Hindawi
Journal of Applied Mathematics
Volume 2017, Article ID 4323590, 9 pages
http://dx.doi.org/10.1155/2017/4323590

Research Article

A Greedy Clustering Algorithm Based on Interval Pattern Concepts and the Problem of Optimal Box Positioning

**Stepan A. Nersisyan, Vera V. Pankratieva,
Vladimir M. Staroverov, and Vladimir E. Podolskii**

Faculty of Mechanics and Mathematics, Lomonosov Moscow State University, Leninskie Gory 1, Moscow 119991, Russia

Correspondence should be addressed to Stepan A. Nersisyan; s.a.nersisyan@gmail.com

Received 19 June 2017; Accepted 15 August 2017; Published 25 September 2017

Academic Editor: Dimitris Fotakis

We consider a clustering approach based on interval pattern concepts. Exact algorithms developed within the framework of this approach are unable to produce a solution for high-dimensional data in a reasonable time, so we propose a fast greedy algorithm which solves the problem in geometrical reformulation and shows a good rate of convergence and adequate accuracy for experimental high-dimensional data. Particularly, the algorithm provided high-quality clustering of tactile frames registered by Medical Tactile Endosurgical Complex.

1. Introduction

We consider the problem of clustering, that is, splitting a finite set $X \subset \mathbb{R}^d$ into disjoint subsets (called *clusters*) in such a way that points from the same cluster are similar (with respect to some criterion) and points from different clusters are dissimilar (see, e.g., [1]). It is convenient to present the input data in the form of a numerical context (table) whose rows correspond to objects and columns correspond to attributes of the objects.

Formal concept analysis (FCA) is a data analysis method based on applied lattice theory and order theory. The object-attribute binary relation is visualized with the use of the line diagram of the concept lattice. Within the framework of this theory a formal concept is defined as a pair (extent, intent) obeying a Galois connection (for exact definitions see the monograph [2] by Ganter and Wille).

There exist several generalizations of FCA to fuzzy and numerical contexts. One of them is known as the theory of pattern structures introduced by Ganter and Kuznetsov in [3]. An important particular case of pattern concepts, which are the key object in the theory of pattern structures, is interval pattern concepts with the operation of interval intersection. Interval pattern concepts allow one to apply cluster analysis to rows of formal numerical contexts. In this case the criterion of similarity consists in belonging of all the differences between the values of the corresponding attributes to given intervals.

It can be easily seen that the problem of finding an interval pattern concept of maximum extent size (i.e., cardinality) can be reformulated as the problem of the optimal positioning of a d-dimensional box with given edge lengths for the given set X, that is, finding a position of the box that maximizes the number of points of the set X enclosed by the box (the details are given below in Section 2.2).

The existing algorithms that solve the problem of finding the optimal position of a box do not allow one to obtain an exact or at least approximate solution for high-dimensional data within a reasonable time (see a detailed survey in Section 2.2). The main goal of this paper is to propose a greedy algorithm which gives an approximate solution to this problem and a clustering algorithm based on the optimal positioning problem. We propose a clustering algorithm with

$$O\left(\left(dn\log(n) + \frac{d^3 n^{1-1/d}}{s_{\min}} f(n,d)\right)\frac{n}{c_{\min}}\right) \qquad (1)$$

worst-case time and $O(dn)$ space complexity, where $f(n,d)$ denotes the number of iterations of the main stage of the algorithm, and parameters $s_{\min}$ and $c_{\min}$ regulate the duration of each iteration. Greater number of iterations and greater duration of each iteration provide better approximation.

The rest of the paper is organized as follows. In Section 2 we introduce the main definitions and formalize the statement of the problem. In Sections 3 and 4 we formulate our algorithms. In Sections 5 and 6 we describe the validation results and make some concluding remarks.

2. Main Definitions and Statement of the Problem

In this section we start with the main definitions from the theory of formal concepts and then present a geometrical reformulation of the problem of finding the interval pattern concept of maximum extent size (we call it simply the *maximum interval pattern concept*).

2.1. Formal Concepts. Let us recall the main definitions which we need to formalize our clustering method based on interval pattern concepts. Additional details can be found in [2, 3].

Definition 1. An *upper (lower) semilattice* is a partially ordered set $(M, \leq)$ such that for any elements $x, y \in M$ there exists a unique least upper bound (greatest lower bound, resp.).

