On Clustering in Qualitative Spatial and Temporal Reasoning

Abderrahmane Boukontar, Jean-François Condotta, Yakoub Salhi
CRIL CNRS UMR 8188 - Université d’Artois
Lens, France

Abstract

Our understanding of the world is intricately linked to both the spatial arrangement of objects and the timing of events. Knowledge-dependent systems employ mechanisms like Qualitative Spatial and Temporal Reasoning (QSTR) to effectively process and interpret this information. This article explores application of QSTR in data clustering, offering several contributions. These include introducing a formal clustering framework for qualitative data, implementing a satisfiability encoding to compute a clustering, introducing two appropriate distance measures for Qualitative Relation Networks, and experimentally validating through adaptations of k-means and Hierarchical Agglomerative Clustering algorithms.

Introduction

Recognizing patterns within complex data is crucial for understanding intricate information. Clustering, a process rooted in discerning inherent structures based on similarities (Kauffman and Rousseeuw 2009), and deals with diverse data forms such as numeric, graphs (Schaeffer 2007), images (Chang et al. 2017), texts (Aggarwal and Zhai 2012), or symbolic data (de Carvalho, Csernel, and Lechevallier 2009; Boudane et al. 2017; Kejžar, Korenjak-Černe, and Batagelj 2021).

Qualitative Spatial and Temporal Reasoning is an AI symbolic framework that is closely aligned with human reasoning, making it easier to interpret extracted knowledge when dealing with complex data. By employing formalisms like Point Algebra (PA) (Vilain, Kautz, and Van Beek 1990), Interval Algebra (IA) (Allen 1983), and Region Connection Calculus (RCC) (Randell, Cui, and Cohn 1992) via qualitative relations, QSTR effectively models and reasons about spatio-temporal knowledge across various domains.

Utilizing qualitative relations as an intermediary representation, QSTR can endow users with an explanatory capability to uncover the rationale behind cluster formation. This facilitates generating insights from data structures, enhancing understanding of discovered patterns. For similar reasons, qualitative approaches have been employed in the field of data mining and knowledge extraction (Wang et al. 2018; Salhi 2019; Homem et al. 2020; Boukontar, Condotta, and Salhi 2022).

Our contributions in this article are manyfold. First, we introduce a formal framework for accomplishing the clustering task in QSTR. Second, we propose a Boolean Satisfiability (SAT) encoding to verify the feasibility of building a clustering comprising exclusively consistent clusters. Third, we introduce distance measures specific to QSTR for their use with conventional clustering algorithms. Finally, we conclude with an experimental study of our framework using an adaptation of the k-means algorithm.

Data Clustering

Let \( X = \{x_1, x_2, \ldots, x_n\} \) represent a finite non-empty set of items referred to as data instances. In traditional data clustering, each data instance \( x_i \) comprises a vector of \( m \) features \( (x_{i,1}, x_{i,2}, \ldots, x_{i,m}) \) where \( x_{i,j} \in \mathbb{R} \). Clustering data instances aims to maximize similarity within clusters and minimize similarity between them (Aggarwal and Reddy 2013). Similarity, commonly measured using distance metrics like Euclidean and Manhattan distances, evaluates how close or similar data instances are from each other (Sinwar and Kaushik 2014). Similarity can also be expressed as a coefficient (Goyal and Aggarwal 2017) like Simple Matching coefficient, or Hamming coefficient, providing an other perspective by quantifying the number of shared features between two data instances relative to the total number of features. Alternatively, when it is suitable to dismiss the paired absence of features in the data instances, Jaccard’s coefficient emerges as an alternative (Ferdous and others 2009; Irani, Pise, and Phatak 2016).

Hierarchical clustering methods like the widely-used Hierarchical Agglomerative Clustering (HAC) algorithm (Murtagh and Contreras 2012), progressively forms clusters in ascending order. Starting with each data instance in its own cluster, the algorithm iteratively merges the two closest clusters using a chosen linkage technique until all instances belong to a single cluster. Various linkage techniques, such as the single linkage method, which evaluates similarity between two clusters \( P_e, P_f \) as \( \text{dis}_{\text{single}}(P_e, P_f) = \min_{x_i \in P_e, x_j \in P_f} d(x_i, x_j) \) (Yim and Ramdeen 2015; Jarman 2020), consider the minimum...
pairwise distance between data points in the clusters. Other techniques are detailed in (Nielsen 2016).

