Quantum clustering in non-spherical data distributions: finding a suitable number of clusters

Raúl Casaña-Eslava[a], Ian H. Jarman[a], Paulo J. G. Lisboa[a], José D. Martín-Guerrero[b]

[a] *Liverpool John Moores University (LJMU), Liverpool, United Kingdom,*
E-Mail: r.v.casanaeslava@ljmu.ac.uk
[b] *Universitat de València (UV), València, Spain,*
*E-Mail:*jose.d.martin@uv.es

Abstract

Quantum Clustering (QC) provides an alternative approach to clustering algorithms, several of which are based on geometric relationships between data points. Instead, QC makes use of quantum mechanics concepts to find structures (clusters) in data sets by finding the minima of a quantum potential. The starting point of QC is a Parzen estimator with a fixed length scale, which significantly affects the final cluster allocation. This dependence on an adjustable parameter is common to other methods. We propose a framework to find suitable values of the length parameter σ by optimising twin measures of cluster separation and consistency for a given cluster number. This is an extension of the Separation and Concordance framework previously introduced for K-means clustering. Experimental results on two synthetic data sets and three challenging real-world data sets show that optimisation of cluster separation identifies QC solutions with consistently high Jaccard score measured against true-cluster labels while optimisation of cluster consistency provides insights into hierarchical cluster structure.

Keywords: Quantum clustering, Non-spherical data distributions, Number of clusters, Parameter optimization, Separation and Concordance

1. Introduction

As interest in knowledge extraction from data grows, this typically includes exploratory analysis especially when the data are unlabelled. A central step in exploratory data analysis is the discovery of different categories or profiles in the data. Clustering algorithms are efficient methods for unsupervised learning among which a frequently used algorithm is K-Means [1]. This method implements a hard partition of the data by identifying representative points, the prototypes, which minimize the sum of within cluster squared Euclidean distances as shown in Eqs. (1) and (2):

$$J(\Theta, U) = \sum_{i=1}^{N} \sum_{j=1}^{K} u_{ij} \|x_i - \Theta_j\|^2 \tag{1}$$

$$u_{ij} = \left\{ \begin{array}{ll} 1, & d(x_i, \Theta_j) = \min_{k=1,\ldots,K} d(x_i, \Theta_k) \\ 0, & otherwise \end{array} \right\} \quad i = 1, \ldots, N \tag{2}$$

where $d(x_i, \Theta_j)$ is the distance between the i-th pattern and the j-th prototype, N the number of patterns and K the number of clusters. In spite of its simplicity, K-Means is an adequate and efficient choice of clustering algorithm in many cases. However, it suffers from a number of drawbacks that limits its applicability. In particular, it tends to find spherical clusters formed by approximately the same number of patterns. Moreover, the final cluster allocation vary significantly with the choice of prototype initialisation. In addition, there is a requirement to pre-set the number of clusters, K, even though the optimal value of K is generally not known in advance. Consequently, K-means may mix natural clusters or break them up with unnecessary intermediate clusters [2, 3, 4]. A previous publication [3] proposed a framework to ensure that optimal results can be reproduced when K-means is repeatedly applied to the same data. This framework relies on a parametrisation of the set of clustering solutions obtained for different prototype initialisations and cluster numbers, using measurements of cluster separation and of the internal consistency, or concordance, between multiple clustering solutions obtained for the same K, hence the term SeCo for Separation and Concordance mapping of the space of clustering solutions.

This paper proposes an extension of this method to find suitable length parameters when applying Quantum Clustering (QC) [5, 6, 7]. This alternative clustering methodology is attractive because it more naturally fits non-spherical data distributions [8, 9] and it is also better suited to model clusters of different sizes present in the same data set. We start with a review of existing methods for optimisation of the value of the scale parameter directly from the dispersion properties of the data, initially proposed in [10, 11], before comparing the results with the proposed alternative method for estimating the length parameter σ using the outcome of QC clustering rather than the data alone.

An estimation of the scale parameter was proposed in [12], where σ and the Parzen estimator are computed locally based on the information of their K-Nearest Neighbours. To tackle the problem of high-dimensional data and their scalability, QC can be combined with techniques of dimensionality reduction [12, 13, 14, 15].

The rest of the paper is outlined as follows. Section 2 introduces the QC algorithm. Section 4, reviews methods for estimating optimal values of σ from the data, by application to a set of benchmarking data sets which include two synthetic examples and three real-world data sets. This is followed in Section 5, by the introduction of the SeCo framework and description of its application to the same data sets to set the length scale from clustering results. The experimental results are discussed in Section 6 from which conclusions are then drawn in Section 7.

2. Methodology

2.1. Quantum clustering

Many clustering algorithms are based on locations of points in the data space. That methodology works sufficiently well in many situations but it describes a problem that might be ill defined. QC proposes a different methodology, inspired in concepts from Quantum Mechanics [5, 6, 7]. It starts with a Parzen-window estimator of the probability distribution based on the data; then, a Gaussian kernel generates a probability distribution from the data points in a Euclidean space, as shown in Eq. (3):

$$\Psi(\mathbf{x}) = \sum_i \exp\left(-\frac{(\mathbf{x}-\mathbf{x_i})^2}{2\cdot\sigma^2}\right) \tag{3}$$

where $\mathbf{x_i}$ are the data points. QC associates maxima of this function with cluster centers in a Hilbert space driven by the Schrödinger equation so that minima of the Schrödinger potential are associated with cluster prototypes. The Schrödinger equation is given by Eq. (4):

$$H\Psi \equiv \left(-\frac{\sigma^2}{2}\nabla^2 + V(\mathbf{x})\right)\Psi(\mathbf{x}) = E\Psi(\mathbf{x}) \tag{4}$$

where $\Psi(\mathbf{x})$ is a solution of the equation (eigenstate), H is the Hamiltonian, V the potential energy and E is an energy eigenvalue. The simplest case is given by a single Gaussian where Ψ represents a single point at x_1. It leads to the potential $V = \frac{1}{2\sigma^2}\left(\mathbf{x}-\mathbf{x_1}\right)^2$; this is a well-known potential in Quantum Mechanics, the so-called harmonic potential whose ground state corresponds to the eigenvalue $E = \frac{\hbar\omega}{2} = \frac{d}{2}$, where $\hbar$ is the reduced Planck constant, ω the angular frequency, and d the space dimension. Therefore, the Gaussian function describes the ground state of H.

Although in Quantum Mechanics the usual strategy is to find solutions for $\Psi(\mathbf{x})$ given the potential, the proposal of QC is the other way around, i.e., since $\Psi(\mathbf{x})$ is already determined by the data points, the goal is to find a potential $V(\mathbf{x})$ whose solution is the given $\Psi(\mathbf{x})$:

$$V(\mathbf{x}) = E + \frac{\frac{\sigma^2}{2}\nabla^2\Psi}{\Psi} = E - \frac{d}{2} + \frac{1}{2\sigma^2\Psi}\sum_i \left(\mathbf{x}-\mathbf{x_i}\right)^2 \exp\left(-\frac{(\mathbf{x}-\mathbf{x_i})^2}{2\cdot\sigma^2}\right) \tag{5}$$

If V is positive definite, $\min V = 0$, and hence $E = -\min \frac{\frac{\sigma}{2}\nabla^2\Psi}{\Psi}$, which implies that $0 < E < \frac{d}{2}$.

After cluster prototypes are found, the final task is to assign each pattern to a given cluster. This can be done by means of a gradient descent algorithm;

defining $\mathbf{y_i}(0) = \mathbf{x_i}$, the trajectories of this point over time, $\mathbf{y_i}(t)$, is iterated as follows, where η is the learning rate that controls the speed of approaching the nearest minimum:

$$\mathbf{y_i}(t + \Delta t) = \mathbf{y_i}(t) - \eta(t)\nabla V(\mathbf{y_i}(t)) \tag{6}$$

letting $\mathbf{y_i}$ reach an asymptotic fixed value coinciding with a cluster prototype [7].

2.2. Parameter optimisation

The QC code used in this work is based on the Matlab COMPACT GUI [16]. Among the different parameters that appear in this implementation, the most important one in the QC algorithm is σ; the rest of the parameters have been set to default because that was the setup that provided the best results in [8], the original work from which this paper is an extended version:

- Learning rate, $\eta = 0.10$
- Number of steps = 100
- Rescale each step = FALSE
- Use of QC Core = FALSE

2.3. Number of clusters

As cluster prototypes are associated with potential minima in QC, and the only undetermined parameter is σ, different clustering solutions will be obtained for different values of σ. In particular, as σ is decreased, more and deeper minima are expected to be found. The tuning of σ is usually carried out by means of varying it smoothly and looking for stability of cluster solutions [7]. Our conjecture is that if one could optimise the value of σ, QC would become an automatic clustering algorithm, able to find the best combination of structures (in principle, of different shapes) that define the data.