Definition 2. A *semilattice operation* on the set M is a binary operation $\sqcap$: $M \times M$ that features the following properties for a certain $e \in M$ and any elements $x, y, z \in M$:

(i) $x \sqcap x = x$ (idempotency).

(ii) $x \sqcap y = y \sqcap x$ (commutativity).

(iii) $(x \sqcap y) \sqcap z = x \sqcap (y \sqcap z)$ (associativity).

(iv) $e \sqcap x = e$.

Definition 3. A *lattice* is an ordered set $(L, \leq)$ which is at the same time an upper and a lower semilattice.

Definition 4. Let $(P, \leq_P)$ and $(Q, \leq_Q)$ be partially ordered sets. A *Galois connection* between these sets is a pair of maps φ: $P \to Q$ and ψ: $Q \to P$ (each of them is referred to as a *Galois operator*) such that the following relations hold for any $p_1, p_2 \in P$ and $q_1, q_2 \in Q$:

(i) $p_1 \leq_P p_2 \Rightarrow \varphi(p_1) \geq_Q \varphi(p_2)$ (anti-isotone property).

(ii) $q_1 \leq_Q q_2 \Rightarrow \psi(q_1) \geq_P \psi(q_2)$ (anti-isotone property).

(iii) $p_1 \leq_P \psi(\varphi(p_1))$ and $q_1 \leq_Q \varphi(\psi(q_1))$ (isotone property).

Applying the Galois operator twice, namely, $\psi(\varphi(p))$ and $\varphi(\psi(q))$, defines a *closure operator*.

Definition 5. A *closure operator* $\overline{(\cdot)}$ on M is a map that assigns a *closure* $\overline{X} \subseteq M$ to each subset $X \subseteq M$ under the following conditions:

(i) $X \leq Y \Rightarrow \overline{X} \leq \overline{Y}$ (monotony).

(ii) $X \leq \overline{X}$ (extensity).

(iii) $\overline{\overline{X}} = \overline{X}$ (idempotency).

Definition 6. A *pattern structure* is a triple $(G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a meet-semilattice of potential object descriptions, and δ: $G \to D$ is a function that associates descriptions with objects.

The Galois connection between the subsets of the set of objects and the set of descriptions for the pattern structure $(G, (D, \sqcap), \delta)$ is defined as follows:

$$A^{\square} := \bigsqcap_{g \in A} \delta(g), \quad \text{where } A \subseteq G,$$
$$d^{\square} := \{g \in G \mid d \sqsubseteq \delta(g)\}, \quad \text{where } A \subseteq G. \tag{2}$$

Definition 7. A *pattern concept of the pattern structure* $(G, (D, \sqcap), \delta)$ is a pair (A, d), where $A \subseteq G$ is a subset of the set of objects and $d \in D$ is one of the descriptions in the semilattice, such that $A^{\square} = d$ and $d^{\square} = A$; A is called the *pattern extent* of the concept and d is the *pattern intent*.

A particular case of a pattern concept is the interval pattern concept. The set D consists of the rows of a numerical context, which are treated as tuples of intervals of zero length. An interval pattern concept is a pair (A, d), where A is a subset of the set of objects and d is a tuple of intervals with ends determined by the smallest and the largest values of the corresponding component in the descriptions of all objects in A.

Interval pattern concepts are convenient to use in the analysis of numerical contexts, when there is a need to divide all data into clusters that comprise objects in which the numerical data is similarly "distributed" in the rows.

For each component of an interval pattern concept we introduce the width σ: the difference between the largest and the smallest values of the component. Then a clustering procedure can be defined using a standard greedy approach. Specifically, at each step the maximum interval pattern concept is identified, that is, an interval pattern concept with the maximum number of objects, whose width with respect to each component does not exceed a predefined σ. The objects of the identified interval pattern concept are combined into a cluster and excluded from the set of objects analyzed at subsequent steps.

In Example 1 presented in Table 1 the objects are pupils and the numerical data of the context consist of the grades they got at exams in various disciplines.

We need to divide the set of pupils into clusters in such a way that the grades of pupils in the same cluster differ by at most 1 for each of the disciplines. Such a setting corresponds to $\sigma = 1$; in this case we obtain 6 clusters (interval pattern concepts whose width is not greater than 1), each containing one pupil. In the case $\sigma = 2$ we arrive at the same 6 clusters.

