due to the technological limits. This makes records belonging to different clusters too similar, and the number of shots was insufficient to estimate a sufficiently precise Euclidean distance, which therefore affected the quality of the clustering. Eventually, we observe that the SIL measures of k-Means and q-k-Means on the IRIS dataset are similar, while in the DIABETES dataset, k-Means performs better than q-k-Means.

q-k-Means on Real Quantum Hardware. All tests reported up to now were carried out using the QASM SIMULATOR, a simulator provided by QISKIT which simulates quantum computation by using classical hardware. Here, we show the best we can do with currently available quantum computers. IBM offers cloud access[5] to some of their quantum computers, so it is possible to delegate the execution of a quantum circuit to a real quantum machine. We had access to quantum computers with no more than five qubits. For this reason, we must consider simple instances of our q-k-Means where no more than five qubits are necessary.

The first test considers q-k-Means where we use three qubits: one qubit for the ancilla $|a\rangle$, one for the register $|r\rangle$, and one for addressing $N = 2$ features (i.e., $|i\rangle$). Instead of simulating each of the Mk quantum circuits locally, we first check the least busy quantum computer

[5] https://quantum-computing.ibm.com/

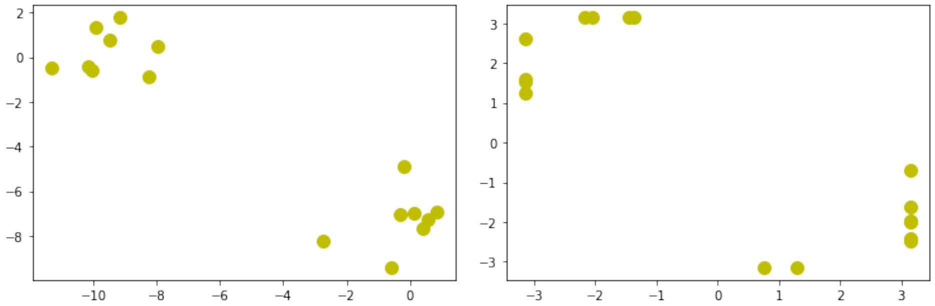

Figure 5: BLOBS3 dataset. (left) Original data, (right) *inf-norm* processing

Table 5
q-k-Means: real hardware vs simulator.

	ITE	SIM	SSE	SIL	VM
real hw	2	100	89.66	0.79	1
simulator	2	100	89.66	0.79	1

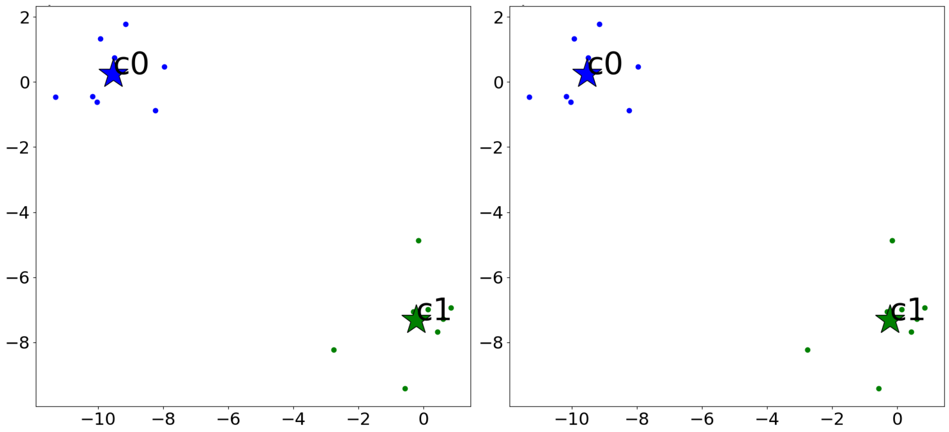

Figure 6: Clustering result on BLOBS3. (left) Real hardware, (right) Simulator.

available and send to it every circuit to be executed. This requires several steps, like waiting on the queue of a quantum computer and receiving the result, so it introduces an overhead. Furthermore, this communication overhead will be paid per pair of vectors, so it could highly affect the overall performance of the algorithm, especially for big datasets. For this reason, we took into account for this experiment a small dataset (BLOBS3) consisting of $M = 16$ two dimensional vectors, which form two well-distinguishable spherical clusters (Fig. 5). Notice that the number of clusters is not involved in the quantum circuit preparation, but we chose $k = 2$ to simplify the overall execution.