In the next section we introduce several data sets, real and synthetic, which will be used to illustrate the application of the proposed methodology.

3. Description of the data

Five different data sets are used in this study to illustrate the application of the proposed methods: two synthetic data sets and three real data sets commonly used to benchmark clustering methods. The data sets are described in detail in Sections 3.1 and 3.2. Both synthetic data sets are generated using Gaussian distributions, some of them highly overlapped, thus producing unique structures formed by different Gaussian distributions that are difficult to separate by clustering. The real-world datasets demonstrate markedly different cluster shapes and mix clusters of different sizes in the same data.

3.1. Synthetic data sets

Artificial Data Set #1 (4 clusters). This data set depicts a first possible scenario, relatively simple, formed by 800 samples in a three-dimensional space and four clusters with the same number of observations each. The aim is to evaluate how QC reacts when there are three groups of clusters equidistant and how it affects the internal Concordance when QC tries to allocate the labels with the wrong cluster number. The four clusters are generated by a spherical Normal distribution. Figure 1 shows that two clusters are partially overlapped and the other two are totally separated.

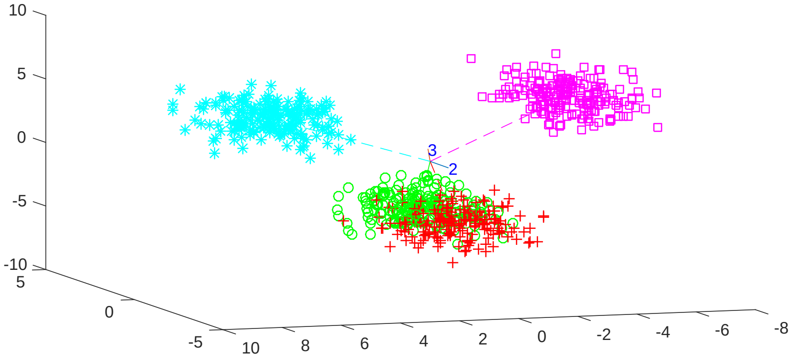

Figure 1: Principal components' visualisation of artificial data set #1. This data set contains four clusters, generated by a Normal distribution.

Artificial Data Set #2 (10 clusters). This data set has been used in [2, 3, 4], it is based on 1076 observations in three dimensions with 10 clusters. Each cluster has a different proportion of observations, being some of them sparse. The overlapping between two clusters is important thus being quite difficult to detect, however there are other clusters easily separable. The covariance matrix of the Gaussian distributions is not spherical. Figure 2 shows the principal components of this dataset.

3.2. Real data sets

Wine data set. This dataset available on the UCI data repository [17] is well known and comprises 178 observations in 13 variables. It was acquired from a chemical analysis of wines grown in one region of Italy. Each of the attributes consists of measurements taken from the various wines, which are created using three distinct cultivars. The attributes are Alcohol, Malic Acid, Ash, Alcalinity of the Ash, Magnesium, Total Phenols, Flavanoids, Nonflavanoid Phenols, Proanthocyanins, Color Intensity, Hue, OD280/OD315 of diluted wines and Proline. The cultivars are well separated with the expectation of good classification by approaches like K-means.

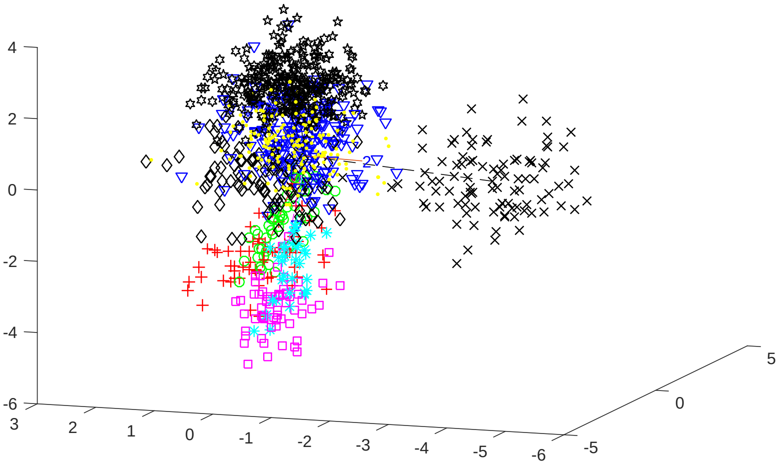

Figure 2: Principal components' visualisation of artificial data set #2 (10 clusters). This data set contains 10 clusters, generated by a Normal distribution.

Olive oil data set. The *Italian olive oil* data set consists of 572 samples and 10 variables. Eight variables describe the percentage composition of fatty acids found in the lipid fraction of these oils, which is used to determine their authenticity. The remaining two variables contain information about the classes, which are of two kinds: three "super-classes" at country level: North, South, and the island of Sardinia; *and* nine collection area classes: three from the Northern region (Umbria, East and West Liguria), four from the South (North and South Apulia, Calabria, and Sicily), and two from the island of Sardinia (inland and coastal Sardinia).

The goal is to distinguish the oils from different regions and areas in Italy based on their combinations of the fatty acids. The clusters corresponding to classes all have different shapes in the eight-dimensional data space defined by the concentration of fatty acids [18, 19].

Iris data set. The Iris dataset [17] was introduced by Sir Ronald Fisher in 1936 for the purpose of using it as an example in explaining discriminant analysis. The dataset comprises 150 data points in four dimensions matching the Sepal and Petal width and height for each observation. There are three cohorts present in the data: Setosa, Virginica and Versicolor.

3.3. Data pre-processing

The QC algorithm is designed to work in a normalised data space so that σ values are bounded in the range $[0, 2]$ [5, 6, 7, 16]. For that reason, it is necessary to implement a previous data pre-processing.

The first step is to apply the reduced Single Value Decomposition, using the $U_{m\text{x}n}$ matrix of left-singular vectors as the new data.

The data need to be normalized to a unit hyper-sphere. However, in order to preserve length information, an extended vector is used with a column of ones added to the original matrix, $U_{m\text{x}n}$. In addition, the original data matrix is re-scaled by a single factor λ to ensure that mean length of the rows is 1. In summary:

$$\begin{aligned} Data_{m\text{x}n} &= U_{m\text{x}n}\Sigma_{nxn}V^*_{n\text{x}n} \\ U'_{m\text{x}n} &= U_{m\text{x}n}/\lambda \\ Z &= rnorm([U', 1]_{m\text{x}(n+1)}) \end{aligned} \tag{7}$$

where *rnorm* is a function that normalizes every matrix row by length 1.

In this way, raw data is transformed in a normalised hyper-sphere space, but keeping sample module information, and where the variance s_i is bounded to $[0, 2]$.

In some datasets QC performance can be improved reducing the data dimensionality, but in this work the option of reducing the dimensionality through PCA has been skipped so that all datasets have the same preprocessing.

4. Setting the length scale from the data

For the optimisation of σ, we make use of a statistical approach for estimating the scale parameter of a potential function presented in [10, 11], that can be translated for the estimation of σ in QC. The estimation is based on calculating the average Euclidean distance to a set of neighbours for each data sample; the resulting local variances are modelled as a Gamma distribution and the scale parameter is estimated as the mean of this Gamma distribution. Given a data sample $\mathbf{x_i}$, a ranking of all other data samples according to their squared Euclidean distance to $\mathbf{x_i}$ is performed:

$$R_K(\mathbf{x_i}) = \left\{ x_{(k)} \mid \ \|x_{(k-1)} - x_i\|^2 < \|x_{(k)} - x_i\|^2 \right\} \tag{8}$$

for $k = 1, 2, \ldots, K$, where $x_{(k)}$ represents the K-nearest neighbours of $\mathbf{x_i}$, and $\| \ \|$ denotes the Euclidean distance between a data sample and x_i. Since the variance s_i of the local neighbourhood around each sample can be calculated, an empirical distribution of local variance estimates can be formed by considering several data samples $\mathbf{x_i}$ and their neighbourhoods $R_K(\mathbf{x_i})$. The probability density function that characterises the empirical local variance is modelled by the Gamma distribution:

$$p(s) = \frac{s^{\alpha-1}}{\beta^\alpha \Gamma(\alpha)} \exp(-\frac{s}{\beta}) \tag{9}$$

where $\alpha > 0$ is the shape parameter, and $\beta > 0$ is the scale parameter of the Gamma distribution $\Gamma(\cdot)$:

$$\Gamma(t) = \int_0^\infty r^{t-1} \exp(-r) dr \tag{10}$$

The parameters α and β are estimated from the empirical distribution of the variance, modelled by Eq. (9). There are different methods to calculate the parameters α and β; the moments method is proposed in [10, 11]:

$$\hat{\alpha} = \left(\frac{\overline{s}}{l}\right)^2 \quad ; \quad \hat{\beta} = \frac{l^2}{\overline{s}} \tag{11}$$

where $\overline{s}$ and l are the sample mean and standard deviation of the distribution of nearest neighbour distances for a given value of K. The estimation of $\hat{\sigma}$ can then be obtained as $\hat{\sigma} = \hat{\alpha}\hat{\beta}$.

