Research Article

Alberto Amato and Vincenzo Di Lecce*

Data preprocessing impact on machine learning algorithm performance

https://doi.org/10.1515/comp-2022-0278
received December 1, 2022; accepted June 6, 2023

Abstract: The popularity of artificial intelligence applications is on the rise, and they are producing better outcomes in numerous fields of research. However, the effectiveness of these applications relies heavily on the quantity and quality of data used. While the volume of data available has increased significantly in recent years, this does not always lead to better results, as the information content of the data is also important. This study aims to evaluate a new data preprocessing technique called semi-pivoted QR (SPQR) approximation for machine learning. This technique is designed for approximating sparse matrices and acts as a feature selection algorithm. To the best of our knowledge, it has not been previously applied to data preprocessing in machine learning algorithms. The study aims to evaluate the impact of SPQR on the performance of an unsupervised clustering algorithm and compare its results to those obtained using principal component analysis (PCA) as the preprocessing algorithm. The evaluation is conducted on various publicly available datasets. The findings suggest that the SPQR algorithm can produce outcomes comparable to those achieved using PCA without altering the original dataset.

Keywords: data analysis, PCA, SPQR, FCM

1 Introduction

When dealing with natural or artificial systems, relationships between their inputs and outputs can be represented in physical, mathematical, or logical ways. These relationships create a link between the input and output data of the system. In some cases, the input to the system can be represented by random variables in a mathematical model, providing a quantitative representation of a natural phenomenon. These models represent an object, a real phenomenon, or a set of phenomena, such as a mathematical model of a physical, chemical, or biological system. While these models are often approximate representations of reality, they are useful for analysis and prognosis. Mathematical models are widely used across various scientific fields, utilizing tools ranging from combinatorics to infinitesimal calculus. For example, in many cases, differential equations provide a concise and intuitive description of phenomena.

To keep things simple, we will refer to the data that enter and exit the model as data input and data output, regardless of the type of characteristic transfer function (such as linear dynamic systems, etc.). In this particular case, we will treat the input and output data as discrete random variables:

$$p(x) = P(X = x).$$

A phenomenon that can be characterized by a random variable can be described in terms of its probability distribution and associated parameters, such as the expected value and variance. Variance is particularly important in the study of models and is commonly utilized in techniques such as Karhunen–Loeve transform, principal component analysis (PCA), and its variants. PCA is frequently used for dimensionality reduction, which involves reducing the number of variables used to describe a dataset to a smaller number of latent variables while minimizing the loss of information [1].

From this perspective, it can be argued that if an original dataset and a reduced dataset produced using PCA are provided as inputs to a model, the resulting outputs should be almost identical. To test this hypothesis, the current study utilized a widely used method (PCA) as well as a relatively new algorithm (semi-pivoted QR [SPQR] approximation) [2,3] to reduce the dimensionality of various publicly available databases [4]. To evaluate the methods under different conditions, these datasets are very different from each other in terms of both the number of features and instances.

* **Corresponding author: Vincenzo Di Lecce,** Department of Electrical and Information Engineering Politecnico di Bari, Bari, Italy, e-mail: vincenzo.dilecce@poliba.it
Alberto Amato: Department of Electrical and Information Engineering Politecnico di Bari, Bari, Italy, e-mail: a.amato@poliba.it
ORCID: Alberto Amato 0000-0002-8107-7047; Vincenzo Di Lecce 0000-0002-4878-185X

The study employed fuzzy clustering and silhouette analysis to compare the results obtained from the different methods. This choice is because the used databases have not been classified a priori, and therefore, there is no ground truth available for evaluating the results obtained by the classification algorithms. The performances were compared to each other, revealing significant variations among the techniques and emphasizing the significant impact that preprocessing methods can have on the performance of machine learning algorithms.

The subsequent analysis evaluates the information loss that occurs when using the well-known PCA method compared to the SPQR algorithm. This evaluation was performed on four public databases obtained from the University of California, Irvine site [4], and for each database, the following procedure was executed according to the workflow in Figure 1:

- the raw data (namely, the original data stored in each database),
- the dataset was reduced using the PCA algorithm, loosing less than 2% of the total variation in the dataset,
- the dataset was reduced using the SPQR algorithm,
- the raw data normalized between 0 and 1,
- the normalized dataset was reduced using the PCA algorithm, loosing less than 2% of the total variation in the dataset, and
- the normalized dataset was reduced using the SPQR algorithm.

To the best of our knowledge, in the literature, few authors have used the SPQR algorithm for feature selection [3] as a preprocessing step in machine learning. An interesting comparative evaluation of the performance of this algorithm can be found in the study of Boutsidis et al. [5]. The results obtained in this work show that the performances of this algorithm in this kind of application are very good, and they are comparable to those obtained using PCA. This is a relevant result because SPQR is a feature selection method, and so it does not modify the original data (while PCA does it).

The remaining part of the article is organized as follows: Section 2 reports a brief overview of related works, Section 3 describes the four databases used for the tests, and Sections 4 and 5 describe, respectively, clustering and silhouette algorithms. In Section 6, SPQR is described, while in Section 7, the PCA algorithm is reported. Experiments and results are reported in Section 8, and conclusions and final remarks are in Section 9.

2 Related works

Modern applications generate vast amounts of data, which may contain irrelevant information. Therefore, the conversion of raw data into valuable insights is a highly relevant topic of research [6]. Many machine learning algorithms aim to increase knowledge density by reducing data dimensionality without losing important information. To enhance the computational efficiency of machine learning algorithms, preprocessing techniques are commonly employed to filter the data. This situation is not novel and has been known in statistics since Karl Pearson introduced PCA in 1901.

PCA is a widely used statistical technique that reduces the dimensionality of a dataset by projecting it onto a new axis system. This makes it easier to visualize multidimensional data. However, such techniques can significantly affect the performance of learning algorithms because they operate in a different space than the original data. Therefore, unseeingly applying these algorithms may not be advisable. Other, less well-known algorithms should be considered, depending on the specific context and data. Before applying any dimensionality reduction technique, it is essential to carefully analyze the data and context. It is also crucial to separate the re-projection of the data onto

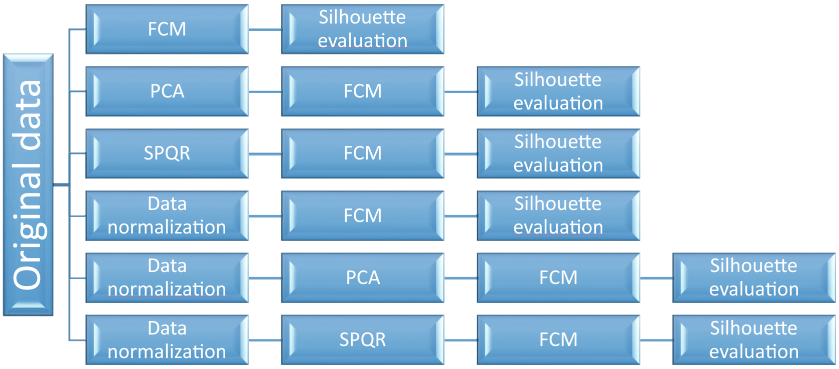

Figure 1: Experimental workflow.

new axes from the evaluation of the amount of information present in each dimension.

In the field of spatio-temporal series dimensionality reduction, the scientific research over the last 15 years has consistently focused on three distinct categories.

2.1 Methods based on statistics and information theory

This family of methods focuses on reducing input data based on statistical or information-theoretic criteria. Information-theoretic methods can be viewed as a generalization of statistical methods, as they can capture nonlinear relationships between variables, handle interval and categorical variables simultaneously, and many are invariant to monotonic transformations of input variables. The most well-known algorithm in this family is PCA, which involves finding orthogonal directions that explain as much of the variance in the data as possible.

To address nonlinear relationships, several dimensionality reduction techniques have been introduced, such as Isomap [7], locally linear embedding (LLE) [8], Hessian LLE [9], Laplacian eigen-maps [10], and their variants [11], including kernel PCA [12]. These methods reveal the inherent geometric structure of high-dimensional data. Giraud [13] demonstrated that high-dimensional spaces tend to be sparse and suffer from distance concentration.

2.2 Dictionary-based methods

Another approach for dimensionality reduction is based on the decomposition of a matrix consisting of all input data as columns. The input data matrix, using the input variables, is transformed into a new data matrix using new variables, which are obtained through a simple linear variation between the two sets of variables. This transformation is expressed through a matrix called a dictionary, which consists of atoms, and there are various ways to create such a dictionary [14]. Methods like singular value decomposition (SVD) and vector quantization (*K*-means) belong to this category, with *K*-means being an extreme case of a dictionary-based algorithm, where input vectors are represented by a single atom instead of a combination of atoms.

While parameter aggregation methods facilitate the updating of Pearson's correlation and/or covariance [15], SVD focuses on computational efficiency when all data are available. Another possibility is offered by NDR (nonlinear dimension reduction), which allows the SVD results to be used to better organize groups (dictionary). Furthermore, if information on such relationships is available, the generic organizational structure proposed here allows this information to be incorporated in the first step. However, this process requires several steps.