When $\sigma = 3$ we have five clusters $\{A, D\}^{\square} = \{[8, 9], [9, 9], [9, 10], [6, 9]\}$, $\{B\}$, $\{C\}$, $\{E\}$, $\{F\}$, and in the case $\sigma = 4$ we obtain three clusters $\{A, C, D\}^{\square} = \{[6, 9], [5, 9], [9, 10], [6, 9]\}$, $\{B, E\}^{\square} = \{[8, 8], [2, 4], [6, 6], [5, 9]\}$, $\{F\}$.

Table 1: A fuzzy formal context, where the objects are pupils and the attributes are disciplines.

	Arts	Mathematics	Computer science	Sports
A	9	9	10	9
B	8	2	6	5
C	6	5	10	7
D	8	9	9	6
E	8	4	6	9
F	6	5	2	10

Example 2. In the conditions of the previous example let us set $\sigma_1 = 1$, $\sigma_2 = 1$, $\sigma_3 = 10$, $\sigma_4 = 3$. Then the set of pupils can be divided into four clusters $\{A, D\}$, $\{C, F\}$, $\{B\}$, $\{E\}$:

$$\begin{gathered}\{A, D\}^{\square} = \{[8,9], [9,9], [9,10], [6,9]\}, \\ \{C, F\}^{\square} = \{[6,6], [5,5], [2,10], [7,10]\}, \\ \{B\}^{\square} = \{[8,8], [2,2], [6,6], [5,5]\}, \\ \{E\}^{\square} = \{[8,8], [4,4], [6,6], [9,9]\}.\end{gathered} \tag{3}$$

Clustering methods based on interval pattern concepts find applications in the analysis of experimental data. For instance, applications of such methods to gene expression analysis were discussed in [4, 5].

2.2. Geometry. Let P be a set of n points in $\mathbb{R}^d$ ($d \in \mathbb{N}$) and $\delta_1, \delta_2, \ldots, \delta_d$ be a set of positive real numbers.

Definition 8. A d-orthotope (also called a box) with center $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and edge lengths $\delta_1, \delta_2, \ldots, \delta_d$ is the Cartesian product of the intervals

$$\left[x_1 - \frac{\delta_1}{2}, x_1 + \frac{\delta_1}{2}\right] \times \cdots \times \left[x_d - \frac{\delta_d}{2}, x_d + \frac{\delta_d}{2}\right]. \tag{4}$$

It can be easily seen that the problem of identification of maximum interval pattern concept can be reformulated in terms of finding the *optimal* position of the box with the edge lengths $\delta_1, \delta_2, \ldots, \delta_d$, that is, maximizing the number of points of set P enclosed by the box. This formulation can be generalized to the problem of finding the optimal position of a ball in an arbitrary metric space, since any box can be treated as a ball in the stretched L_∞ metric in which the distance $\rho(x, y)$ between the points $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$ is defined as

$$\rho(x, y) = \max_{1 \le i \le d} \delta_i^{-1} |x_i - y_i|. \tag{5}$$

The problem of optimal positioning has been well studied for $d = 2$: some lower and sharp upper bounds on complexity are known (see, e.g., [6, 7]). However, to the best of our knowledge for the case of an arbitrary dimension d no lower bounds and no efficient exact algorithms are available so far. de Figueiredo and da Fonseca noted [8] that the problem can be solved exactly in roughly $O(n^{d+1/d})$ time by projecting the points onto a $(d + 1)$-dimensional paraboloid and using half-space range searching data structures [9]. In the same paper for the case of weighted points under certain additional restrictions they also obtained a lower bound $\Omega(n^d)$ for exact algorithms and indicated that existing algorithms for the unweighted version of the problem do not beat this lower bound in the worst case. Eckstein et al. showed that a generalization of the problem of optimal positioning whose input also includes a set of prohibited points is NP-hard [10].

Known approximate algorithms for optimal positioning also have time complexity which depends on d exponentially. For example, de Figueiredo and da Fonseca suggested an approximate algorithm [8] which solves the problem in worst-case time $O(3^d n/\varepsilon^{d-1})$, where $0 < \varepsilon < 1$ is a given approximation parameter. Due to exponential dependence on d these approximate algorithms are also practically inapplicable in the case of high dimension, and there is a need to develop an algorithm which can produce an approximate solution in reasonable time.