As detailed in next sections of the paper, this methodology is tested in QC to find out whether it can be successfully applied to detect a suitable number of clusters in several data sets (both synthetic and real) with different characteristics.

Two methods were used to find the most suitable fit from the data, both following the procedure described in [10, 11]. The first method estimates σ using the average dispersion of the data and the second fits the distribution of the dispersion using gamma functions. We show that both methods lead to similar values of the scale parameter σ for each data set, but these values are not necessarily optimal.

The data dispersion at each data point is estimated using K-nearest neighbours (K-NN) with increasingly large numbers of near neighbours. Given a certain K, the α and β parameters of the Gamma distribution can be obtained either using the moments method as described in Eq. (11) or fitting the s_i empirical distribution to the Gamma distribution; in this work the *gamfit* Matlab built-in function was used. Figure 3 shows the estimation carried out by these two different ways of calculation; as expected, both produce the same result: $\hat{\sigma} = \hat{\alpha}\hat{\beta}$. The vertical lines show the best σ solution according to the Jaccard score for the two external labels of the olive oil data set; that helps to visualise the value of the optimal σ. The bottom-left graph of Figure 3 shows the function $\sigma = f(KNN)$. Also one may observe the linear regression fitted to the interquartile range of σ values. Over $K = N/2$ a Normal distribution behaviour is expected, where the variance increases linearly as K increases. The bottom-right graph shows the s_i standard deviation with the aim of providing additional information to estimate the best K.

The next step is to decide which K is the appropriate to select σ. Two options have been discussed:

- According to [10, 11], $K = N/4$, being N the sample size, is a reasonable choice. This approach suggests that the first quartile is the Separation border between the variance of close neighbours and the variance produced

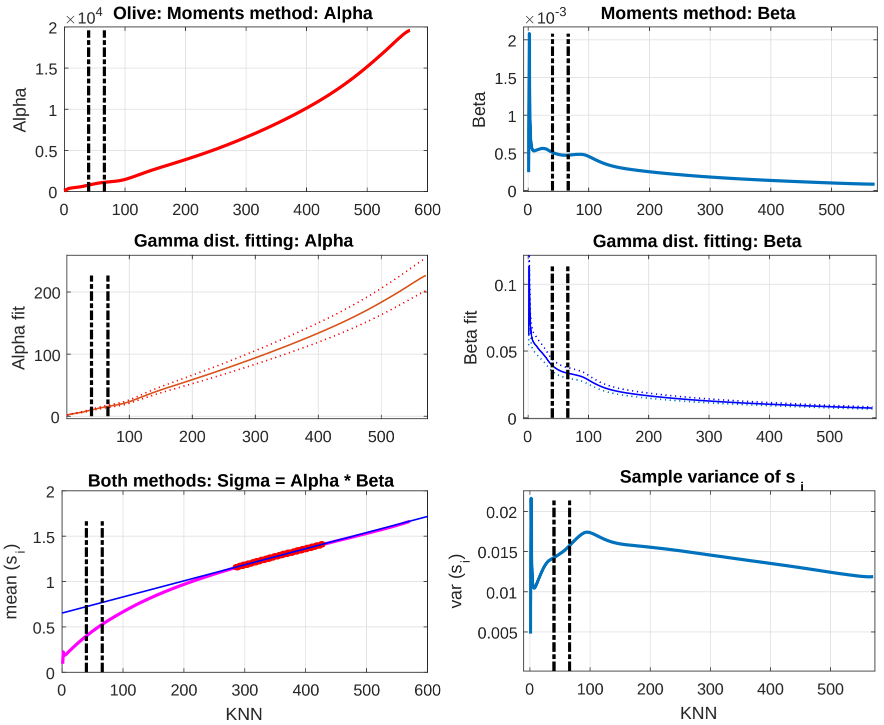

Figure 3: Olive oil data set: Two methods to estimate $\hat{\sigma}$. Moments method in top graphs, fitting Gamma distribution in middle graphs. Estimated $\hat{\sigma}$ in bottom left and s_i sample standard deviation in bottom right. Vertical lines indicate the best σ solution according to the Jaccard score for the two external labels of the olive oil data set.

by remote-enough samples for Normal behaviour.

- The other option goes beyond [10, 11] and it is based on the assuming that the variance has a normal behaviour when K is sufficiently large to include remote neighbours, like K in the third quartile of the sample size. Fitting a linear regression in this range of K enables to compare the variance with near neighbours against the far ones. One criterion could be to choose a K that separates more than 50% of the total distance between the variance s_i and the linear regression.

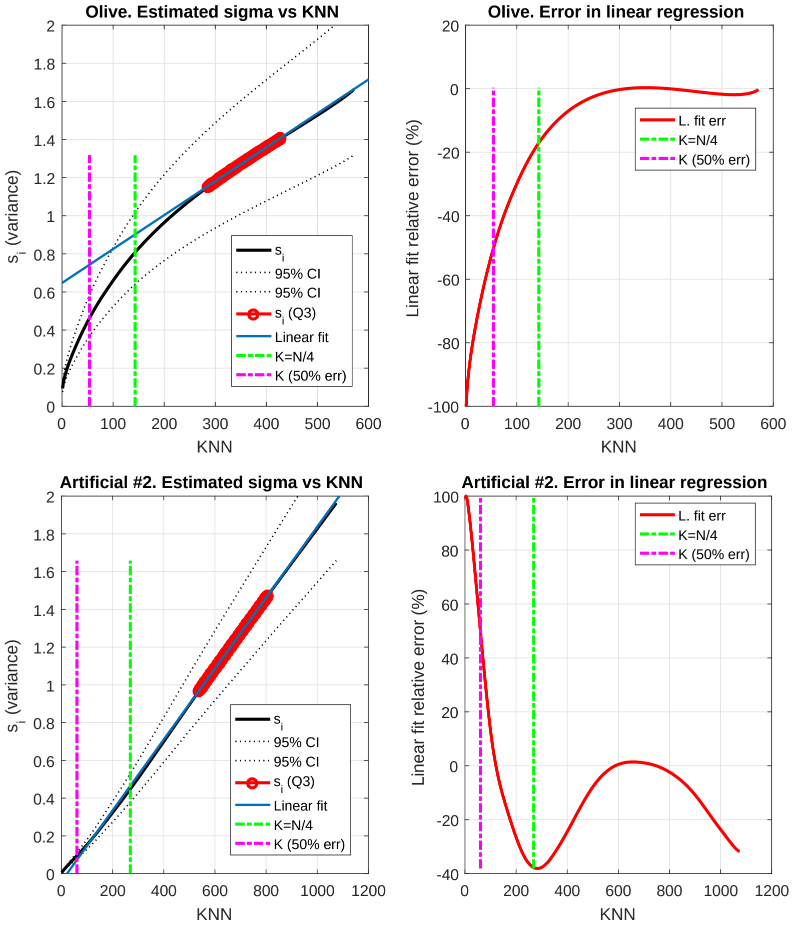

Figure 4: Olive oil and artificial data set #2 (10 clusters): Left graphs show the estimated s_i variance curve with confidence intervals at 95%, the linear regression on the interquartile range of KNN, and the two suggested KNN solutions, $K = N/4$ and K at 50% with the linear regression. Right graphs show the error between s_i and the linear fit, and the two suggested KNN solutions.