2.3 Projection-based methods

In this particular family of algorithms, the task of dimensionality reduction is framed as a projection of the initial data onto a subspace with specific properties [16]. The projection search involves identifying the output subspace by seeking out "interesting" directions. The definition of "interesting" is dependent on the particular problem being addressed, but generally, interesting directions are those where the projection values exhibit non-Gaussian behavior. Projection Pursuit is an approach that seeks out directions that maximize the deviation from the normal distribution (kurtosis) of the projected values as a measure of non-Gaussianity.

Similar to NDR, another algorithm called SPQR is an efficient deterministic method for reducing a given matrix to its most important columns. SPQR, which stands for SPQR approximation, was introduced by Stewart [2,17,18].

This article evaluates this algorithm for its suitability in reducing the dimensionality of data for machine learning applications. As the name suggests, the approach is based on QR decomposition, which expresses matrix A as the product of an orthogonal matrix Q and an upper triangular matrix R. These factors are obtained by orthonormalizing the columns of A one by one using the Gram–Schmidt algorithm from the first to the last. This procedure is preferred because it uses the rotated QR, which differs because the Gram–Schmidt procedure takes the largest column left at the beginning of each new step [19]. The new step is then taken, and a permutation matrix P is created such that:

$$A - P = Q - R.$$

It is important to note that, for this article, there is no re-projection of the data into a new space, and there is no loss of significance in the results. The user only sees the columns of their data structure reordered according to decreasing significance.

3 Database description

All the experiments carried out in this work have used public databases downloaded from the study of Dua and Graff [4]. In particular, the used databases are as follows:

1. Gender Gap in Spanish WP Dataset [20]: Dataset used to estimate the number of women editors and their editing practices in the Spanish Wikipedia. It is composed of 21 attributes and 4,746 instances.
2. TUANDROMD (Tezpur University Android Malware Dataset) Dataset [21]: this dataset contains 4,465 instances and 241 attributes. The target attribute for classification is a category (malware vs goodware).
3. Room Occupancy Estimation Dataset [22]: this dataset contains 10,129 instances and 16 attributes, and it is used to estimate the occupation level of the room. The setup consisted of seven sensor nodes and one edge node in a star configuration, with the sensor nodes transmitting data to the edge every 30 s using wireless transceivers. Each sensor node contains various sensors such as temperature, light, sound, $CO_{2,}$ and digital passive infrared.
4. Myocardial Infarction Complications Dataset [23]: this dataset contains 1,700 instances and 124 attributes. The main application of this database is to predict complications of Myocardial Infarction based on information about the patient at the time of admission and on the third day of the hospital period.

In all the experiments, for each database, only the numerical features have been considered.

4 Clustering

Clustering algorithms typically fall under unsupervised learning techniques as they do not use labeled data to group objects [24]. However, there are also semi-supervised clustering algorithms that incorporate some labeled data in the clustering process [25]. A variety of algorithms fall under the category of clustering, including hierarchical, partitional, grid, density, and model-based techniques, all of which aim to group objects based on a defined similarity criterion [24].

In this work, the authors employed the fuzzy *C*-means (FCM) algorithm [26,27], which generates fuzzy partitions and prototypes for numerical datasets by optimizing a generalized least-squares objective function:

$$Q = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m} \, \|\boldsymbol{X}_k - \boldsymbol{V}_i\|^2, \tag{1}$$

where $\| \because \|$ is a distance function such as: Euclidean, Mahalanobis, etc.; $\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_c}$ are the centroids of the clusters also called prototypes; $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}\}$ is the set of the points to be clustered; $U = [u_{ik}]$ is the partition matrix; c is the number of clusters; N is the number of points to be clustered; i is an index that varies from 1 to c; k is an index that varies from 1 to N; "m" is a coefficient called "fuzzification coefficient." It is greater than 1, and it is responsible for the level of "fuzziness" of the partition matrix. In other words, it controls the level of fuzziness with which each point belongs to the various clusters.

This study uses this algorithm to generate fuzzy partitions and prototypes for a set of numerical data. It optimizes a generalized least-squares objective function, minimizing it with respect to the given prototypes and partition matrix $\boldsymbol{U}$ to discover the internal structure of the dataset $\boldsymbol{X}$. The minimization process is iterative and involves updating the partition matrix and prototypes until a stopping criterion is met.

As an example, during the iterative process of updating the partition matrix and prototypes in FCM algorithm, the procedure may stop when the quantity

$$\|U - U'\| = \max_{i,k}|u_{ik} - u'_{ik}|, \tag{2}$$

which measures the difference between two consecutive partitions matrices U and U', becomes smaller than a predefined positive threshold ε.

It is important to note that the optimization function used in the FCM algorithm results in a solution that reflects the geometry of the input dataset to some extent.

On the other hand, the clustering algorithm can also be used to classify new elements added to the dataset after the initial clustering based on the hidden structure discovered by the algorithm. This task can be accomplished by applying the following rules:

- The "anchor points" of the classificator are the prototypes of the clusters;
- Each cluster defines a class;
- A point x belongs to a class defined by the cluster with prototype v_j if:

$$j = \arg\left(\min_i \|x - v_i\|^2\right). \tag{3}$$

5 Silhouette method for clustering evaluation

Evaluating the performance of a clustering algorithm is not an easy task, as it can lead to one of the following scenarios:

- The correct solution is known: in this case, the performance evaluation of a clustering algorithm involves computing the number of misclassified patterns or error rates. In this case, the classification of each point is

known a priori, and the clustering algorithm's performance can be measured based on how many patterns are incorrectly classified.
- The correct solution is subjective: in this case, there is no ground truth against which to evaluate the results of the clustering algorithm. As the classification is subjective, there is no universally acceptable solution, and the classification task falls into the semantic gap problem [28].
- The correct solution is unknown: in this case, evaluating the performance of a clustering algorithm can be challenging because there is no ground truth against which to evaluate the results of the clustering algorithm. Various approaches have been proposed to address this issue, as described in the study by Rand [29]. One commonly used method in recent years is the silhouette parameter, which measures the quality of clusters [30].

In this work, the authors utilized the silhouette parameter to evaluate the performance of their clustering algorithm. This evaluation method compares the similarity levels of each object within its own cluster (tightness) and with objects in other clusters (separation). The silhouette parameter is defined as follows: for a point y that belongs to cluster $\boldsymbol{A}$, the mean distance between $\boldsymbol{y}$ and all other points of $\boldsymbol{A}$ is computed and denoted as $\boldsymbol{t(y)}$. Then, for any cluster $\boldsymbol{B}$ different from $\boldsymbol{A}$, the average distance between $\boldsymbol{y}$ and all points of $\boldsymbol{B}$ ($\boldsymbol{d(y,B)}$) is calculated. After calculating the distance between point $\boldsymbol{y}$ and each cluster $\boldsymbol{B}$ where $\boldsymbol{B} \neq \boldsymbol{A}$, we choose the minimum value among all these distances and denote it by

$$\boldsymbol{v}(\boldsymbol{y}) = \min_{\boldsymbol{A}\neq\boldsymbol{B}} \boldsymbol{d}(\boldsymbol{y}, \boldsymbol{B}). \tag{4}$$

Starting from these considerations, the silhouette for the point y is defined as shown in the following formula:

$$s(y) = \begin{cases} 1 - \dfrac{t(y)}{v(y)}, & \text{if } t(y) < v(y) \\ 0, & \text{if } t(y) = v(y) \\ \dfrac{v(y)}{t(y)} - 1, & \text{if } t(y) > v(y). \end{cases} \tag{5}$$

From this definition, it is possible to say that for each point x in the dataset: $-1 < s(y) < +1$

An in-depth analysis of this parameter is reported in the study of Rousseeuw [30].

On the other hand, utilizing the silhouette parameter method allows us to present the clustering results through a visual representation that showcases how well each point has been classified. Additionally, it is worth noting that this method is flexible since it allows the use of various distance metrics such as Mahalanobis, Euclidean, and Manhattan, among others.

All these features give the silhouette parameters useful for evaluating clustering algorithm performance on datasets with no prior knowledge (that is also the case study of this article).

The silhouette method can also be used as a reference guide to determine the optimal number of clusters for a dataset. This can be achieved through an iterative process involving the following steps:
1. Run the clustering algorithm using a certain number of clusters "C."
2. Compute the silhouette for the obtained clusters.
3. If there is a satisfying number of points with a good level of silhouette then "C" can be considered a good number of clusters; else, change the value of "C" and return to step 1.

6 SPQR algorithm

Numerous data applications require the representation of $\boldsymbol{m}$ entities with $\boldsymbol{n}$ attributes. A frequently utilized approach to represent this data is to create a matrix $\boldsymbol{A}$ that has $\boldsymbol{m}$ rows and $\boldsymbol{n}$ columns. However, in modern applications of data analysis like environmental datasets and image analysis, these matrices often possess a high number of dimensions, resulting in complications in data mining, representation, storage, and communication.