We compare the output of q-k-Means executed using real quantum hardware with the output of the algorithm executed by the simulator. In Table 5 we report this comparison, while in Figure 6 we show the clustering obtained.

From the table, we can see that both clusterization were successful with respect to the ground truth with the same number of iterations. However, to conclude, until a large-scale noise-free quantum computer is available, testing complex quantum circuits on real quantum hardware will be an unfeasible task.

6. Conclusion

We have proposed q-k-Means, a hybrid approach for clustering classical data. The algorithm implements a quantum subroutine to boost the *cluster assignment step* of the classical k-Means. In particular, this quantum subroutine computes the Euclidean distance between two N-dimensional vectors, i.e., a record and a cluster centroid. The complexity of this step is $O(Mk)$, where M and k are the number of records and the number of centroids, respectively. The experiments show that q-k-Means could be in principle more efficient than classical k-Means, yet obtaining comparable clustering results. In this work, we exploited quantum parallelism only to compute distances. Our future work is to design and analyze two variants of q-k-Means that leverage quantum parallelism to compute *(i)* the distances between a single record and k centroids simultaneously, and *(ii)* the distances between M records and k centroids simultaneously.

Acknowledgment. This work is supported by Università di Pisa under the "PRA – Progetti di Ricerca di Ateneo" (Institutional Research Grants) - Pr. no. PRA 2020-2021 92, "Quantum Computing, Technologies and Applications", and the work of Gianna Del Corso is also supported by GNCS-INdAM. This work has been partially supported by the European Community Horizon 2020 programme under the funding schemes: H2020-INFRAIA-2019-1: Research Infrastructure G.A. 871042 *SoBigData++*.

References

[1] M. Schuld, I. Sinayskiy, F. Petruccione, An introduction to quantum machine learning, Contemporary Physics 56 (2015) 172–185.

[2] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, Oakland, CA, USA, 1967, pp. 281–297.

[3] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum 2 (2018) 79.

[4] D. Horn, A. Gottlieb, Algorithm for data clustering in pattern recognition problems based on quantum mechanics, Physical review letters 88 (2001) 018702.

[5] J. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong, et al., Unsupervised machine learning on a hybrid quantum computer, arXiv preprint arXiv:1712.05771 (2017).

[6] S. Lloyd, M. Mohseni, P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv preprint arXiv:1307.0411 (2013).

[7] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, Quantum computation by adiabatic evolution, arXiv preprint quant-ph/0001106 (2000).

[8] E. Aïmeur, G. Brassard, S. Gambs, Machine learning in a quantum world, in: Conference of the Canadian Society for Computational Studies of Intelligence, Springer, 2006, pp. 431–442.

[9] E. Aïmeur, G. Brassard, S. Gambs, Quantum clustering algorithms, in: Proceedings of the 24th international conference on machine learning, 2007, pp. 1–8.

[10] L. K. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 1996, pp. 212–219.

[11] I. Kerenidis, J. Landman, A. Luongo, A. Prakash, q-means: A quantum algorithm for unsupervised machine learning, Advances in Neural Information Processing Systems 32 (2019).

[12] P. Tan, et al., Introduction to Data Mining, Addison-Wesley, 2005.

[13] S. Vassilvitskii, D. Arthur, k-means++: The advantages of careful seeding, in: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2006, pp. 1027–1035.

[14] A. Berti, A. Bernasconi, G. M. Del Corso, R. Guidotti, Effect of Different Encodings and Distance Functions on Quantum Instance-based Classifiers, in: Proceedings 26th Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD, 2022.

[15] M. Schuld, M. Fingerhuth, F. Petruccione, Implementing a distance-based classifier with a quantum interference circuit, EPL (Europhysics Letters) 119 (2017) 60002.

[16] M. Schuld, et al., Supervised learning with quantum computers, Springer, 2018.

[17] D. K. Park, F. Petruccione, J.-K. K. Rhee, Circuit-based quantum random access memory for classical data, Scientific reports 9 (2019) 1–8.

[18] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics 20 (1987) 53–65.

[19] A. Rosenberg, J. Hirschberg, V-measure: A conditional entropy-based external cluster evaluation measure, in: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), 2007, pp. 410–420.

[20] T. M. de Veras, I. C. De Araujo, D. K. Park, A. J. da Silva, Circuit-based quantum random access memory for classical data with continuous amplitudes, IEEE Transactions on Computers 70 (2020) 2125–2135.