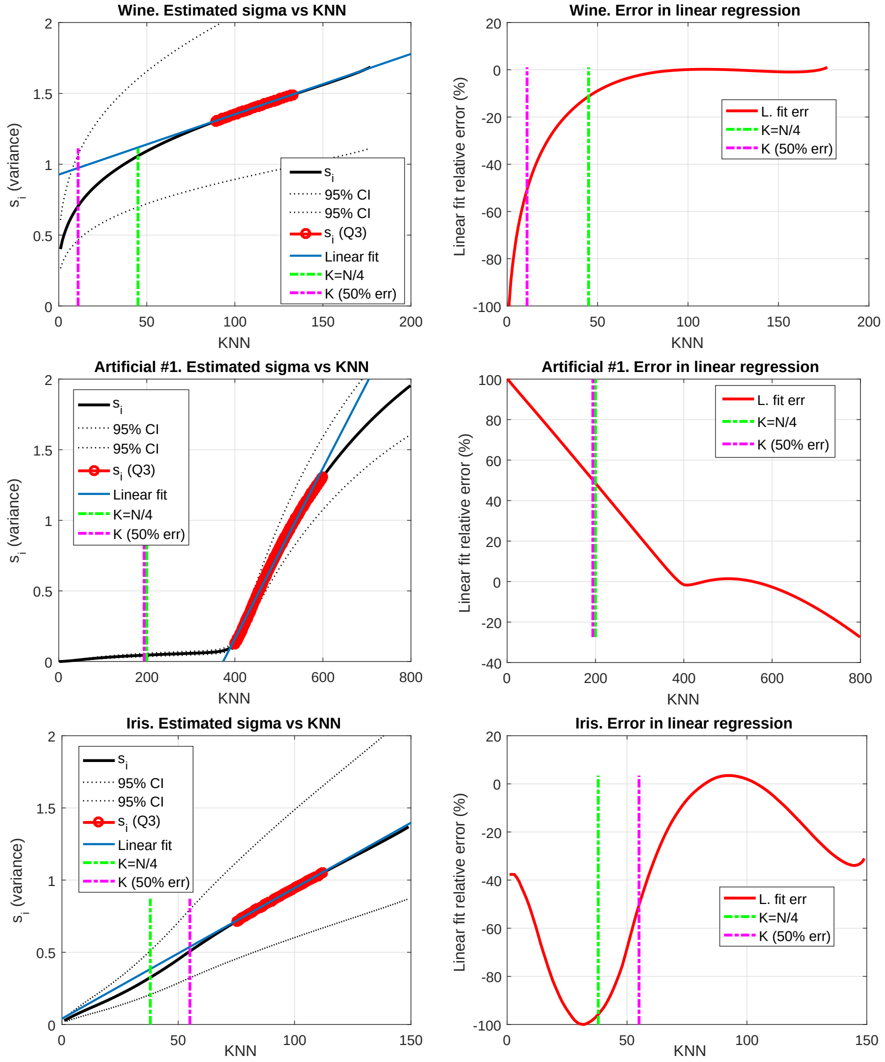

Figure 5: Wine, artificial data set #1 (4 clusters) and Iris data set: Left graphs show the estimated s_i variance curve with confidence intervals at 95%, the linear regression on the interquartile range of KNN, and the two suggested KNN solutions, $K = N/4$ and K at 50% with the linear regression. Right graphs show the error between s_i and the linear fit, and the two suggested KNN solutions.

Figures 4 and 5 present the two methods for estimating σ as a function of KNN in different datasets. One may observe a completely different s_i behaviour depending on the dataset, this affects the confidence intervals, the K selected and hence the σ estimated. This issue will be discussed thoroughly later.

It would be expectable that the s_i curves could reveal some information about the data internal structure, and that it would relate the KNN with the proper σ. But it is not the case, choosing a σ with this method seems somewhat arbitrary. There is a case to be especially noticed in Figure 5 for the artificial data set #1, which is formed by 4 clusters, 2 of them totally separated, having 200 samples per cluster approximately; the variance curve presents an abrupt behaviour when KNN has to include observations from the more distant clusters. This should provide a clear KNN to choose σ, but the best actual σ range is about $[0.65, 0.70]$, quite far away from the suggested $[0.2, 0.3]$. An additional problem is that this method offers a single solution that it varies strongly depending on a single premise, and hence, it is hard to create a general criterion that fits all the datasets. As QC needs more σ precision than that yielded by this method, we came up with an alternative approach, presented in Section 5.

These results illustrate the difficulty in establishing a criterion for estimation of consistently good values of the length parameter σ. This is addressed further in the next section.

5. Setting the length scale from clustering results

One of the main objectives in optimising QC to a given data set is to assess the QC solutions in an unsupervised way. This amounts to finding values of the length scale for the initial Parzen estimator, which is controlled by the Gaussian with parameter σ. In this section we will propose a framework using complementary measures of cluster performance to a) map the QC solution space, b) find suitable values for the number of clusters, K, and length parameter σ and c) generate insight into possible hierarchical structure in the data.

In the absence of external labels, we propose the use of a two-dimensional performance assessment framework, which we call Separation and Concordance (SeCo). This was first introduced in [2, 3, 4] to assess K-Means and Adaptive Resonance Theory (ART) models. The SeCo performance assessment is based on ΔSSQ and Concordance measurements grouped per cluster number solutions; by plotting Concordance versus ΔSSQ one may visualise how concordant is a group of K solutions at the same time assessing the cluster Separation. For K-Means, the Concordance measure is very important because of K-Means strongly dependence on the centroids initialisation; however, the Concordance is much less relevant for QC.

The unsupervised performance assessment can be done following the next steps:

a) The first step is to run the QC over σ values between $[0, 2]$ in regular intervals.

b) The second step is to measure the number of clusters per σ value; this information shows how the potential $V(\mathbf{x})$ evolves according to σ values, and reveals where the data structure is more stable: the wider σ range the more stable data structure. Figure 6 shows the solutions for the olive oil data set.

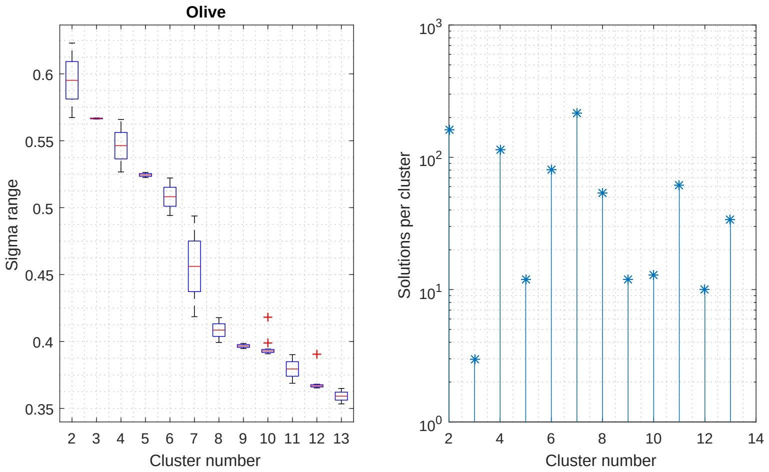

Figure 6: Olive oil data set: Left graph shows the σ range per number of clusters. Right graph shows the number of QC solutions per number of clusters. Solutions with clusters from 2 to 13 were filtered from the initial 1000 different σ values.

c) The third step is to obtain the SeCo framework. For every QC solution grouped by number of clusters, the ΔSSQ and the internal Concordance are calculated. The SeCo framework can be observed in figure 7 for the olive oil data set. Unfortunately, the graph needs to zoom in to appreciate each K in detail, and this justifies the plot of the next step.

d) In order to adapt the SeCo framework to QC, σ has been added as an additional variable in the SeCo framework. Plotting ΔSSQ against σ, and Concordance against σ, it is possible to observe all the relevant information in a straightforward way. Figure 8 shows that representation for the olive oil data set.

Section 6 will show a deeper analysis of the procedure to select the most useful K and the corresponding solution. The process of finding a sufficiently good solution for unknown data consists in two parts; firstly, a selection of an appropriate K, and secondly, a solution within all K-groups' solutions. The criterion to select K has been based on choosing the lowest K (for simplicity) that improves considerably the Separation and has a good Concordance (not necessarily the best).

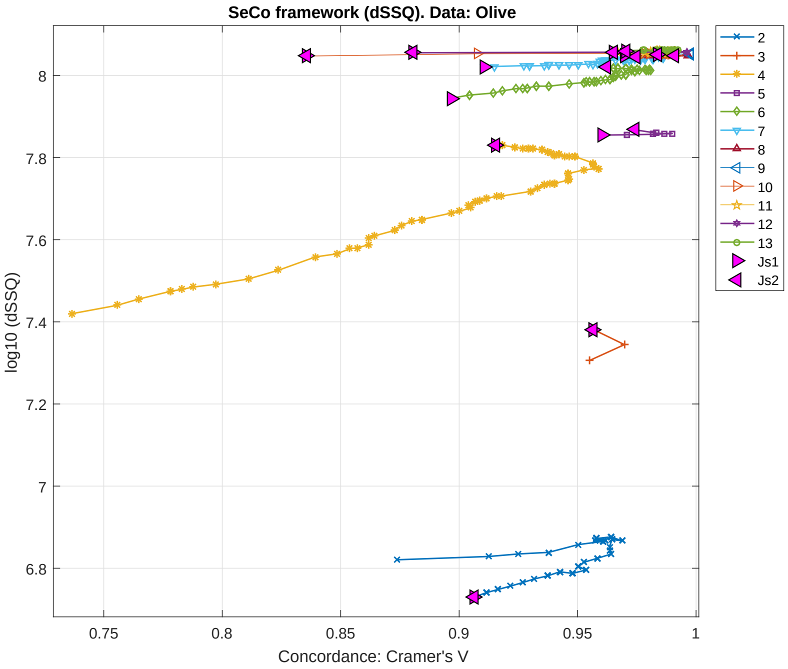

Figure 7: SeCo representation for the olive oil data set. The plots contains the Separation and the Concordance for each number of clusters, marking the solution with the best Jaccard score for each cluster number, using two sets of external labels: Js1 stands for the solution corresponding with three regions and Js2 for the case of nine areas. For example, for cluster numbers $K = 3$ and 4, a better ΔSSQ score is achieved with the smallest value of σ for that K, which is highest on the y-axis. Although it is not obvious from the value of ΔSSQ, the best Jaccard score for 3 labels, Js1, occurs for $K = 4$. And the best for 9 labels, Js2, for $K = 8$.