Over the past few years, numerous studies in the area of feature selection [31,32] have shown that it is feasible to detect and remove redundant or irrelevant features while analyzing a dataset. By utilizing feature selection techniques, several benefits can be achieved in the data analysis process, including reduced data size, enhanced prediction accuracy, identification of critical features, easier comprehension of attributes or variables, and reduced execution time [31]. An informative paper providing an overview of feature selection methods can be found in the study of Venkatesh and Anuradha [31].

Many methods employed in data analysis aim to approximate matrix $\boldsymbol{A}$ by utilizing a "smaller" matrix created by combining its rows and columns. However, these techniques typically result in dense factorizations that are much harder to comprehend than the original terms. For instance, truncating the SVD at $\boldsymbol{k}$ terms is a prevalent approach for obtaining the "best" ***rank-k*** approximation of $\boldsymbol{A}$ regarding any unitarily invariant matrix norm. Nonetheless, this approach produces a representation of the dataset that is challenging to relate to the original dataset and the processes that generated it. A similar issue is present in another commonly used technique for feature selection, the PCA.

Numerous techniques have been developed to address the column subset selection problem based on the issues [33]. These methods aim to identify the subset of $\boldsymbol{k}$ original columns from $\boldsymbol{A}$, where $\boldsymbol{k}$ is less than $\boldsymbol{n}$, that contains the majority of the information of $\boldsymbol{A}$, with respect to the spectral or the Frobenius norm.

Generally, there are two categories of methods that can be defined:

1. Randomized methods: These techniques utilize probability distributions to select the most representative columns in a matrix.
2. Deterministic methods: These approaches select columns in a deterministic manner.

One effective deterministic technique for reducing matrix $\boldsymbol{A}$ to its most important columns is the SPQR approximation, developed by Stewart [2,17,18]. The central approach in this method is to compute the QR decomposition of $\boldsymbol{A}$, where $\boldsymbol{A}$ is decomposed into an orthogonal matrix $\boldsymbol{Q}$ and a triangular matrix $\boldsymbol{R}$. This factorization is performed using the Gram–Schmidt algorithm to orthonormalize the columns of $\boldsymbol{A}$ sequentially, from first to last. In many cases, the pivoted QR method is preferred, where columns are exchanged at the start of each new stage to select the largest remaining column. In this way, a permutation matrix $\boldsymbol{P}$ is built such that

$$\boldsymbol{A} \cdot \boldsymbol{P} = \boldsymbol{Q} \cdot \boldsymbol{R}. \tag{6}$$

In cases where $\boldsymbol{A}$ is rank deficient, column pivoting is applied by using the matrix $\boldsymbol{A{\cdot}P}$ to enhance the numerical accuracy. Additionally, the selection of $\boldsymbol{P}$ ensures that the diagonal entries of $\boldsymbol{R}$ do not increase, which is a beneficial property for subsequent steps. More in detail, we can partition the aforementioned expression as follows:

$$[\boldsymbol{B}_1 \quad \boldsymbol{B}_2] = [\boldsymbol{Q}_1 \quad \boldsymbol{Q}_2]\begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ 0 & \boldsymbol{R}_{22} \end{bmatrix}, \tag{7}$$

and the following properties hold:

1. $\boldsymbol{B}_1 = \boldsymbol{Q}_1 \cdot \boldsymbol{R}_{11,}$
2. $||\boldsymbol{B}_2 - \boldsymbol{Q}_1 \cdot R_{12}|| = ||R_{22}||.$

The *semi-QR* algorithm exploits these results to use the approximation

$$\boldsymbol{A} \cdot \boldsymbol{P} \approx \boldsymbol{Q}_1 \cdot [\boldsymbol{R}_{11}\, \boldsymbol{R}_{12}], \tag{8}$$

that, thanks to property 1, reproduces the first $\boldsymbol{k}$ columns of $\boldsymbol{A{\cdot}P}$ exactly, by introducing a quantifiable error (property 2). Another advantage of the SPQR method is that it does not require the explicit computation of the non-sparse orthogonal matrix $\boldsymbol{Q}_1$. In practice, the SPQR algorithm provides the $\boldsymbol{k}$ columns of $\boldsymbol{A}$ whose span approximate the column space of A, given a rank parameter $\boldsymbol{k}$. These $\boldsymbol{k}$ columns form matrix $\boldsymbol{B1}$ of size $\boldsymbol{m} \times \boldsymbol{k}$, while the factor $\boldsymbol{R}_{11}$ contains the coefficients of the column orthogonalization.

7 PCA

PCA is a widely used method for a dimensionality reduction in large databases. While reducing the dimensions of a dataset can lead to a loss of accuracy, it can also improve the efficiency of data analysis algorithms, such as data exploration, data visualization, and machine learning. Therefore, when using PCA, it is necessary to find a balance between the benefits of performance improvement and the potential drawbacks of accuracy loss.

In this section, we will provide a brief operational description of the PCA method, while a more detailed analysis can be found in the study of Jolliffe [34].

PCA is a process that can be divided into five steps. The first step is standardization, which involves standardizing the range of each initial variable so that they contribute equally to the analysis. The second step is computing the covariance matrix to determine the degree of relationship among the variables. In the third step, principal components are identified by computing eigenvectors and eigenvalues of the covariance matrix. These new variables are uncorrelated and contain most of the information of the initial variables. The fourth step involves selecting feature vectors by sorting the eigenvectors in descending order by their eigenvalues, allowing for the identification of the most significant principal components. Finally, the fifth step involves recasting the data along the principal component axes.

8 Experiments and results

In this section, a brief description of the carried-on experiments is reported. For each database described in Section 3 a clustering analysis using the FCM algorithm has been conducted varying the number of clusters. The results of each clustering have been evaluated using the silhouette parameter. These analyses have been carried-on six times using the workflow reported in Figure 1:

1. the raw data (namely, the original data stored in each database),

2. the dataset reduced using the PCA algorithm loosing less than 2% of the total variation in the dataset,
3. the dataset reduced using the SPQR algorithm,
4. the raw data normalized between 0 and 1,
5. the normalized dataset reduced using the PCA algorithm loosing less than 2% of the total variation in the dataset, and
6. the normalized dataset reduced using the SPQR algorithm.

In the following, the obtained results for each database are reported.

8.1 Gender gap in Spanish WP dataset

8.1.1 Clustering and silhouette using raw data

The original dataset is composed of 20 numerical features. Applying PCA to this dataset loosing less than 2% of the total variation in the data produces a dataset with a single dimension.

Figures 2–4 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50. Each line represents the percentage of points with silhouette greater than a given threshold that is 0.7 for the blue line, 0.8 for the red line, and 0.9 for the yellow line.

8.1.2 Clustering and silhouette using normalized data

The original dataset has been normalized between 0 and 1. Applying PCA to this dataset loosing less than 2% of the total variation in the data produces a dataset with six dimensions.

Figures 5–7 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50. Each line represents the percentage of points with silhouette greater than a given threshold that is 0.7 for the blue line, 0.8 for the red line, and 0.9 for the yellow line. In these experiments, the results obtained using normalized data seem to be worse than that obtained on the raw data.

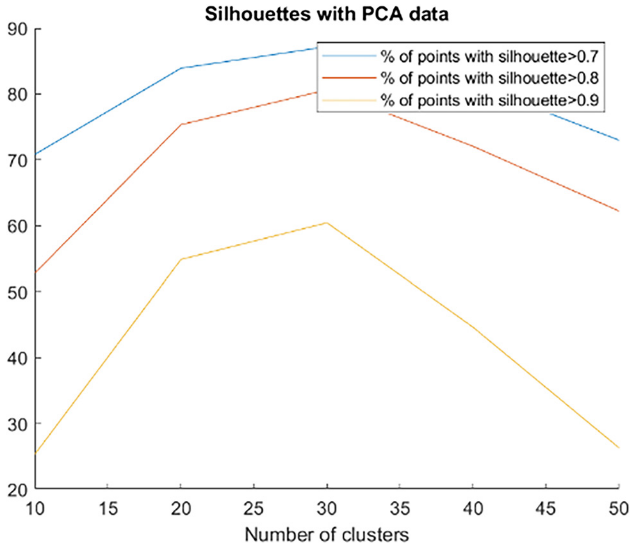

Figure 3: Clustering performance in terms of silhouette using the reduced dataset with PCA.