Partitional clustering methods (Celebi 2014), differ from hierarchical approaches by aiming to create a clustering into a predefined number of clusters \( k \), optimizing a specific objective function to form these \( k \) clusters. One well-known partitional clustering method is the \( k \)-means algorithm (Krishna and Murty 1999). It begins by initializing \( k \) cluster centers, either arbitrarily or using methods studied in the literature (Celebi, Kingravi, and Vela 2013). In each iteration, data instances in \( X \) are assigned to the nearest cluster based on the distance to the center. New cluster centers are then recalculated based on the assigned points. This iterative process continues until convergence, resulting in a clustering into \( k \) clusters.

**Qualitative Spatial and Temporal Formalisms**

Spatial and temporal formalisms are designed to represent and reason about entities like points, lines, intervals, and regions. In this context, let \( B \) be a set of base relations describing relationships on a perceived domain of entities \( \mathcal{O} \). \( B \) forms a partition of \( \mathcal{O} \times \mathcal{O} \), includes the identity relation \( \text{id}_B = \{(x, x) | x \in \mathcal{O}\} \), and is closed under the converse relation \(^{-1} \), where \( b^{-1} = \{(y, x) | (x, y) \in b\} \) for all \( b \in B \). One base relation from set \( B \) can represent definite knowledge between any two entities, while indefinite knowledge can be described by the union of possible base relations between the entities. This union of base relations is referred to a qualitative relation or simply a relation. It will be represented by the set of its included base relations. Thus, the set \( 2^B \) represents the relations in the qualitative formalism based on \( B \). Within the relations of \( 2^B \), the particular relation \( B \) is known as the universal relation representing the absence of any information between two entities. This relation is always satisfied for any pair of entities. On the other hand, the relation \( \{\} \) is called the empty relation and is never satisfied. The set of relations \( 2^B \) is equipped with the usual set-theoretic operations (intersection and union).

For instance, Interval Algebra (IA) (Allen 1983) is a formalism for temporal reasoning using binary relations between time intervals: \{eq, p, i, m, i, o, i, s, i, d, di, f, fi\} (see Figure 1), while Point Algebra (PA) involves binary relations between points on a line (<, =, >).

![Figure 1: The thirteen base relations of IA.](image)

The Region Connection Calculi (RCC) (Randell, Cui, and Cohn 1992) are widely used for spatial qualitative reasoning. It is based on binary topological relations between regions. Two well-known formalisms are the RCC-5 (see Figure 2) and RCC-8 (Bennett 1994).

![Figure 2: The five base relations of RCC-5.](image)

Qualitative information about a set of entities can be modeled as a Qualitative Relation Network (QRN).

A QRN is an ordered pair \( \mathcal{N} = (V, R) \) where \( V \) is a set of (spatial or temporal) entities and \( R \) is a function associating a relation in \( 2^B \) to each element \((i, j)\) of \( V \times V \) s.t. \( R(i, i) = \text{id}_B \) and \( R(j, i) = (R(i, j))^{-1} \).

A scenario is a QRN where \( R(i, j) \) represents exactly one base relation for each \((i, j)\) of \( V \times V \).

Given \( \mathcal{N}_1 = (V, R_1) \) and \( \mathcal{N}_2 = (V, R_2) \), we define \( \mathcal{N}_1 \cap \mathcal{N}_2 \) as the QRN \((V, R)\) where \( R(i, j) = R_1(i, j) \cap R_2(i, j) \) for each \((i, j)\) of \( V \times V \).

A QRN \( \mathcal{N} = (V, R) \) is consistent if there exists a solution \( s \), i.e. a function from \( V \) to the domain \( \mathcal{O} \) such that, for every \((i, j)\) in \( V \times V \), \((s(i), s(j))\) is in \( b \) for some \( b \) in \( R(i, j) \). Hence, the consistency problem is equivalent to the problem of determining whether a QRN has a consistent scenario. Solving the consistency problem is NP-complete in general, whereas is polynomial for the Point Algebra formalism (Van Beek 1992).