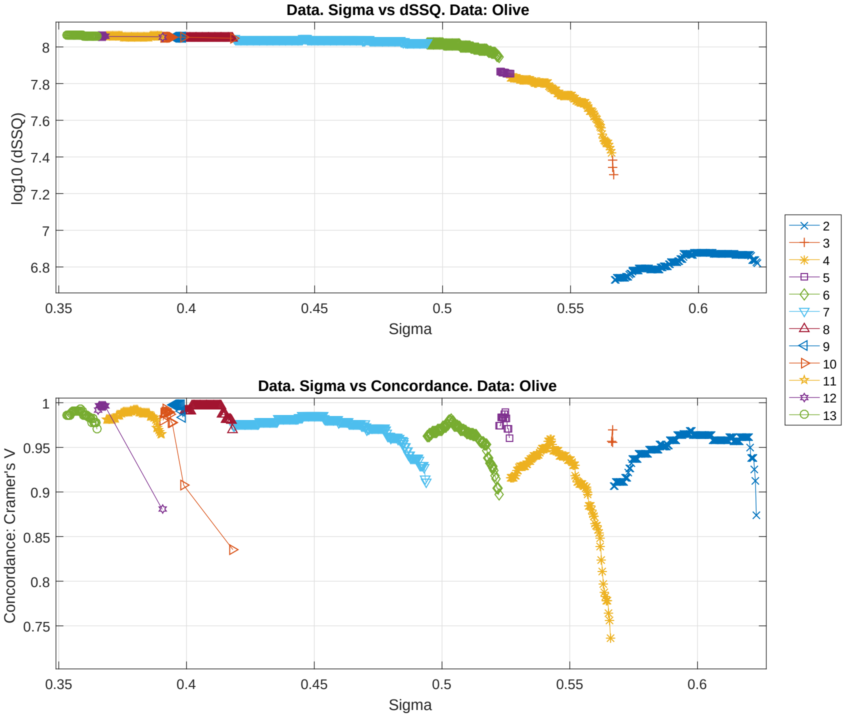

Figure 8: SeCo vs. σ for the olive oil data set. Top graph shows ΔSSQ vs. σ. Bottom graph internal Concordance vs. σ. Additionally it is possible to appreciate the σ range as an estimation of cluster stability in the quantum potential $V(\mathbf{x})$. This figure illustrates three main points: First, there is a value of σ where the Separation stabilizes. In this case it is $K = 6$. Second, within the range of K with high Separation, i.e. 6 and above, there is an increase in internal Concordance for $K = 8$. Thirdly, the value of K finally selected to be optimal for this data set, also has a wide range of values of σ. This confirms $K = 8$ in this case.

The external labels with the Jaccard scores can help to verify the conclusions obtained in this framework. For each K, three main solutions can be extracted, the solution with highest Separation, the solution with highest Concordance and the solution with highest Jaccard score (knowing the true labels). Comparing between them it is possible make an inference about which is the most relevant criterion. For instance, in Figure 9 they can be compared for the olive oil data set, where one may see that the solution with the best Separation (ΔSSQ) is frequently almost as good as the one with the best Jaccard score.

6. Discussion of the experimental results

This section is focused on the SeCo vs. σ plots for the different data sets described in Section 3. The rest of the graphs presented in the previous section have been omitted to limit the length of the paper, and also because SeCo vs. σ is the most relevant plot in order to decide a useful K. To support the conclusions, the Jaccard score plots of the true labels are presented, as well.

6.1. Synthetic data sets

6.1.1. Artificial data set #1 (4 clusters)

This dataset is designed to produce a Concordance conflict when $K \neq 4$ because there are three groups of equidistant clusters, one group contains two close clusters and the other two have a single cluster. The Concordance conflict is due to different label assignments when they are equally probable. At least, this is the expected behaviour for K-Means.

Figure 10 shows the SeCo vs σ. Here the expected conflict in Concordance is not as significant as it would be in K-Means. QC depicts the $K = 4$ as the widest σ range and it has constant high Concordance compared with other K values, what reveals the importance of the σ range as a cluster stability estimation.

In QC, the Concordance is not as relevant as it is in other algorithms because QC does not depend on random initialisations; every solution at σ_i is a slight variant of the solution at σ_{i-1} when σ values are sufficiently similar. The exception happens in the point when the cluster number changes, then again the solutions evolve gradually till the next K.

Figure 11 shows that any $K \geq 4$ is a suitable K; the solution with highest ΔSSQ has the same performance as the one with highest Concordance, given any K.

6.1.2. Artificial data set #2 (10 clusters)

In this data set, SeCo vs σ plot in Figure 12 shows a curve plateau in the representation of ΔSSQ, thus suggesting that the solutions for $K \geq 4$ are quite separated. Attending the group Concordance, the best are $K \in [5, 8]$, all have a reasonably wide σ range compared with $K > 8$. Thus, the chosen solution should be one of the $K \in [5, 8]$, depending on the desirable number of clusters.

Comparing these results with the supervised Jaccard score plot in Figure 13, it is observable the increasing performance with K even for values greater than

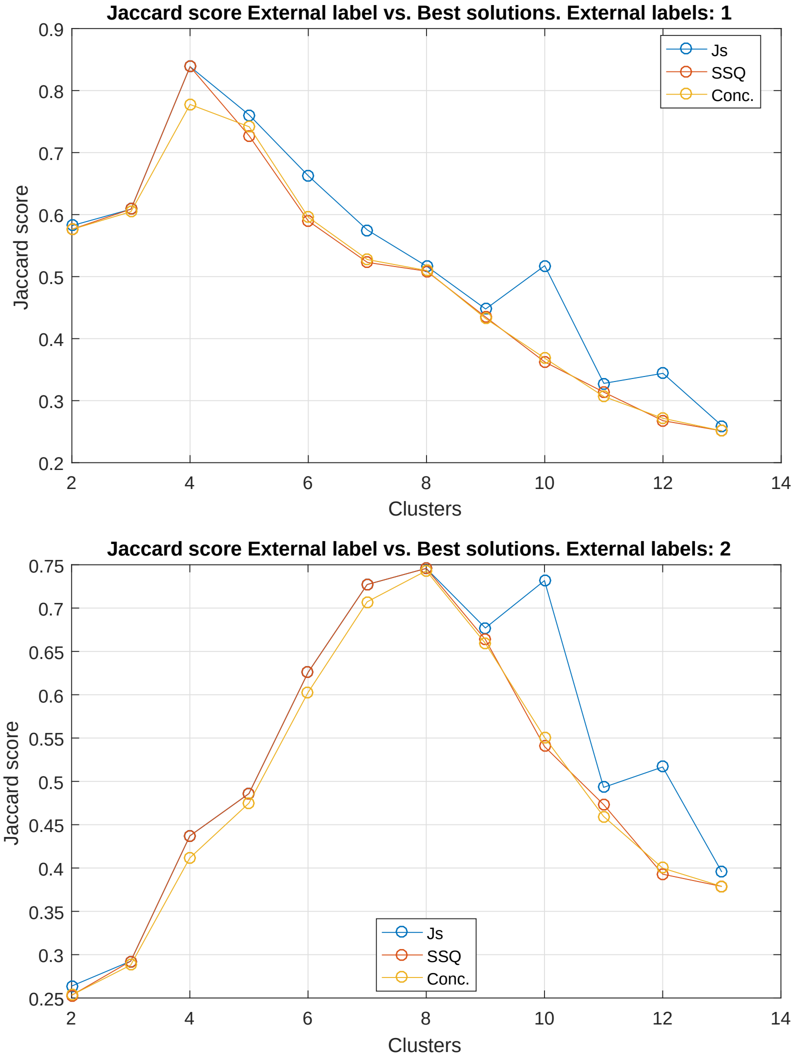

Figure 9: Jaccard score for three and nine regions of the olive oil data set. There are three solutions, the one with best Jaccard score for that K, the one with best Separation and the one with best internal Concordance. These solutions may not be the same, i.e. they may have different values of σ for the same K. Refer to Figure 7. This figure shows that, in general, a) the σ with best Separation (ΔSSQ) has a better Jaccard score than the σ with best Concordance, and b) the value of K selected by the proposed method, i.e. $K = 4$ and 8, have strong Jaccard scores.