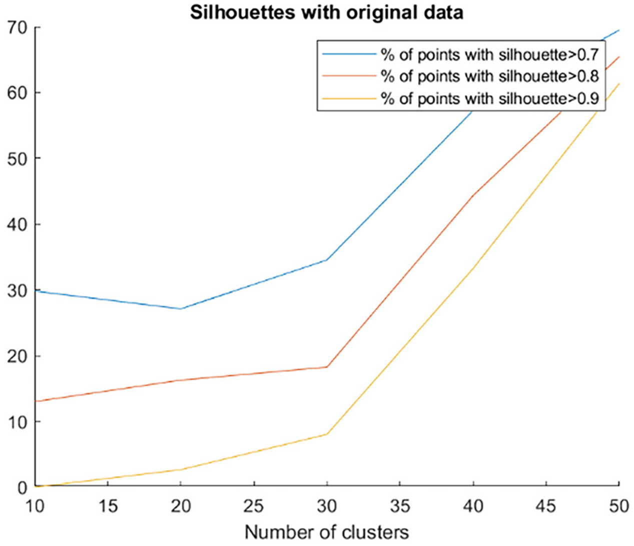

Figure 2: Clustering performance in terms of silhouette using original data.

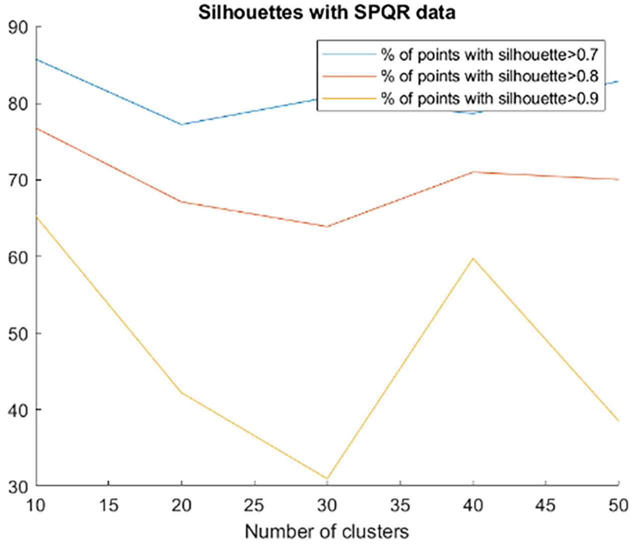

Figure 4: Clustering performance in terms of silhouette using the reduced dataset with SPQR.

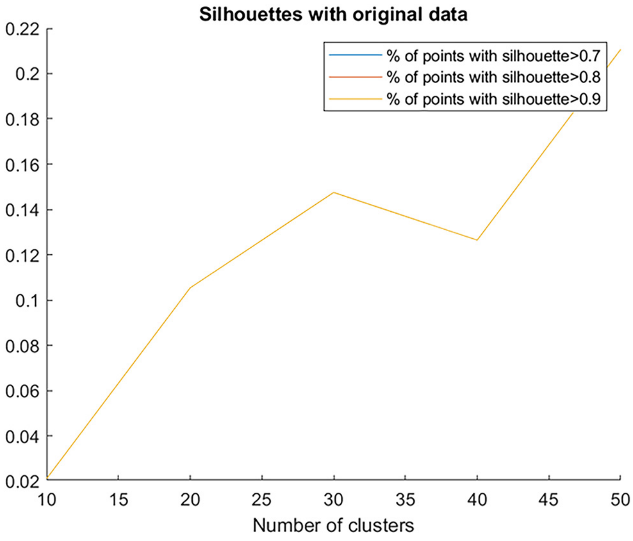

Figure 5: Clustering performance in terms of silhouette using normalized data.

8.2 Room occupancy estimation dataset

8.2.1 Clustering and silhouette using raw data

The original dataset is composed of 20 numerical features. Applying PCA to this dataset loosing less than 2% of the total variation in the data produces a dataset with two dimensions.

Figures 8–10 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50. Each line represents the percentage of points with silhouette greater than a given threshold that is 0.7 for the blue line, 0.8 for the red line, and 0.9 for the yellow line.

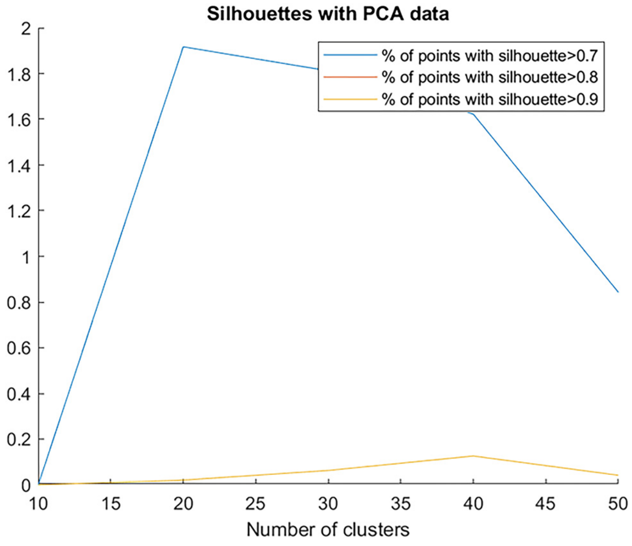

Figure 6: Clustering performance in terms of silhouette using the reduced dataset with PCA applied to normalized data.

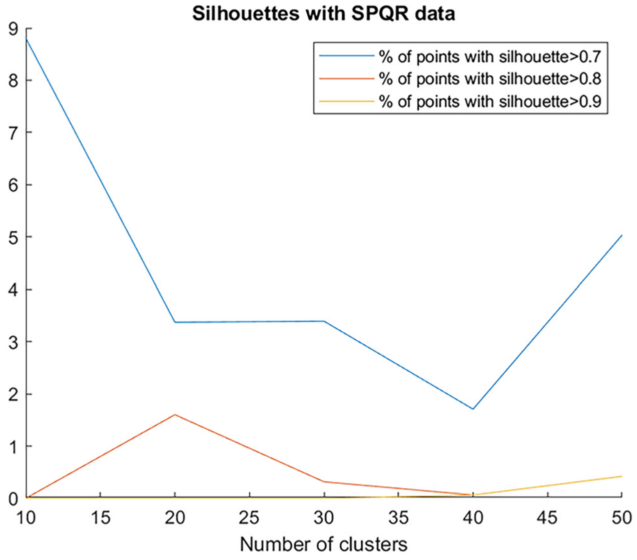

Figure 7: Clustering performance in terms of silhouette using the reduced dataset with SPQR applied to normalized data.

8.2.2 Clustering and silhouette using normalized data

The original dataset has been normalized between 0 and 1. Applying PCA to this dataset loosing less than 2% of the total variation in the data produces a dataset with five dimensions.

Figures 11–13 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50. Each line represents the percentage of points with silhouette greater than a given threshold that is 0.7 for the blue line, 0.8 for the red line and 0.9 for the yellow line. In these

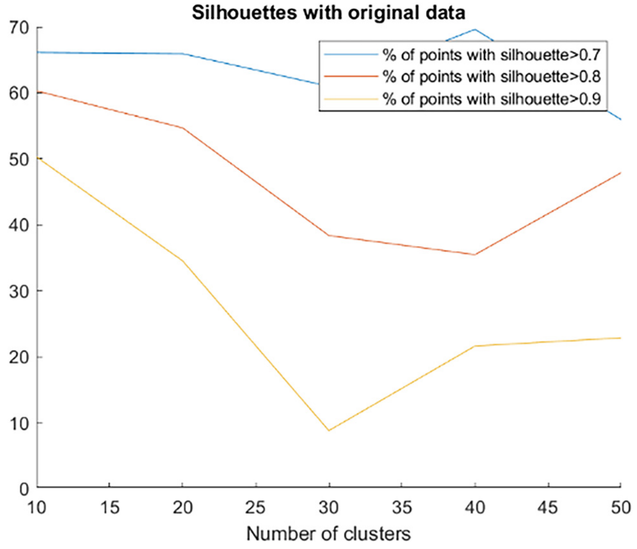

Figure 8: Clustering performance in terms of silhouette using original data.

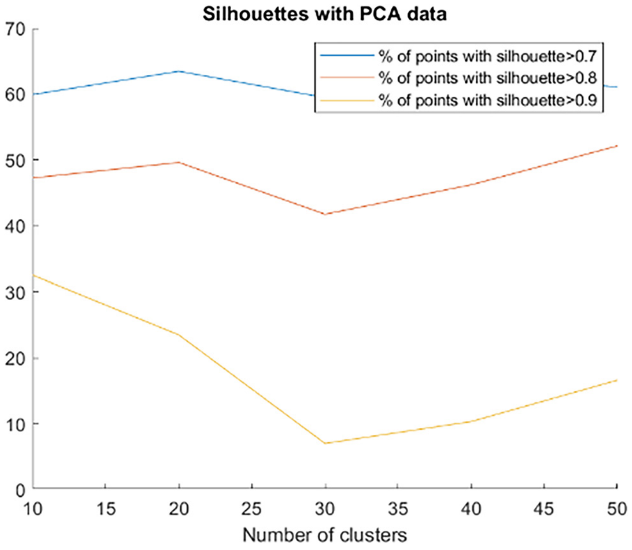

Figure 9: Clustering performance in terms of silhouette using the reduced dataset with PCA.