3. A Greedy Algorithm for Finding an Approximately Optimal Position of a Box

In this section we present a greedy algorithm for finding an approximately optimal position of a box with edge lengths $\delta_1, \delta_2, \ldots, \delta_d$ for a set $P = \{p_i\}_{i=1}^n \subset \mathbb{R}^d$ (the order in which points are listed in P is insignificant). This algorithm is auxiliary for the clustering method described in Section 4.

The algorithm has several input parameters: positive real numbers s, $s_{\min}$, $\lambda < 1$, and a function $f\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. The parameters s, $s_{\min}$, and λ regulate the duration of one iteration. The function f takes the values n and d as inputs and returns the number of iterations at the main stage of the algorithm. Greater number of iterations and greater duration of each iteration provide better approximation.

The algorithm includes two basic stages: the preprocessing stage and the main stage.

3.1. Preprocessing

(1) At the first stage of our algorithm the box with the edge lengths $\delta_1, \delta_2, \ldots, \delta_d$ is transformed into the *unit cube* (we call it simply the *cube*) by means of dividing the ith coordinate of each point by δ_i, $i = 1, \ldots, d$. This stage can be performed in $O(dn)$ operations.

(2) We consider the integer lattice with edges of length 1, compute the number of points of P in each cell, and denote the cell that contains the maximum number of points by C_0. The cell C_0 is called the *base cube*. Let $y_0 \in \mathbb{R}^d$ denote the center of C_0. This stage requires $O(dn)$ operations as well.

(3) At the final step of the preprocessing stage we build a k-d tree data structure (which is used at the main stage to organize the fast range search) in $O(dn \log(n))$ operations with the space complexity of $O(dn)$ (see [11, 12]).

3.2. The Main Stage. Let $q: 2^{\mathbb{R}^d} \to \mathbb{Z}^+$ denote the function which counts number of points of the set P in an arbitrary subset of $\mathbb{R}^d$. The main idea of our algorithm consists in constructing a finite sequence of cubes that starts from a random point y in the base cube and satisfies the condition that the next cube contains more points than the previous one. Let $D_1^y, \ldots, D_{k(y)}^y$ denote these cubes with centers $z_1^y, \ldots, z_{k(y)}^y$, respectively. In our notation we have $z_1^y = y$ and $q(D_i^y) < q(D_{i+1}^y)$ for all $i \in \{1, \ldots, k(y)-1\}$. After $f(n, d)$ iterations the algorithm returns a locally optimal cube C.

Definition 9. The *t-neighborhood* of a cube D with center $x = (x_1, \ldots, x_d)$ is the set consisting of all cubes with centers at points of the form $(x_1, \ldots, x_{i-1}, x_i \pm t, x_{i+1}, \ldots, x_d)$ for all $i \in \{1, \ldots, d\}$, that is, all cubes obtained through translation of D along one of the axes by the distance $\pm t$.

Now we describe the procedure of constructing the sequence of cubes. Let y be an arbitrary point in the base cube C_0 and $z_1^y = y$, D_1^y be the cube with center at z_1^y, $s_1 = s$. In order to get a definite estimate on the precision of the algorithm (see Theorem 11) we initialize the first iteration deterministically by taking the center of C_0 as y. Other iterations are initialized randomly.

Suppose that the cubes $D_1^y, \ldots, D_m^y$ with centers $z_1^y, \ldots, z_m^y$, respectively, and the numbers $s_1, \ldots, s_m$ have been already constructed. There are two possible cases.

(1) If there exists a cube D in the s_m-neighborhood of D_m^y such that $q(D) > q(D_m^y)$, then we set $D_{m+1}^y = D$, take the center of D as z_{m+1}^y, and take $s_{m+1} = s_m$. In other words, if there exists a cube in the s_m-neighborhood of the current cube which contains more points of P, then we move the current cube to this position.

(2) If there are no such cubes (i.e., all cubes in the s_m-neighborhood of the current cube contain at most the same number of points), then we set $D_{m+1}^y = D_m^y$, $z_{m+1}^y = z_m^y$, and $s_{m+1} = \lambda s_m$ (i.e., decrease the current step size). If $s_{m+1} < s_{\min}$ (the step size threshold is reached), then the procedure is ended and D_m^y is returned as the procedure result.