A qualitative database (q-database) is defined as a finite set of QRNs.

Figure 3 is an illustration of a qualitative database \( D \) of 4 QRNs describing 4 spatial entities using the RCC-5 formalism. For the sake of simplicity, \( \text{id}_B \) loops on \((R(i, i))\), converse relations and universal relations are omitted, and a node \( i \) is omitted iff for every \( j \) in \( V \setminus \{i\} \), \( R(i, j) = \emptyset \).

![Figure 3: Example of a qualitative database.](image)

**QRN Clustering**

Within a q-database \( D \), our main goal is to identify clusters based on shared solutions.

**Definition 1 (Consistent Partition).** Given a q-database \( D \) and a positive integer \( k \), a consistent \( k \)-partition of \( D \) is a partition \( \mathcal{P} \) of \( D \) s.t. \( |\mathcal{P}| \leq k \) and \( \bigcap_{A \in \mathcal{P}} \mathcal{N} \) is consistent for each \( A \in \mathcal{P} \).

It is important to emphasize that the presence of an inconsistent QRN in \( D \) consequently leads to the absence of any consistent partition of \( D \).
Theorem 2. Let \( Q \) be a QSTR formalism. If the consistency problem in \( Q \) is NP-complete, then the problem of determining whether a q-database admits a consistent \( k \)-partition is NP-complete.

Proof. Clearly, if q-database \( D \) admits a consistent partition with an arbitrary size, then it admits a consistent \( k \)-partition with \( k \leq |D| \). A proof that a q-database \( D \) admits a \( k \)-partition can be set of the form \( \{ (A_1, \sigma_1), \ldots, (A_l, \sigma_l) \} \) where \( l \leq \min(k, |D|) \), \( \{ A_1, \ldots, A_l \} \) is a partition of \( D \), and for every \( 1 \leq i \leq l \), \( \sigma_i \) is a solution of \( \bigcap_{N \in A_i} N \), which is verifiable in polynomial time (the consistency problem is in NP). Hence, the positive instances of the consistent partition problem admit proofs that can be verified in polynomial time. Consequently, the problem of determining whether a q-database admits a consistent \( k \)-partition is in NP.

To show NP-hardness, we only need to observe that a QRN \( N \) is consistent iff the q-database \( \{ N \} \) admits a consistent 1-partition.

\[ \square \]

Definition 3 (QRN Clustering). Given a q-database \( D \) and a positive integer \( k \), a \( k \)-clustering of \( D \) is a set \( C \subset 2^D \) where:

- Exhaustiveness: \( \bigcup_{A \in C} = D \)
- Bounded-Size: \( |C| \leq k \)
- Mutual-Exclusivity: for every \( A, A' \in C \) if \( A \neq A' \) then \( A \cap A' = \emptyset \)
- Consistency. If \( D \) admits a consistent \( k \)-partition, then for every \( A \in C \), \( \bigcap_{N \in A} N \) is consistent.

The Exhaustiveness postulate expresses that every QRN within the database \( D \) should manifest within the clustering. The Bounded-Size postulate expresses that the number of clusters must not surpass \( k \). The Mutual-Exclusivity postulate declares the absence of any two clusters sharing the same QRN. Finally, the Consistency postulates affirm that each cluster maintains consistency, i.e. the QRN defined as the intersection of the QRNs of the cluster is consistent.

The subsequent proposition is a direct consequence of the definitions of consistent partition and QRN clustering.

Proposition 4. If \( C \) is a consistent \( k \)-partition of a q-database \( D \), then \( C \) is \( k \)-clustering of \( D \).

Consistent Partition Computation

The Boolean Satisfiability Problem (SAT) consists in determining whether a CNF formula is satisfiable, i.e. there exists an assignment of truth values to the Boolean variables of the formula that satisfies the formula. A propositional formula in Conjunctive Normal Form (CNF) is a conjunction of clauses where a clause is a disjunction of literals, and a literal is either a variable or the negation of a variable.