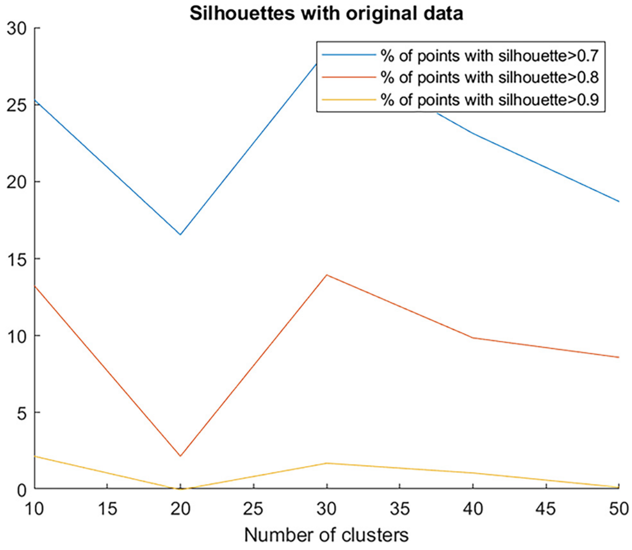

Figure 11: Clustering performance in terms of silhouette using normalized data.

experiments, the results obtained using normalized data seem to be slightly worse than that obtained on the raw data.

8.3 Myocardial infarction complications Dataset

8.3.1 Clustering and silhouette using raw data

This dataset could be considered a sort of spare matrix. There are many 0, and some features contain not a number values (NaN). The proposed results have been obtained selecting only the features without NaN values obtaining a dataset composed of 13 dimensions. Applying PCA to this dataset loosing less than 2% of the total variation in the data produces a dataset with six dimensions.

Figures 14–16 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50. Each line represents the percentage of points with silhouette greater than a given threshold that is 0.7 for the blue line, 0.8 for the red line, and 0.9 for the yellow line.

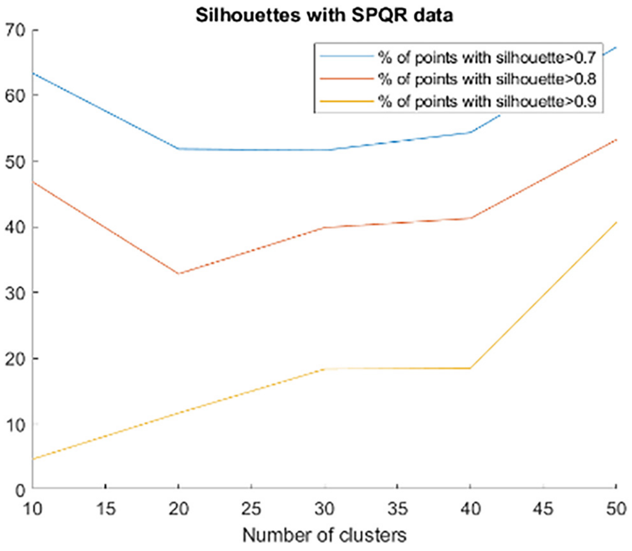

Figure 10: Clustering performance in terms of silhouette using the reduced dataset with SPQR.

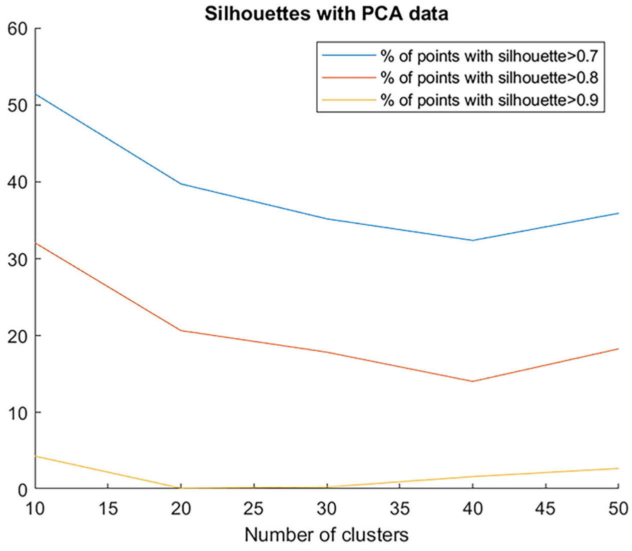

Figure 12: Clustering performance in terms of silhouette using the reduced dataset with PCA applied to normalized data.

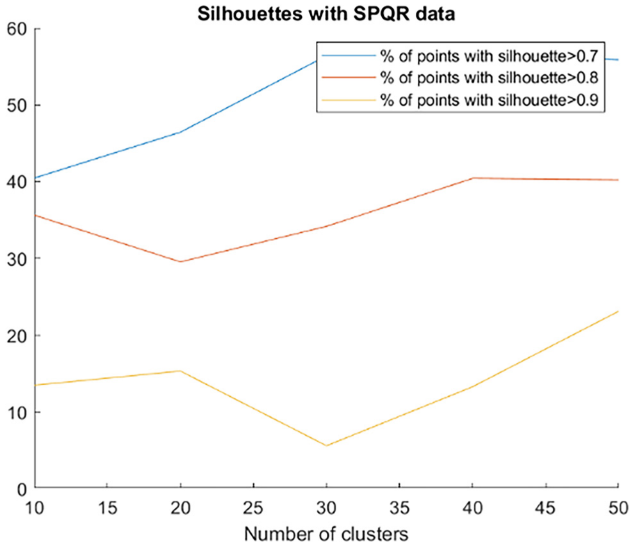

Figure 13: Clustering performance in terms of silhouette using the reduced dataset with SPQR applied to normalized data.

8.3.2 Clustering and silhouette using normalized data

The original dataset (without the features containing NaN values) has been normalized between 0 and 1. Applying PCA to this dataset loosing less than 2% of the total variation in the data produces a dataset with ten dimensions.

Figures 17–19 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50. Each line represents the percentage of points with silhouette greater than a given threshold that is 0.7 for the blue line, 0.8 for the red line, and 0.9 for the yellow line. In these experiments, the results obtained using normalized data seem to be worse than that obtained on the raw data.

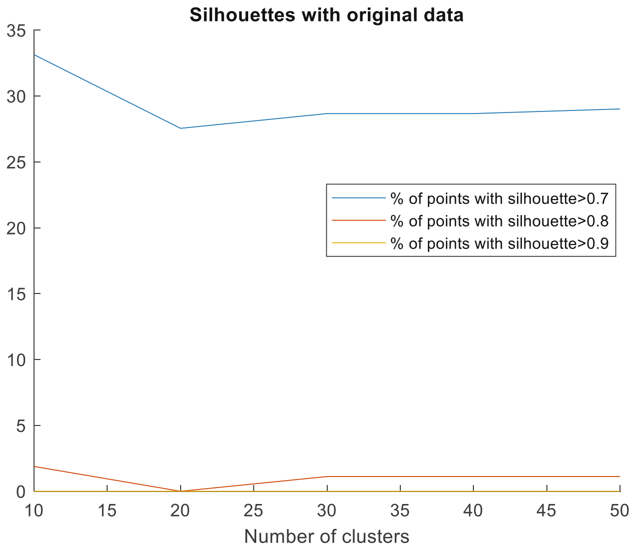

Figure 14: Clustering performance in terms of silhouette using original data.

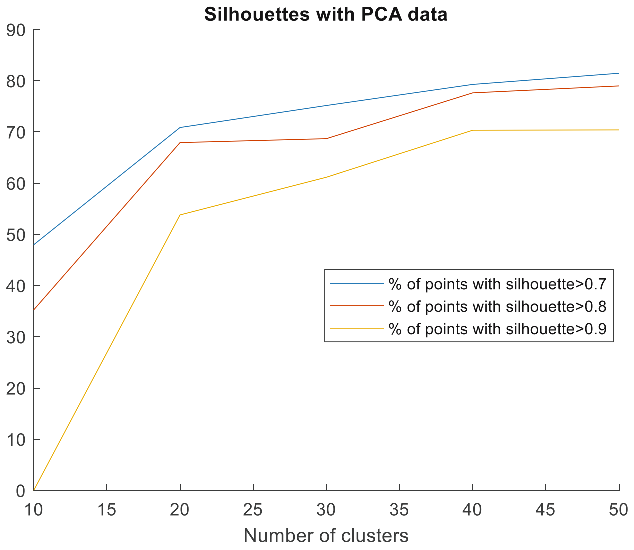

Figure 15: Clustering performance in terms of silhouette using the reduced dataset with PCA.

8.4 TUANDROMD

8.4.1 Clustering and silhouette using raw data

This dataset could be considered a sort of spare matrix. There are many 0, but there are not features containing NaN. Applying PCA to this dataset loosing less than 2% of the total variation in the data produces a dataset with seven dimensions.