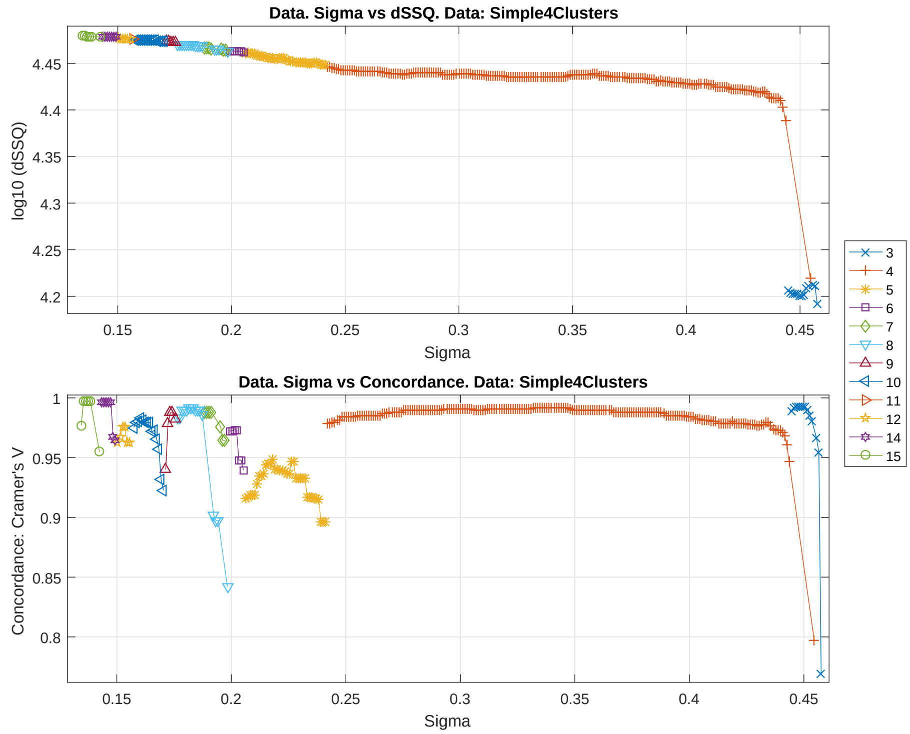

Figure 10: SeCo vs. σ for the artificial data set #1 (4 clusters). Top graph shows ΔSSQ vs σ. Bottom graph internal Concordance vs. σ. Additionally it is possible to appreciate the σ range as an estimation of cluster stability in the quantum potential $V(\mathbf{x})$.

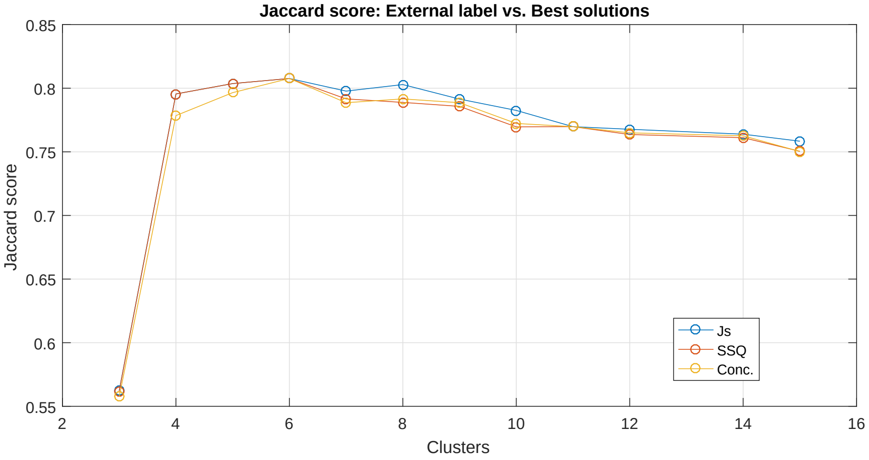

Figure 11: Jaccard score for four external labels on artificial data set #1 (4 clusters). There are three solutions, the one with best Jaccard score for that K, the one with best Separation and the one with best internal Concordance.

the actual number of clusters, although the Jaccard score performance for $K \in [5, 8]$ is close to the plateau curve.

In any case, it is remarkable the poor performance in general of the QC for this dataset, with scores lower than 0.35. Other interesting aspect is the performance differences for the best Concordance solution and the best Separation solution with $K > 8$, being better for the latter.

6.2. Real data sets

6.2.1. Olive oil

Figure 8 shows a considerable improvement in terms of ΔSSQ when the cluster number passes from $K = 2$ to 3 and so on, but ΔSSQ seems to stabilize in $K = 6$. This is a common pattern observed in all the tested datasets, there is a certain K where the ΔSSQ reaches the curve plateau. In this point QC already has found the main clusters, but beyond this point, more K only splits some clusters in additional subdivisions without really improving ΔSSQ. This is the main hint to indicate a suitable K beyond this point.

Next hints to pay attention are the internal Concordance and the σ range per K. From solutions with $K \geq 4$, the K with the highest internal Concordance as a group would be a good candidate. Next priority should be to give priority to those values of K with a wider σ range, for instance avoiding $K = 3$, 5, 9, 10 or 12. With those priorities, the best candidate should be $K = 8$. Regarding which is the best solution within $K = 8$, the priority should be the solution with highest ΔSSQ. This statement is based on the observation of the Jaccard score plots shown in Figure 9, where the solution with highest ΔSSQ is always closer to the best Jaccard score solution than the one with highest Concordance.

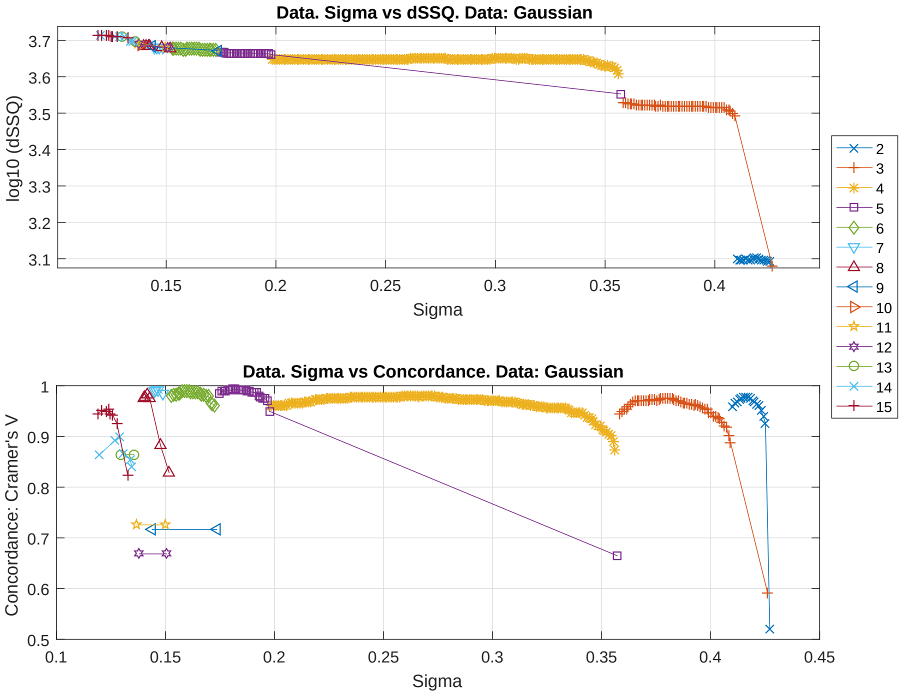

Figure 12: SeCo vs. σ for the artificial data set #2 (10 clusters). Top graph shows ΔSSQ vs. σ. Bottom graph internal Concordance vs. σ. Additionally it is possible to appreciate the σ range as an estimation of cluster stability in the quantum potential $V(\mathbf{x})$.