We present in this section the SAT encoding of the problem seeking to determine whether a q-database admits a consistent \( k \)-clustering. We emphasize that the use of the path-consistency (PC) method on the QRNs of the q-database, consisting in the calculation of the closure by weak composition can reduce the size of the QRNs by removing basic relations that do not satisfy PC, which can be an improvement when aiming to achieve consistent clusters.

Let \( D = \{ N_1, N_2, \ldots, N_n \} \) be a q-database and \( k \) a positive integer.

To define our encoding for the instance \( (D, k) \), we associate a propositional variables \( x_{ij}^\beta \) with each base relation \( b \in B \) and each ordered pair of variables \( (i, j) \) such that \( i < j \). Moreover, for every QRN \( N_\alpha \), we consider \( k \) additional variables \( y_{b\alpha}^*, y_{b\beta}^* \). Intuitively, if \( y_{b\alpha}^* \) is set to true, then \( N_\alpha \) belongs to the \( \beta \)-th part. This encoding is an adaptation of the one given in (Pham, Thornton, and Sattar 2008).

For each \( \alpha \in \{ 1, \ldots, n \} \) and each \( \beta \in \{ 1, \ldots, k \} \), we use \( \Phi_{\alpha}^\beta \) to denote the conjunction of the following two formulas:

\[ \bigwedge_{i,j \in V} \bigvee_{b \in R_n(i,j)} x_{ij}^b \] (1)

\[ \bigwedge_{i,j \in V} (x_{ij}^b \land x_{jk}^b) \rightarrow \bigvee_{b' \in R_n(i,k) \cap \{ bob' \}} x_{ij}^{b'} \] (2)

for every \( i, j, k \in V \) with \( i < j < k \).

It is wrotting noting that \( \Phi_{\alpha}^\beta \) is satisfiable iff \( N_\alpha \) is consistent in the \( \beta \)-th part.

The first formula of our encoding states that if \( y_{b\alpha}^* \) is true, the QRN \( N_\alpha \) belongs to the \( \beta \)-th consistent part:

\[ y_{b\alpha}^* \rightarrow \Phi_{\alpha}^\beta \] (3)

for every \( \alpha \in \{ 1, \ldots, n \} \) and \( \beta \in \{ 1, \ldots, k \} \).

The second formula ensures that every QRN in the q-database belongs to at least one consistent part:

\[ \bigwedge_{1 \leq \beta \leq k} ( \bigvee_{1 \leq \beta \leq k} y_{b\alpha}^* ) \] (4)

We use ConsPart(\( D, k \)) to denote the encoding that consists of the conjunction of (3) and (4).

Using Formula (3), we have for every \( \alpha \in \{ 1, \ldots, n \} \) and every \( \beta \in \{ 1, \ldots, k \} \), if \( y_{b\alpha}^* \) is true then \( \Phi_{\alpha}^\beta \) is also true, and consequently the QRN \( N_\alpha \) is in the \( \beta \)-th consistent part. Formula (4) implies that every QRN in \( D \) belongs to at least one consistent part, thus, \( D \) admits a consistent \( k \)-partition.

Measuring the distance between two QRNs

In this section, we introduce distance measures designed specifically for QRNs, with the intention of integrating them into established clustering algorithms. Notably, these measures are inspired by the inconsistency measures introduced in (Condotta, Radaoui, and Salhi 2016).

The subsequent measures depend on solving the Partial MaxSAT problem, a variation of the SAT problem involving two sets of clauses, \( \Sigma_1 \) and \( \Sigma_2 \). The objective is to find the largest subset of soft clauses (\( \Sigma_2 \)) that can be satisfied along with all the hard clauses (\( \Sigma_1 \)), where soft clauses are desirable but not mandatory, and hard clauses are clauses that must be satisfied.