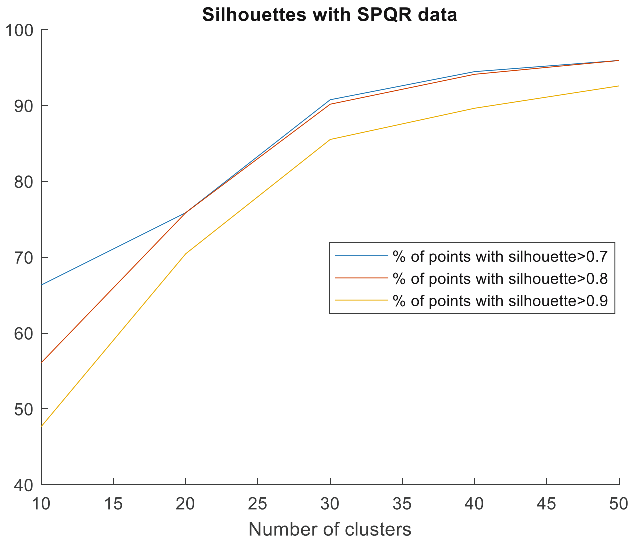

Figure 16: Clustering performance in terms of silhouette using the reduced dataset with SPQR.

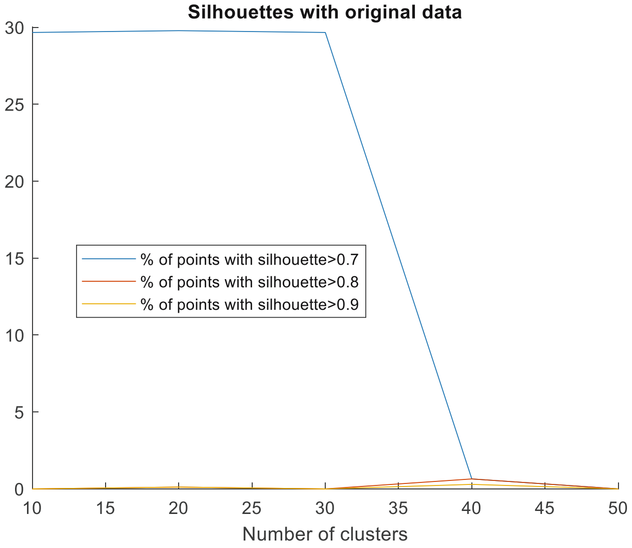

Figure 17: Clustering performance in terms of silhouette using normalized data.

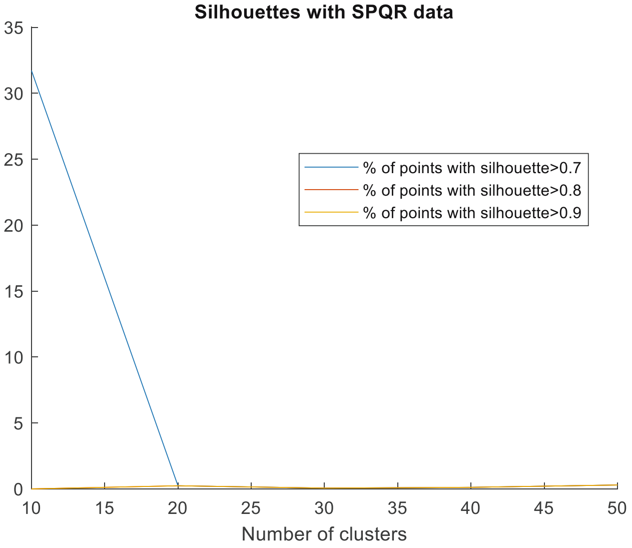

Figure 19: Clustering performance in terms of silhouette using the reduced dataset with SPQR applied to normalized data.

Figures 20–22 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50. Each line represents the percentage of points with silhouette greater than a given threshold that is 0.7 for the blue line, 0.8 for the red line, and 0.9 for the yellow line.

8.4.2 Clustering and silhouette using normalized data

The original dataset has been normalized between 0 and 1. Applying PCA to this dataset loosing less than 2% of the total variation in the data produces a dataset with seven dimensions.

Figures 23–25 show the obtained results in terms of silhouettes varying the number of clusters from 10 to 50. Each line represents the percentage of points with silhouette greater than a given threshold that is 0.7 for the blue line, 0.8 for the red line, and 0.9 for the yellow line. In these experiments, the results obtained using normalized data seem to be similar to those obtained using the raw data.

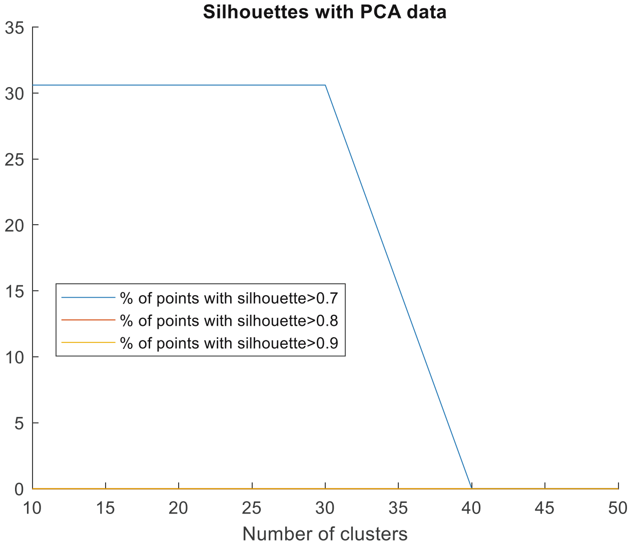

Figure 18: Clustering performance in terms of silhouette using the reduced dataset with PCA applied to normalized data.

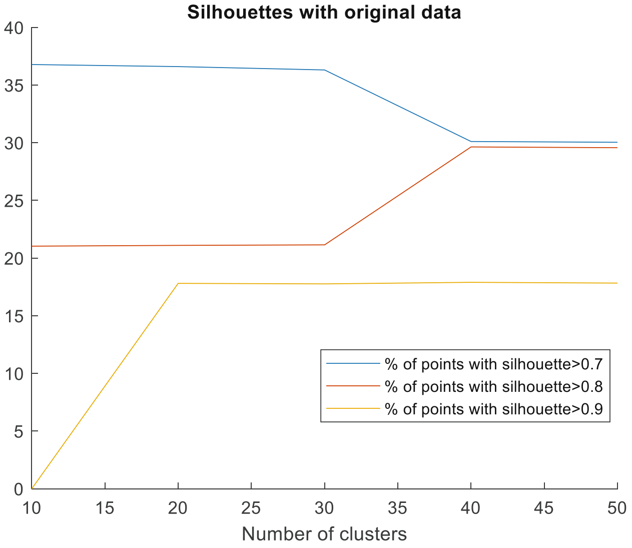

Figure 20: Clustering performance in terms of silhouette using original data.

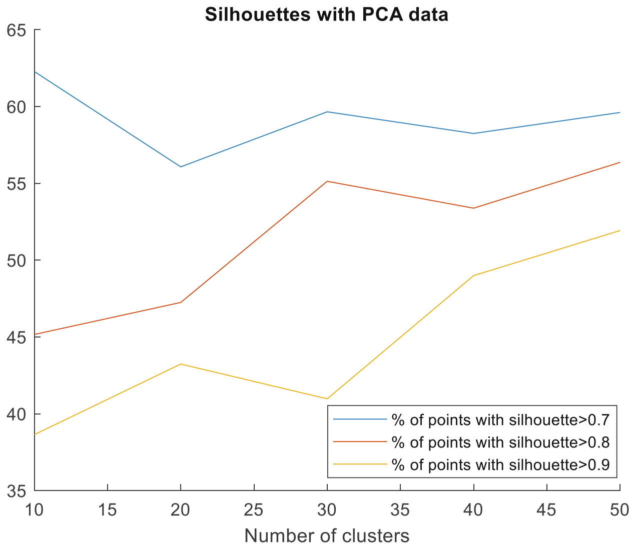

Figure 21: Clustering performance in terms of silhouette using the reduced dataset with PCA.

8.5 Synthesis

The mean values of all the results shown in the previous sections are reported in Figure 26. Figure 26 shows the mean percentage of points classified with a silhouette greater than 0.7 using the various methods, while Figures 27 and 28 report the mean percentage of points classified, respectively, with a silhouette greater than 0.8 and 0.9.

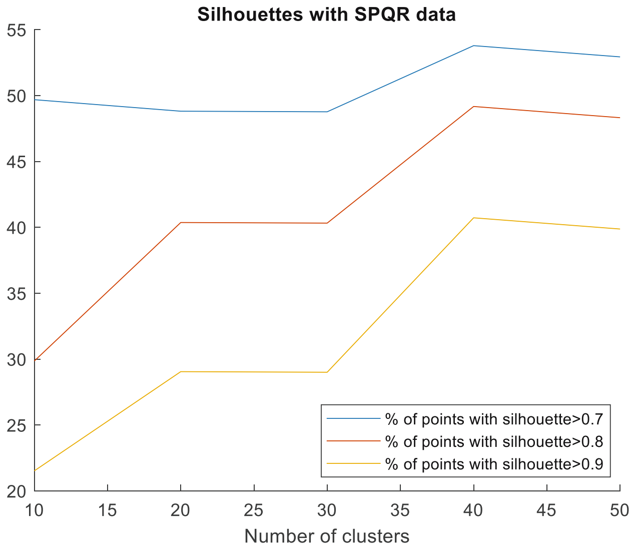

Figure 22: Clustering performance in terms of silhouette using the reduced dataset with SPQR.