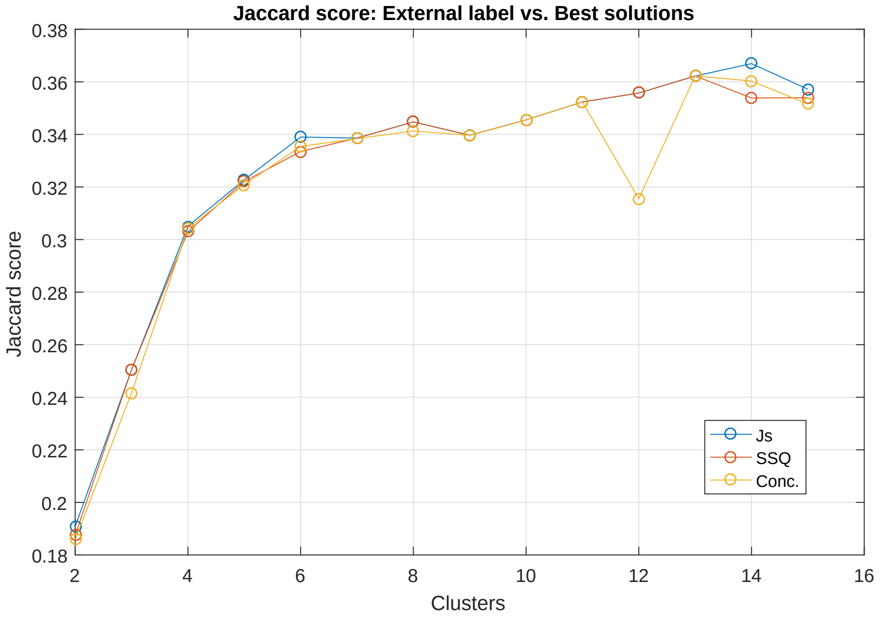

Figure 13: Jaccard score for 10 external labels in the artificial data set #2 (10 clusters). There are three solutions, the one with best Jaccard score for that K, the one with best Separation and the one with best internal Concordance.

6.2.2. Wine

The results of this dataset can be observed in figure 14; they are different from those obtained with the olive oil data set. The first difference is in the number of solutions, the main reason is due to the narrow σ range; for $K \in [2, 13]$, $\sigma \in [0.452, 0.493]$; this situation helps to explain how difficult is to find a suitable value of σ with the KNN approach.

Other aspects to remark are the unexpected valley in the ΔSSQ curve or the K fluctuation for adjacent σ values. These aspects point to a bad performance of QC in this dataset, and in fact, if one observes the Jaccard score (Figure 15), a poor performance is observed. The wine data set is supposed to have easily separable clusters, but QC does not work well, probably because the hypersphere space transformation overlaps two labels when in the raw data it does not occur. In addition, the ratio observations / features is quite low, 174/13.

In any case, a suitable K based on SeCo vs σ plot would be $K = 8$ because it is the first K with high ΔSSQ and with good Concordance, despite its low σ range.

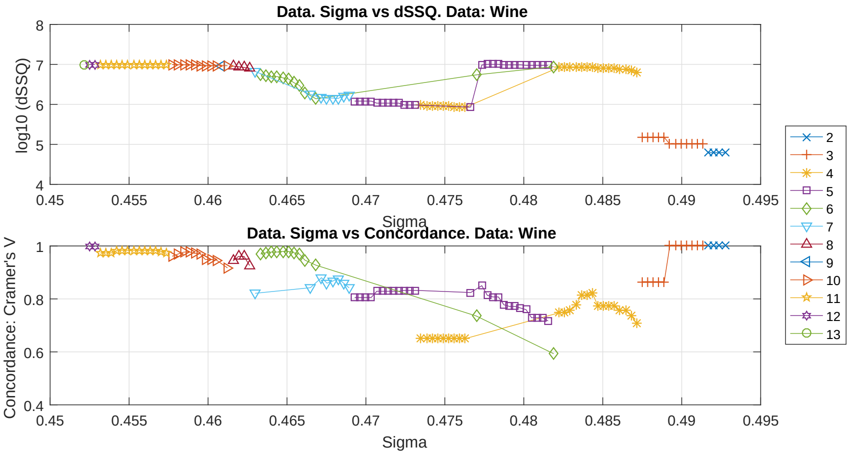

Figure 14: SeCo vs. σ for the wine data set. Top graph shows ΔSSQ vs. σ; bottom graph internal Concordance vs. σ. Additionally it is possible to appreciate the σ range as an estimation of cluster stability in the quantum potential $V(\mathbf{x})$.

6.2.3. Iris

For the results corresponding to Iris data set, shown in Figure 16, and following the ΔSSQ priority, the chosen K would be $K = 4$, which has a good Concordance and a wide σ range. However, Figure 17 shows that the solutions for $K = 2$ or 3 have a higher Concordance and a higher Jaccard score than the chosen ones with $K \geq 4$, although this information would be unknown in an unsupervised scenario.

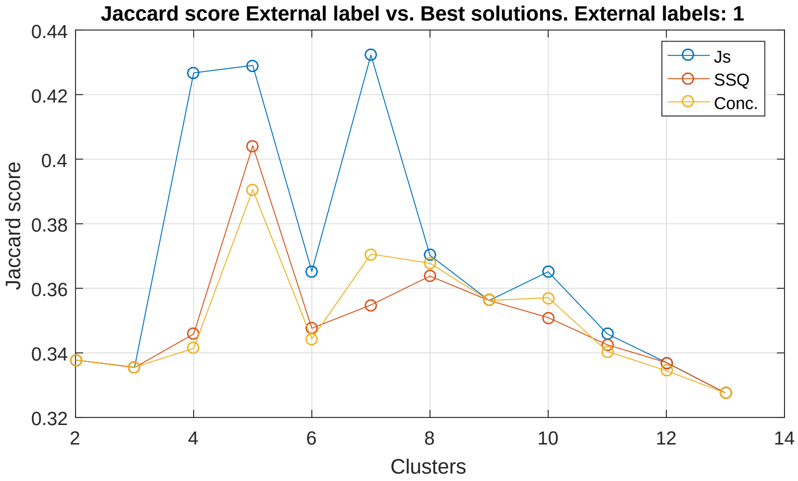

Figure 15: Jaccard score for three external labels in the wine data set. There are three solutions, the one with best Jaccard score for that K, the one with best Separation and the one with best internal Concordance.

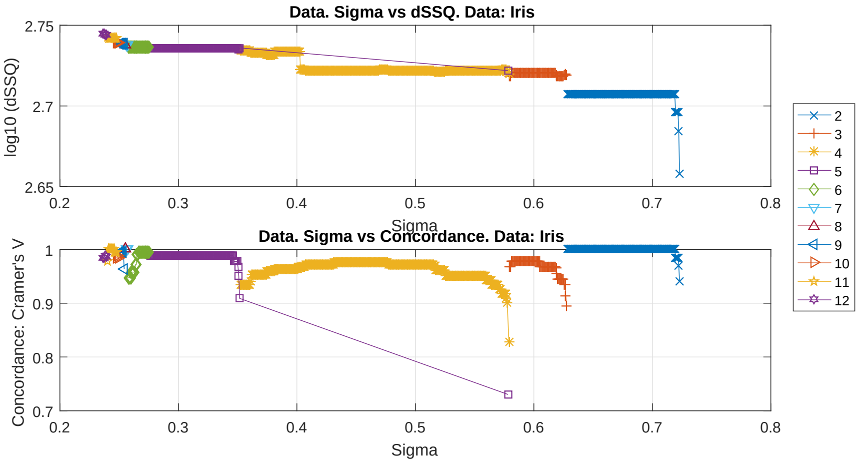

Figure 16: SeCo vs. σ for the Iris data set. Top graph shows $\Delta SSQvs.\sigma$. Bottom graph internal Concordance vs. σ. Additionally it is possible to appreciate the σ range as an estimation of cluster stability in the quantum potential $V(\mathbf{x})$.

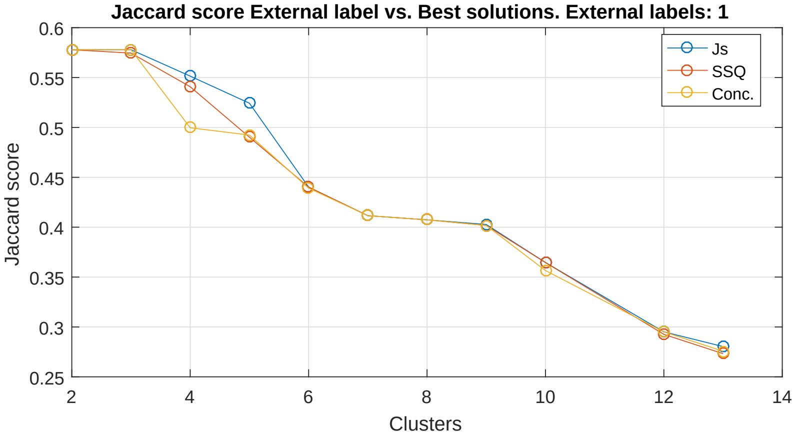

Figure 17: Jaccard score for three external labels in the Iris data set. There are three solutions, the one with best Jaccard score for that K, the one with best Separation and the one with best internal Concordance.