Definition 5 (R-Relaxation). Let \( N = (V, R) \) be a QRN. A R-relaxation of \( N \) is a QRN \( N' = (V', R') \) s.t. \( V' = V \) and \( N' \subseteq N \). A R-relaxation is said to be consistent if it corresponds to a consistent QRN.
We use \( \text{ConsRR}(\mathcal{N}) \) to denote the set of consistent \( R \)-relaxations of \( \mathcal{N} \). Moreover, given a \( R \)-relaxation \( \mathcal{N}' \), we use \( \text{diff}(\mathcal{N}, \mathcal{N}') \) to refer to the set of ordered pairs of variables \( \{(i, j) \in V \times V : i < j, \mathcal{N}[i, j] \neq \mathcal{N}'[i, j]\} \). It follows from definition 5 the following distance measure:

- \( D_{rr}(\mathcal{N}, \mathcal{N}') = \min\{\text{diff}(\mathcal{N} \cap \mathcal{N}', \mathcal{N}'') : \mathcal{N}'' \in \text{ConsRR}(\mathcal{N})\} \)

The measure \( D_{rr} \) counts the minimum number of constraints that have to be changed to recover consistency.

We now introduce a Partial MaxSAT encoding for computing \( D_{rr}(\mathcal{N}, \mathcal{N}') \). Let us fix \( \mathcal{N} \cap \mathcal{N}' = (V, R) \). Similarly to \( \text{ConsPart}(\mathcal{D}, k) \), we associate a propositional variable \( x_{ij}^{b} \) with each base relation \( b \in \mathcal{B} \) and each ordered pair of variables \( (i, j) \in V \times V \) such that \( i < j \). Additionally, we associate a distinct propositional variable \( r_{ij} \) with each \( (i, j) \in V \times V \) such that \( i < j \); these variables are used to measure the distance by capturing the elements in \( \text{diff}(\mathcal{N}, \mathcal{N}') \).

The hard part of our encoding is defined as the conjunction of the following formulas:

\[
\bigwedge_{i,j \in V} \bigvee_{b \in \mathcal{B}} x_{ij}^{b} \quad (5)
\]

\[
\bigwedge_{b, b' \in \mathcal{B}} (x_{ij}^{b} \land x_{jk}^{b'} \rightarrow \bigvee_{b'' \in \mathcal{B}} x_{ij}^{b''}) \quad (6)
\]

for every \( i, j, k \in V \) with \( i < j < k \)

\[
\bigwedge_{i,j \in V} (r_{ij} \rightarrow \bigvee_{b \in R(i,j)} x_{ij}^{b}) \quad (7)
\]

The soft part of our encoding consists simply of the following set of unit clauses \( S = \{r_{ij} : i, j \in V \text{ and } i < j\} \). Indeed, by maximizing the number of true elements in \( S \), we maximize the number of constraints that share the same base relations with \( \mathcal{N} \cap \mathcal{N}' \).

Let us now introduce our second distance measure which is based on the notion of variable relaxation.

**Definition 6 (V-Relaxation).** Let \( \mathcal{N} = (V, R) \) be a QRN. A \( V \)-relaxation of \( \mathcal{N} \) is a QRN \( \mathcal{N}' = (V', R') \) s.t. \( V' \subseteq V \) and \( R'(i, j) = R(i, j) \) for every \( i, j \in V \). We use \( \mathcal{N}'_{V} \) to denote the \( V \)-relaxation \( \mathcal{N}' \). A \( V \)-relaxation is said to be consistent if it is a consistent QRN.

The set of consistent \( V \)-relaxations is denoted by \( \text{ConsVR}(\mathcal{N}) \).

The distance measure obtained from the notion of \( V \)-relaxation is defined as follows:

- \( D_{vr}(\mathcal{N}, \mathcal{N}') = \min\{|V \setminus V'| : \mathcal{N}'_{V} \in \text{ConsVR}(\mathcal{N} \cap \mathcal{N}')\} \)

This measure counts the minimum number of variables that have to be removed to get consistency.

Let us describe our Partial MaxSAT encoding for computing \( D_{vr}(\mathcal{N}, \mathcal{N}') \). Let \( \mathcal{N} \cap \mathcal{N}' = (V, R) \). To define our encoding, we use variables of the form \( v_{ij}^{b} \) in addition to the variables of the form \( x_{ij}^{b} \). The variable \( v_{ij} \) is used to know whether the variable \( i \) is ignored or not.