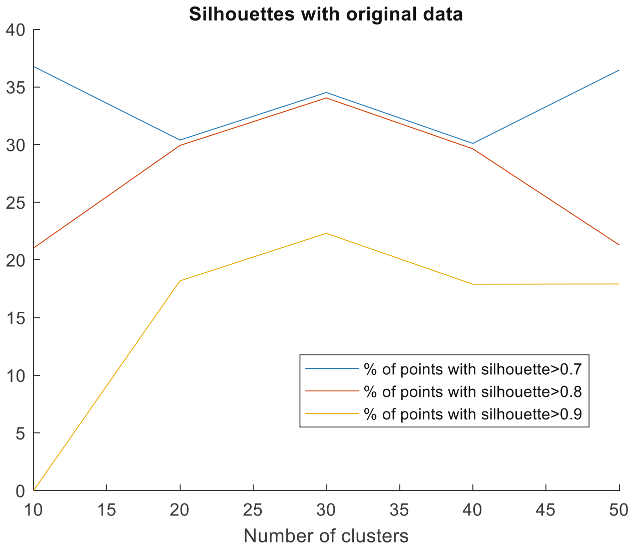

Figure 23: Clustering performance in terms of silhouette using normalized data.

9 Conclusions

The study aimed to evaluate a new data preprocessing technique called SPQR approximation for machine learning. To the best of our knowledge, the use of the SPQR algorithm as a preprocessing stage in machine learning applications is an original contribution of this article. The obtained results show that this preprocessing technique can improve both

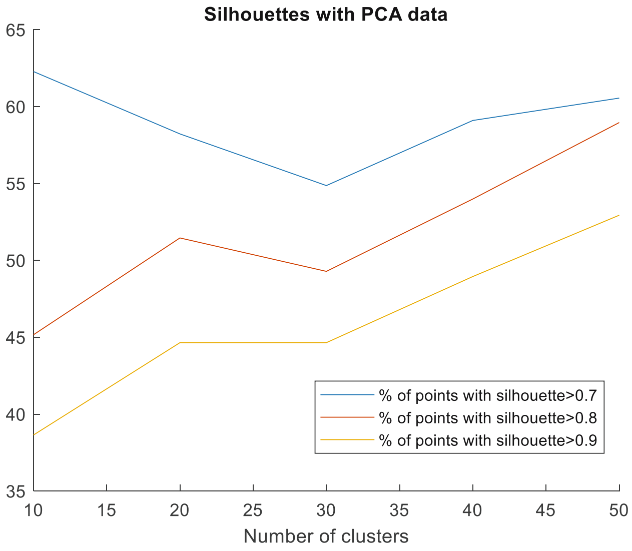

Figure 24: Clustering performance in terms of silhouette using the reduced dataset with PCA applied to normalized data.

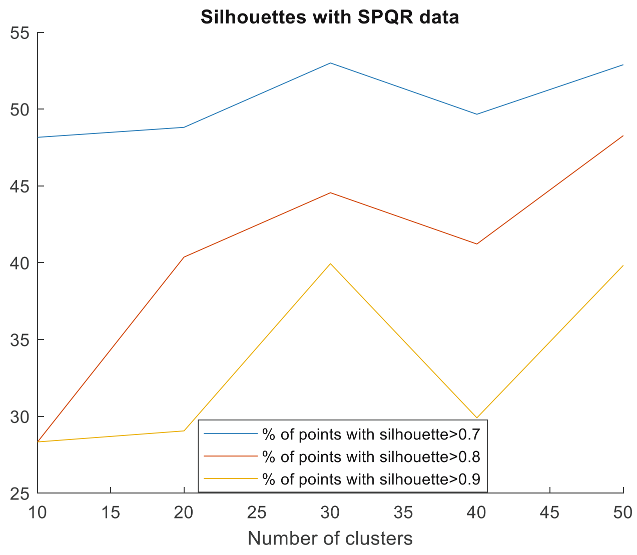

Figure 25: Clustering performance in terms of silhouette using the reduced dataset with SPQR applied to normalized data.

the classification performance and the computational efficiency of a machine learning algorithm. A point of strength of this method is that it belongs to the family of feature selection algorithms; hence, it does not modify the original data making it simple to add new data points for further classification tasks.

These results have been obtained using the FCM algorithm to clusterize data in four publicly available databases using different preprocessing techniques:

1. No preprocessing, where the raw data was directly used for clustering using FCM.
2. Data normalization, where each feature in the dataset was normalized.
3. PCA, which involved reducing the dimensionality of the dataset using PCA while retaining at least 98% of the total variation in the data.
4. SPQR, which also involved reducing the dimensionality of the dataset while retaining at least 98% of the total variation in the data, but unlike PCA, it did not modify the original dataset.

The results obtained by the FCM algorithm have been measured using the silhouette method. This evaluation method compares the similarity levels of each object within its own cluster (tightness) and with objects in other clusters (separation). To a certain extent, this method performs a geometrical evaluation of the structure of the obtained clusters. This approach is due to the fact that these datasets are not labeled, so there is no ground truth against which to evaluate the correctness of the clusters from a semantic point of view.

The previous sections display the obtained results, which enable us to draw certain considerations:

- The performance of clustering algorithms is affected by data normalization, which has an “equalization effect” on the shape of the feature space in a dataset. Normalizing a dataset ensures that each dimension in the feature space has an equal extension 1, giving each dimension equal weight in the distance function used by the clustering algorithm. However, from a semantic point of view, this may cause issues when analyzing a dataset with features of varying importance.

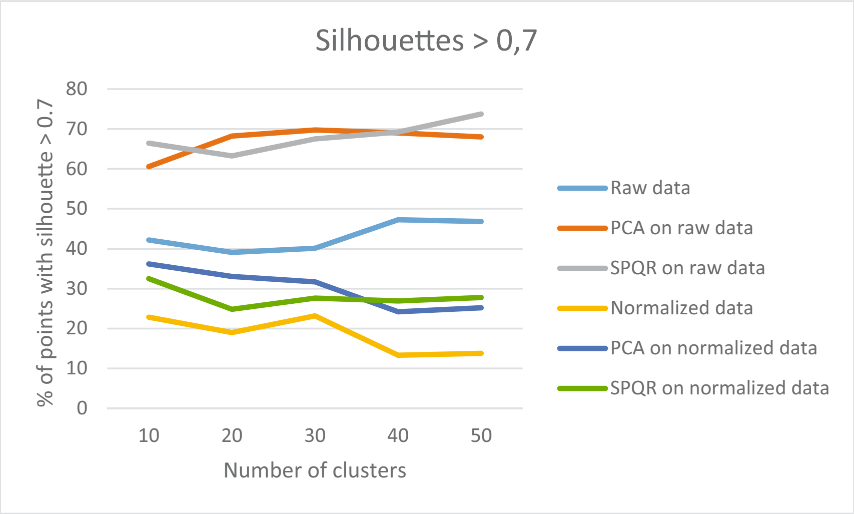

Figure 26: Mean results in terms of silhouettes >0.7 obtained by varying the number of clusters from 10 to 50.

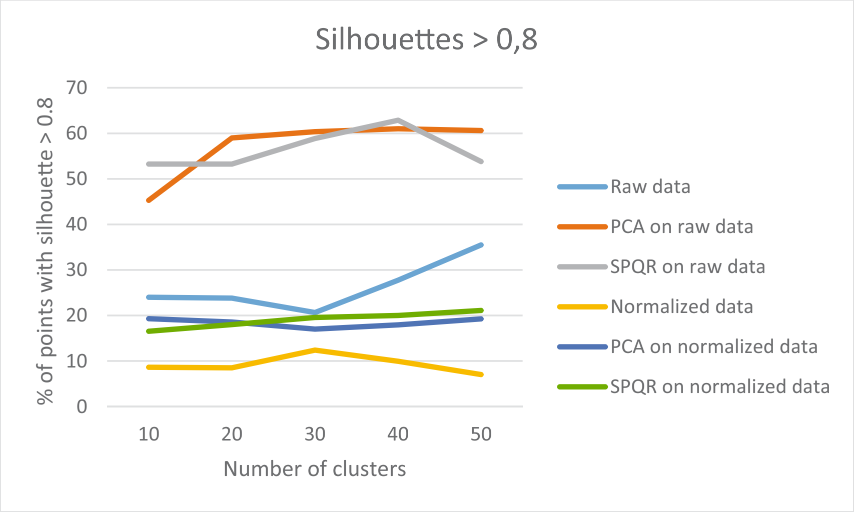

Figure 27: Mean results in terms of silhouettes >0.8 obtained by varying the number of clusters from 10 to 50.