In order to obtain acceptable time complexity we impose additional restrictions on the selection of the next cube. These assumptions are necessary to avoid the situation where the length of the sequence grows exponentially with d. Validation on experimental data confirmed that these restrictions do not essentially affect the clustering results.

Restriction 1. All cubes in the sequence must have common points with the base cube C_0.

In Figure 1 we present an example of a set P for which this requirement causes a significant difference between the exact solution and the solution produced by the algorithm. However, this difference is essentially reduced at further steps of the clustering algorithm as generally it affects only the order in which clusters are constructed.

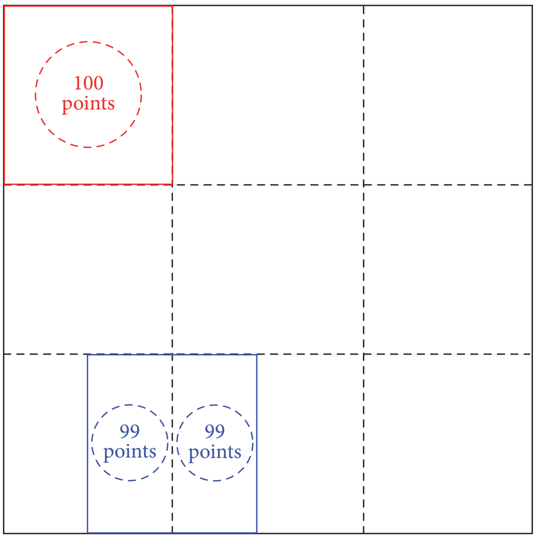


Figure 1: The base cube is colored red; the global optimum is blue. There is no way to move from the red cube to the blue one without losing touch with the base cube.

Restriction 2. For each individual coordinate it is not allowed to translate the cube in the opposite directions at different steps of the procedure described above.

The above restrictions lead to the following lemma.

Lemma 10. *The main stage of the algorithm has*

$$O\left(\frac{d^3 n^{1-1/d}}{s_{\min}} f(n, d)\right) \tag{6}$$

worst-case time complexity.

Proof. First we get an upper estimate for the length $k(y)$ of the sequence of cubes (for an arbitrary $y \in C_0$). Due to Restrictions 1 and 2 we have

$$k(y) \le O\left(\frac{d}{s_{\min}}\right). \tag{7}$$

Thus, Restrictions 1 and 2 avoid the situation where the length of the sequence grows exponentially with d. Each step of the procedure of constructing the sequence of cubes requires $2d$ evaluations of the function q for the cubes (i.e., $2d$ range searches). With the use of a k-d tree the range search can be performed with $O(dn^{1-1/d})$ worst-case time complexity (see [13]). The procedure of constructing the sequence of cubes involves $f(n, d)$ iterations, so the above complexity bound holds. □

Note that we also have a trivial estimate $k(y) \le n$, as $q(D_m^y)$ grows, and hence $k(y) \le q(D_{k(y)}^y) \le n$. Thus, without the imposed restrictions the worst-case complexity estimate

$$O\left(d^2 n^{2-1/d} f(n, d)\right) \tag{8}$$

holds, and hence the restrictions can be omitted without violation of practical feasibility in case if the number of objects n has the same order as the dimensionality d.

3.3. Precision and Complexity of the Algorithm

Theorem 11. *Let y_0 be a center of C_0, $D_{alg} = D^{y_0}_{k(y_0)}$ be a cube produced by an algorithm iteration which was initialized with y_0 (and so for this iteration $D^{y_0}_1 = C_0$), and D_{opt} be an optimal cube (i.e., $D_{opt} \in \arg\max q(D)$, where maximum is taken over all unit cubes in $\mathbb{R}^d$). Then*

$$\frac{1}{2^d} \leq \frac{q\left(D_{alg}\right)}{q\left(D_{opt}\right)} \leq 1, \tag{9}$$

and this estimate is sharp.