The hard part is defined as the conjunction of the following formulas:

\[
\bigwedge_{i,j \in V} \bigvee_{b \in R(i,j)} x_{ij}^{b} \quad (8)
\]

\[
\bigwedge_{b, b' \in \mathcal{B}} (x_{ij}^{b} \land x_{jk}^{b'} \land v_{i} \land v_{j} \land v_{k} \rightarrow \bigvee_{b'' \in R(i,j) \cap \{b, b'\}} x_{ij}^{b''}) \quad (9)
\]

for every \( i, j, k \in V \) with \( i < j < k \)

The soft part consists of the set of unit clauses \( S = \{v_{i} : i \in V\} \), and by maximizing the true elements of \( S \), we minimise the number of ignored variables.

**Adaptation of the \( k \)-means Algorithm**

We adapt the \( k \)-means algorithm by considering the distance measures introduced in the previous section as measures for similarity between QRNs, and by defining the center of a each cluster \( P \) as the QRN \( P \circ = \bigcap_{\mathcal{N} \in P} N \). The objective is to minimize the inconsistency of the overall clustering. To achieve this, we define the objective function to be minimized as \( \max_{1 \leq l \leq k} \text{inc}_d(P_{l}) \) where \( \text{inc}_d \) measures the inconsistency of \( P_{l} \) using \( D_{rr} \) (resp. \( D_{vr} \)) as the ratio between the minimum number of edges (resp. edges) that must be removed to make \( P_{l} \) consistent.

The \( k \)-means requires an initialization phase of the \( k \) centers. In our context, Algorithm 1 performs this task with the idea of considering the \( k - 1 \) most dispersed QRNs from a QRN \( \mathcal{N}_{i} \) called the QRN initialization seed. The QRN initialization seed is generated randomly and considered as the first center \( P_{1} \), and we proceed iteratively to consider the QRN \( \mathcal{N} \in D \setminus \{P_{1}, P_{2}, \ldots, P_{l-1}\} \) as the center of the cluster \( P_{l} \) if \( \mathcal{N} \) is the most distant QRN from \( \mathcal{N} \setminus P_{l-1} \) with respect to the chosen distance measure being \( D_{rr} \) or \( D_{vr} \).

**Algorithm 1:** Initialization Algorithm

<table>
<thead>
<tr>
<th>Data: A set ( D = {N_{1}, \ldots, N_{n}} ) of QRNs, number of clusters ( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result: A set of centers ( {P_{1}^{o}, P_{2}^{o}, \ldots, P_{k}^{o}} )</td>
</tr>
<tr>
<td>1 ( P_{1}^{o} \leftarrow \text{randomchoice}(D) )</td>
</tr>
<tr>
<td>2 for ( l \leftarrow 2 ) to ( k ) do</td>
</tr>
<tr>
<td>3 ( P_{l}^{o} \leftarrow \text{arg \ max}_{\mathcal{N} \in D} \text{inc}<em>d(\mathcal{N}, P</em>{l-1}) )</td>
</tr>
<tr>
<td>4 end</td>
</tr>
<tr>
<td>5 return ( {P_{1}^{o}, P_{2}^{o}, \ldots, P_{k}^{o}} )</td>
</tr>
</tbody>
</table>

Algorithm 2 initiates with \( k \) centers representing the initial clusters. The completion phase sequentially assigns each QRN \( \mathcal{N} \) not among the initial centers to its nearest cluster, determined by the distance measure (\( D_{rr} \) or \( D_{vr} \)) between the cluster center and \( \mathcal{N} \). After all QRNs in \( D \) are assigned, cluster centers are updated, and the objective function is recalculated. The algorithm then enters a rectification phase, computing distances between each QRN and cluster centers to identify potential moves. Moves are validated with a specific rule: if all QRNs in a cluster are set to change cluster,
only the most distant QRN can move, preventing clusters from becoming black holes which is arising from QRN re-assignments not considering concurrent moves and centers updates occurring after all moves.