6.3. Summary of results

Table 1 summarizes the main results obtained in the tested datasets. The ΔSSQ row indicates the chosen K according to the separation measure. The Cv row depicts those K which have the highest Cramér's V, revealing a possible underlying hierarchical structure. The Js row shows which K has the QC solution with the highest Jaccard score compared with the true labels, and finally, the last row shows the number of clusters of the true labels.

6.4. Methodology to find the value of K

This section describes a schematic procedure for finding the most suitable K and its QC solution for unknown data.

The algorithm 1 describes the methodology to find the SeCo parameters. Once the parameters have been obtained, the value of K with consistently best separation should be selected; then, the value of σ for the best separation is the QC of choice for that value of K. High values of the concordance measure for different number of clusters indicate the presence of a hierarchical structure.

6.5. Scalability

Regarding the time complexity of QC, it should be considered that QC depends on the length of the observations that generate the potential, n_{gen}, the number of points where the potential is computed, n_{alloc}, the dimension of the data, dim, and the number of steps applied in the stochastic gradient descent (SGD), $steps_{SGD}$. Therefore, an estimation of its time complexity is $O(n_{gen} * n_{alloc} * D * steps_{SGD})$.

The SeCo framework increases the time complexity in a factor that depends on the number of different σ to be sampled, being a time complexity of $O(n_{gen} *$

Table 1: Cluster number K identified by different criteria with corresponding Jaccard scores against true labels shown within [ ].

Maximal criterion	Artificial dataset 1	Artificial dataset 2	Olive oil data	Wine data	Iris data
ΔSSQ	6 [0.81]	6 [0.34]	8 [0.75]	5 [0.4]	5 [0.49]
Cv	6 [0.81]	5 [0.32] 6 [0.34]	5 [0.70] 8 [0.75]	2, 3, 6 All [0.34]	2 [0.58] 6 [0.44]
Js	≥4 [0.79-0.81]	10 [0.35]	4 [0.85] vs 3 lab. 8 [0.75] vs 9 lab.	4, 5, 7 All [0.43]	2, 3 Both [0.58]
Number of true clusters	4	10	3, 9	3	4

Algorithm 1 Get N QC solutions and SeCo parameters

```
$\sigma_N \leftarrow$ N-vector evenly distributed $\in ]0, 2[$
for $i = 1 : N$ do
    $Solution_i \leftarrow QC(\sigma_i)$        ▷ Solution: QC outcome (vector of labels)
    $\Delta SSQ_i \leftarrow SSQ_{1cluster} - SSQ_{Solution_i}$
    $CN_i \leftarrow$ Cluster Number($Solution_i$)
end for
Filter Solutions with $CN_i \in [CN_{min}, CN_{max}]$        ▷ To avoid excessive CN
for $CN_j = CN_{min} : CN_{max}$ do
    $Sol_j = Solutions$ with $CN = CN_j$
    for $i = 1 : max(Sol_j)$ do
        $CV_i = median(Cramer'sV(Sol_{ji}, Sol_{CN_j}))$    ▷ Internal Concordance
    end for
end for
```

$n_{alloc} * D * steps_{SGD} * \#\sigma$); in particular, 1,000 sigma samples have been used in the experiments presented in this paper. Nevertheless, this work has not been focused on time complexity or scalability, which are undoubtedly relevant topics for future research.

7. Conclusions

This paper has proposed two figures of merit to characterise the quality of solutions obtained by QC, from which cluster numbers can be identified which maximise the fit against true cluster labels. Maximisation of cluster separation identifies clusters with consistently high Jaccard score against true labels, while high values of cluster consistency provide insights about hierarchical cluster structures. The proposed framework provides useful guidance to set an optimal value for the length scale parameter σ for each data set.

Two approaches have been proposed to find the best σ, and in turn, the correct number of clusters. The first one is based on the variance of K-Nearest Neighbours, and the second one on sampling QC solutions to apply the SeCo framework. The results yielded by the former approach do not suggest its use as a rule of thumb, due to three main reasons:

- It offers only one solution.
- The σ confidence intervals are too wide for the QC variability.
- The estimated σ strongly depends on the data structure, being very difficult to establish a general procedure.

The SeCo framework approach is based on measures of Separation (ΔSSQ), internal Concordance and σ range per cluster. The σ range parameter subtracts importance from the Concordance. Although the Concordance is very important in other algorithms like K-Means, it is less critical in QC due to its unique solution per σ value. The SeCo framework approach involves a higher computational cost than the KNN approach because of the need to run the algorithm multiple times. However, the results offer a consistent method to make a performance assessment in an unsupervised way. The SeCo plots have been adapted to the QC, adding an extra parameter: σ. The advantage of the SeCo vs σ plots is that they depict the data structure and the most suitable QC solutions in a straightforward way.

A procedure to select the appropriate K and the most suitable solution based on the empirical results has been proposed in Section 6.4.

Future work is needed to introduce better local tuning of the local variances across the data points, together with a principled approach to allocate points to clusters and for detection of outliers.

A rigurous analysis of time complexity and scalability for complex data sets will also be considered in our future research.

Acknowledgements

The authors thank Prof. David Horn, from the School of Physics and Astronomy (Tel Aviv University, Israel) for his valuable feedback.

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under project with reference number TIN2014-52033-R and by the Spanish Ministry of Education, Culture and Sport under project with reference number PR2015-00217.

References

[1] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Fourth Edition, 4th Edition, Academic Press, 2008.

[2] P. J. G. Lisboa, T. A. Etchells, I. H. Jarman, S. J. Chambers, Finding reproducible cluster partitions for the k-means algorithm, BMC Bioinformatics 14 (Suppl. 1) (2013) S8.

[3] S. J. Chambers, I. H. Jarman, T. A. Etchells, P. J. G. Lisboa, Inference of number of prototypes with a framework approach to k-means clustering, International Journal of Biomedical Engineering and Technology 13 (4) (2013) 323–340.

[4] S. J. Chambers, A framework approach to initialisation dependent clustering, Ph.D. thesis, Liverpool John Moores University, UK (2015).

[5] D. Horn, I. Axel, Novel clustering algorithm for microarray expression data in a truncated svd space, Bioinformatics 19 (9) (2003) 1110–1115.

[6] D. Horn, A. Gottlieb, Algorithm for data clustering in pattern recognition problems based on quantum mechanics, Physical Review Letters 88 (1 – 018702) (2002) 1–4.

[7] D. Horn, A. Gottlieb, The method of quantum clustering, in: Proceedings of Neural Information Processing Systems NIPS 2001, 2001, pp. 769–776.

[8] R. V. Casaña-Eslava, J. D. Martín-Guerrero, I. H. Jarman, P. J. G. Lisboa, Performance assessment of quantum clustering in non-spherical distributions, in: Proceedings of the 24th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2016, pp. 339–344.

[9] J. D. Martín-Guerrero, A. Vellido, P. J. G. Lisboa, Physics and machine learning: Emerging paradigms, in: Proceedings of the 24th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2016, pp. 319–326.

[10] N. Nasios, A. G. Bors, Finding the number of clusters for nonparametric segmentation, in: Computer Analysis of Images and Patterns, 11^{th} International Conference CAIP, LNCS 3691, 2005, pp. 213–221.

[11] N. Nasios, A. G. Bors, Kernel-based classification using quantum mechanics, Pattern Recognition 40 (2006) 875–889.

[12] Y. Li, Y. Wang, Y. Wang, L. Jiao, Y. Liu, Quantum clustering using kernel entropy component analysis, Neurocomputing 202 (2016) 36–48.

[13] R. Shang, Z. Zhang, L. Jiao, W. Wang, S. Yang, Global discriminative-based nonnegative spectral clustering, Pattern Recognition 55 (2016) 172–182.

[14] S. K. Tasoulis, D. K. Tasoulis, V. P. Plagianakos, Random direction divisive clustering, Pattern Recognition Letters 34 (2) (2013) 131–139.

[15] S. Tasoulis, L. Cheng, N. Välimäki, N. J. Croucher, S. R. Harris, W. P. Hanage, T. Roos, J. Corander, Random projection based clustering for population genomics, in: Big Data (Big Data), 2014 IEEE International Conference on, IEEE, 2014, pp. 675–682.

[16] R. Varshavsky, M. Linial, D. Horn, COMPACT: A Comparative Package for Clustering Assessment, in: ISPA Workshops 2005, LNCS 3759, 2005, pp. 159–167.

[17] UCI Machine Learning Repository (July 4th 2016, https://archive.ics.uci.edu/ml/datasets.html).

[18] D. Cook, D. F. Swayne, Interactive and Dynamic Graphics for Data Analysis, Springer Verlag, Berlin, Germany, 2007.

[19] M. Forina, C. Armanino, S. Lanteri, E. Tiscornia, Food Research and Data Analysis, Applied Science Publishers, 1983, Ch. Classification of Olive Oils from their Fatty Acid Composition, pp. 189–214.