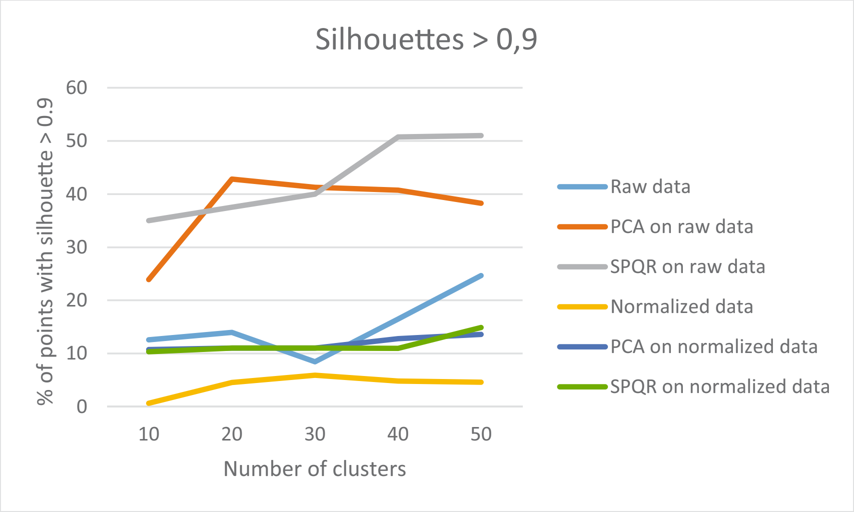

Figure 28: Mean results in terms of silhouettes >0.9 obtained by varying the number of clusters from 10 to 50.

- The performance of classification algorithms can be improved by reducing the dimensions of the dataset. This can result in improvements in both computational and classification performances, as observed in the experiments.
- In Section 7, it was demonstrated that reducing dimensions using PCA involves re-projecting the original feature space into a new one that is defined by the eigenvectors of the covariance matrix. This makes it challenging to determine the contribution of each feature in the original dataset to the classification process. Additionally, any new data points that are added to the original dataset cannot be classified without being transformed into the new feature space.
- The SPQR algorithm has been presented in Section VI. This algorithm does not modify the original dataset and only changes the position of some features within it. This eliminates the limitations of PCA mentioned earlier. Additionally, the results indicate that the performance achieved using the SPQR algorithm for data preprocessing is similar to that obtained using PCA.

Funding information: Authors state no funding involved.

Author contributions: A.A., V.D.L. contributed to the design and implementation of the research, the analysis of the results, and the writing of the manuscript.

Conflict of interest: Authors state no conflict of interest.

Ethical approval: Authors state no ethical approval is required.

Data availability statement: This evaluation was performed on public databases obtained from the UCI site [4].

References

[1] G. Tufféry, "Factor analysis," in *Data mining and statistics for decision making*, Wiley, 2011, pp. 175–180.
[2] G. W. Stewart, "Four algorithms for the efficient computation of truncated pivoted QR approximations to a sparse matrix," *Numer. Math.*, vol. 83, pp. 313–323, 1999.
[3] M. Popolizio, A. Amato, V. Piuri, and V. Di Lecce, "Improving Classification Performance Using The Semi-Pivoted QR approximation algorithm," In *2nd FICR International Conference on Rising Threats in Expert Applications and Solutions*. 7–8 January 2022.
[4] D. Dua and C. Graff, *UCI machine learning repository*. Irvine, CA: University of California, School of Information and Computer Science, 2019. http://archive.ics.uci.edu/ml
[5] C. Boutsidis, J. Sun, and N. Anerousis, "Clustered subset selection and its applications on it service metrics," *Proceedings of the 17th ACM conference on Information and knowledge management (CIKM '08)*. New York, NY, USA: Association for Computing Machinery, 2008, pp. 599–608. doi: 10.1145/1458082.1458162.
[6] A. Tăuţan, A. Rossi, R. de Francisco, and B. Ionescu, "Dimensionality reduction for EEG-based sleep stage detection: comparison of autoencoders, principal component analysis and factor analysis," *Biomed. Eng./Biomedizi Tech.*, vol. 66, no. 2, pp. 125–136, 2021. doi: 10.1515/bmt-2020-0139.
[7] M. Balasubramanian and E. L. Schwartz, "The isomap algorithm and topological stability," *Science*, vol. 295, no. 5552, p. 7, 2002.
[8] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[9] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Natl. Acad. Sci. U S A.*, 2003, vol. 100, no. 10, pp. 5591–5596.
[10] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput*, vol. 15, no. 6, pp. 1373–1396, 2003.
[11] H. Huang and H. Feng, "Gene classification using parameter-free semi-supervised manifold learning," *IEEE/ACM Trans. Comput. Biology, Bioinf.*, vol. 9, no. 3, pp. 818–827, May–Jun 2012.
[12] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge, UK: Cambridge University Press, 2004.
[13] C. Giraud, *Introduction to high-dimensional statistics*, vol. 138, Boca Raton, FL, USA: CRC Press, 2014.
[14] R. Rubinstein, M. Zibulevsky, and M. Elad, *Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. No. CS Technion report CS-2008-08*, Computer Science Department, Technion, 2008.
[15] R. A. Johnson, D. W. Wichern, *Applied multivariate statistical analysis*, Englewood Cliffs, NJ, USA: Prentice, 1992, p. 4.
[16] M. C. Thrun and A. Ultsch, "Uncovering High-dimensional Structures of Projections from Dimensionality Reduction Methods," *MethodsX*, vol. 7, p. 101093, 2020. doi: 10.1016/j.mex.2020.101093.
[17] M. W.Berry, S. A. Pulatova, and G. W. Stewart, "Computing sparse reduced-rank approximations to sparse matrices," *ACM Trans. Math. Softw.*, vol. 31, pp. 252–269, 2005.
[18] G. W. Stewart, "Error analysis of the quasi-Gram–Schmidt algorithm," *SIAM J. Matrix Anal. Appl*, vol. 27, no. 2, pp. 493–506, 2004.
[19] M. Popolizio, A. Amato, V. Piuri, and V. Di Lecce, "Improving classification performance using the semi-pivoted QR approximation algorithm," in *Rising Threats in Expert Applications and Solutions. Lecture Notes in Networks and Systems*, vol. 434, V. S. Rathore, S. C. Sharma, J. M. R. Tavares, C. Moreira, B. Surendiran, Eds., Singapore: Springer, 2022. doi: 10.1007/978-981-19-1122-4_29.
[20] J. Minguillón, J. Meneses, E. Aibar, N. Ferran-Ferrer, and S. Fãbregues, "Exploring the gender gap in the Spanish Wikipedia: Differences in engagement and editing practices," *PLoS One*, vol. 16, no. 2, p. e0246702, 2021.
[21] P. Borah, D. K. Bhattacharyya, and J. K. Kalita, "Malware dataset generation and evaluation," in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, IEEE, 2020.
[22] A. P. Singh, V. Jain, S. Chaudhari, F. A. Kraemer, S. Werner, and V. Garg, "Machine learning-based occupancy estimation using multivariate sensor nodes," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018.
[23] S. E. Golovenkin, J. Bac, A. Chervov, E. M. Mirkes, Y. V. Orlova, E. Barillot, et al., "Trajectories, bifurcations, and pseudo-time in large clinical datasets: applications to myocardial infarction and diabetes data," *GigaScience*, vol. 9, no. 11, p. giaa128, 2020, doi: 10.1093/gigascience/giaa128
[24] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P.Patel, A. Tiwari, et al., "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–81, 2017, doi: 10.1016/j.neucom.2017.06.053.
[25] W. Pedrycz, "Algorithms of fuzzy clustering with partial supervision," *Pattern Recog. Lett.*, vol. 3, pp. 13–20, 1985.
[26] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, pp. 32–57, 1973.
[27] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm, *Comput Geosci*, vol. 10, no. 2–3, pp. 191–203, 1984.
[28] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content based image retrieval at the end of the early years," *IEEE Trans. PAMI*, vol. 22, pp. 121349–1380, Dec 2000.
[29] W. M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *J. Am. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971, doi: 10.2307/2284239.
[30] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.
[31] B. Venkatesh and J. Anuradha, "Fuzzy Rank Based Parallel Online Feature Selection Method using Multiple Sliding Windows," *Open*

Comput. Sci., vol. 11, no. 1, pp. 275–287, 2021, doi: 10.1515/comp-2020-0169.

[32] S. Visalakshi and V. Radha, "A literature review of feature selection techniques and applications: Review of feature selection in data mining," in *2014 IEEE International Conference on Computational Intelligence and Computing Research*, 2014, pp. 1–6. doi: 10.1109/ICCIC.2014.7238499.

[33] P. Kromer, J. Plato and V. Snael, "Genetic algorithm for the column subset selection problem," in *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), Birmingham, UK*, 2014, pp. 16–22. doi: 10.1109/CISIS.2014.3

[34] I. T. Jolliffe, *Principal component analysis*, New York: Springer Verlag, 1986.