The process iterates until the objective function reaches 0 or no further moves can be made, indicating convergence.

Algorithm 2: Clustering Algorithm

<table>
<thead>
<tr>
<th>Data: A set ( D = {N_1, \ldots, N_n} ) of QRNs, number of clusters ( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result: A set of clusters ( {P_1, P_2, \ldots, P_k} )</td>
</tr>
<tr>
<td>for ( N \in D \setminus {P_1, P_2, \ldots, P_k} ) do</td>
</tr>
<tr>
<td>( j = \arg \min_{1 \leq i \leq k} \text{dist}(N, P_i) )</td>
</tr>
<tr>
<td>( P_j \leftarrow P_j \cup {N} )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>( \text{updateCenters}(P_1, P_2, \ldots, P_k) )</td>
</tr>
<tr>
<td>while ( \max_{1 \leq i \leq k} \text{incdist}(P_i) &gt; 0 ) do</td>
</tr>
<tr>
<td>( \text{moves} = \emptyset )</td>
</tr>
<tr>
<td>for ( (N \in D) ) do</td>
</tr>
<tr>
<td>( j = \arg \min_{1 \leq i \leq k} \text{dist}(N, P_i) )</td>
</tr>
<tr>
<td>( \text{moves} \cup {(N, j)} )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>if ( \text{moves} = \emptyset ) then</td>
</tr>
<tr>
<td>break</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>for ( \text{move} = (N, j) \in \text{arcValid}(\text{moves}) ) do</td>
</tr>
<tr>
<td>Move ( N ) to cluster ( P_j )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>( \text{updateCenters}(P_1, P_2, \ldots, P_k) )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>return ( {P_1, P_2, \ldots, P_k} )</td>
</tr>
</tbody>
</table>

Experiments

We conducted experiments with 336 instances, each consisting of 10 QRNs from either \( \text{RCC-5} \) or \( \text{RCC-8} \). The choice of the two formalisms is just to illustrate our framework. Two sets of instances were considered: one with only consistent QRNs (\( \text{COH-instances} \)) and (\( \text{INC-instances} \)) were inconsistent QRNs are present. These QRNs were generated using the \( A(m, d, l) \) model from (Nebel and Renz 2011). The model produces QRNs with \( m \) nodes, each with an average degree of \( d \) and an average relation size of \( l \) on selected edges. We vary \( m \) from 7 to 10, while \( d \) ranges between \( \frac{|m|}{2} - 1 \) and \( \frac{|m|}{2} \) in increments of 0.1, and \( l \) remains fixed at \( \frac{|B|}{2} \) for each formalism. We explore different values of \( k \) ranging from 2 to 6. Experiments were performed on a computer with an Intel i9-Core 2.80 GHz processor, 64 GB of RAM, and coded in Python 3.9.16.

To evaluate the results obtained using the HAC and the \( k \)-means algorithms, we utilized two metrics to assess the intra-cluster resemblance of the QRNs within the same cluster, and, two metrics to evaluate the inter-cluster dissimilarity. The first metric \( mi_1 \), indicates the average number of shared base relations among edges in QRNs within a cluster. A higher \( mi_1 \) value implies a greater symbolic similarity among QRNs in the cluster.

\[
mi_1(P) = \frac{2}{m(m-1)} \sum_{i,j \in V} |R_P(i,j)| \bigcup_{N \in P} R(i,j)
\]

Thus, we evaluate a \( k \)-clustering \( C \) through \( \text{intra}_1(C) = \min_{P \in C} mi_1(P) \). In Figure 4, both the \( k \)-means and HAC algorithms, utilizing distances \( D_{vr} \) and \( D_{rr} \), yield \( k \)-clustering where the QRNs of the least similar cluster exhibit a symbolic similarity of at least 30% on average, as indicated by mean values as black triangles. Notably, the \( k \)-means algorithm, especially with the \( D_{rr} \) distance, generally outperforms. The second metric, denoted as \( mi_2 \), calculates the average number of consistent triplets shared among the triplets of the QRNs of each cluster. We use \( T_N(i,j,k) \) to refer to the set of consistent triplets of base relations of \( N \) defined as \( \{ (b,b',b'') \in R(i,j) \times R(j,k) \times R(i,k) : b'' \in b \land b' \} \).

Thus,

\[
mi_2(P) = \frac{1}{\binom{n}{3}} \sum_{i,j,k \in V} |\bigcup_{N \in P} T_N(i,j,k)|
\]

This metric helps gauge the level of consistency in the triplets of the QRNs within each cluster. We evaluate a \( k \)-clustering \( C \) through \( \text{intra}_2(C) = \min_{P \in C} mi_2(P) \).
In Figure 5, both methods, using distances $D_{vr}$ and $D_{rr}$, yield similar results. The least similar clusters show a minimum of 30% consistent triplets on average.

![Figure 5: Obtained values of $intra_2$.](image)

The third and fourth metrics, $me_1$ and $me_2$, calculate the average number of shared base relations among edges and consistent triplets of base relations among triplets of two cluster centers:

$$me_1(P_\varphi, P_\theta) = \frac{2}{m(m-1)|B|} \sum_{i,j \in V} R_{P_\varphi \cap P_\theta}(i,j)$$

$$me_2(P_\varphi, P_\theta) = \frac{1}{{m \choose 3} |\mathcal{T}|} \sum_{i,j \in V, i<j} T_{P_\varphi \cap P_\theta}(i,j,k)$$

where $\mathcal{T} = \{(b, b', b'') \in B^3 : b'' \in b \circ b'\}$. A lower value of $me_1$ respectively $me_2$ signifies a greater symbolic dissimilarity between the two clusters respectively a lower number of shared consistent triplets, indicating a reduced likelihood that the two clusters exhibit a form of consistency merged together. These two metrics, $me_1$ and $me_2$, offer valuable insights for assessing the divergence of clusters within a $k$-clustering. Thus, other evaluations of a $k$-clustering $\mathcal{C}$ are defined as $inter_1(\mathcal{C}) = \min_{(P_\varphi, P_\theta) \in \mathbb{C} \times \mathbb{C}} me_1(P_\varphi, P_\theta)$ and $inter_2(\mathcal{C}) = \min_{(P_\varphi, P_\theta) \in \mathbb{C} \times \mathbb{C}} me_2(P_\varphi, P_\theta)$.

Figures 6 and 7 reveal that, the two most similar clusters in each $k$-clustering exhibit a maximum symbolic similarity and consistent triplets shared capped to 30% on average.

In addition to the mentioned metrics, we consider the minimum of consistency of $\mathcal{C}_k$, defined as $m_{coh}(\mathcal{C}_k) = 1 - (\max_{P \in \mathbb{C}} inc_{D_{vr}}(P^\circ) \times \max_{P \in \mathbb{C}} inc_{D_{rr}}(P^\circ))^\frac{1}{2}$. This additional metric provide a more comprehensive evaluation of the consistency of the results.

![Figure 6: Obtained values of $inter_1$.](image)

![Figure 7: Obtained values of $inter_2$.](image)

![Figure 8: Values of $m_{coh}$ on instances where consistent partitions are feasible.](image)

In Figure 8, all methods identified consistent $k$-clusterings for at least 100 out of 168 instances with possible consistent partitions. Additionally, when $k$-clusterings are inconsistent, the average minimum consistency value is $m_{coh} = 0.88$ for the least consistent cluster.

**Conclusion**

We have presented a formal framework for QSTR clustering, introducing a SAT encoding to verify the feasibility of consistent partitions. We also proposed QRN-specific distance measures to calculate similarity, and conducted experiments using adapted $k$-means and HAC algorithms using these measures. In future work, we aim to explore other objective functions such as maximizing the number of consistent clusters, develop additional measures to assess QRN similarity and evaluate the quality of a $k$-clustering, and enhance our $k$-means adaptation algorithm by leveraging intuitive rules for QRN relocation.
References


Nielsen, F. 2016. Hierarchical clustering. Introduction to HPC with MPI for Data Science 195–211.