Proof. The upper estimate is trivial. The lower estimate follows from the fact that D_{opt} is covered by at most 2^d cells of the integer lattice with edges of length 1, and hence

$$q\left(D_{\text{opt}}\right) \leq 2^{-d} q\left(C_0\right) \leq 2^{-d} q\left(D_{\text{alg}}\right). \tag{10}$$

An example that shows that the estimate is sharp is similar to the example from Figure 1. For example, we can locate the center of D_{opt} at the integer lattice node and put 2^d points in D_{opt} in such a way that each cell of the integer lattice contains at most one of these points. Then, we select an arbitrary cell of the integer lattice that is distant from D_{opt} and put one point to this cell, which becomes C_0. □

Theorem 12. *The algorithm for finding an approximately optimal position of the box has*

$$O\left(dn\log(n) + \frac{d^3 n^{1-1/d}}{s_{\text{min}}} f(n,d)\right) \tag{11}$$

worst-case time complexity and $O(dn)$ space complexity.

Proof. Combining the estimates for the time and space complexity of the preprocessing stage and the main stage of the algorithm gives the bounds mentioned above. □

Note that omitting Restrictions 1 and 2 results in the worst-case time complexity estimate

$$O\left(d^2 n^{2-1/d} f(n,d)\right). \tag{12}$$

4. Clustering Algorithm

Now let us consider the clustering problem, that is, the problem of splitting the given set $P = \{p_i\}_{i=1}^n \subset \mathbb{R}^d$ into mutually disjoint subsets $C_1, \ldots, C_k$. Following interval pattern concept approach, we construct clusters with controlled interval pattern concept width. We propose a clustering algorithm based on the greedy approach and the procedure for finding an approximately optimal position of a box described in Section 3. The algorithm is not sensitive to the order in which points P are given. The parameters of the algorithm include positive real numbers $\delta_1, \delta_2, \ldots, \delta_d$ and all parameters of the positioning algorithm, namely, s, s_{min}, λ, and $f(n,d)$.

First, we put $P_1 = P$ and find an approximately optimal position D_1 of the box with the edge lengths $\delta_1, \ldots, \delta_d$ for the set P_1. Now suppose that the sets $D_1, \ldots, D_i$ and $P_1, \ldots, P_i$ have been already constructed and let $P_{i+1} = P_i \setminus D_i$. If $P_{i+1} = \varnothing$ then the procedure is ended. Else we find an approximately optimal position D_{i+1} of the box for the set P_{i+1}. The output of this procedure is a set of clusters $C_i = P_i \cap D_i$.

In order to avoid producing a lot of small clusters consisting of outliers we impose one more restriction.

Restriction 3. The resulting clusters must include at least c_{min} objects.

With this restriction if the size of $P_{i+1} \cap D_{i+1}$ is less than c_{min} then the procedure ends (and points belonging to $P \setminus (C_1 \cup \cdots \cup C_i)$ are considered unclustered and referred to as outliers).

Restriction 3 together with Theorem 12 immediately leads to the following theorem.

Theorem 13. *The clustering algorithm has*

$$O\left(\left(dn\log(n) + \frac{d^3 n^{1-1/d}}{s_{\text{min}}} f(n,d)\right) \cdot \frac{n}{c_{\text{min}}}\right) \tag{13}$$

worst-case time complexity and $O(dn)$ space complexity.

If Restrictions 1–3 are omitted, the worst-case time complexity estimate is

$$O\left(d^2 n^{3-1/d} f(n,d)\right). \tag{14}$$

5. Validation

Validation of the clustering algorithm developed in this study was performed on a dataset of tactile images registered by the Medical Tactile Endosurgical Complex (MTEC) during examination of artificial samples. MTEC allows intraoperative mechanoreceptor tactile examination of tissues and is already used in endoscopic surgery [14–16]. As methods for automated analysis of medical tactile images are still insufficient, validation results in particular and the developed clustering algorithm in general provide new opportunities for the medical domain applications.

The key component of MTEC is a tactile mechanoreceptor [17, Fig. 1]. Its operating head is equipped with 19 pressure sensors that perform synchronous measurements 100 times per second. Each measurement result (called "a tactile frame" and consisting of 19 values) is wirelessly transmitted to a computer that performs preprocessing and visualization. The sensors are located at the operating head surface which is a circle with diameter 20 mm.

In order to create a dataset of tactile images we utilized MTEC for tactile examinations of three types of artificial samples. The samples were similar to the L-samples utilized in the study [17]—they were made using a soft silicone (Ecoflex 00-10, Shore hardness 00-10A) according to manufacturer's instructions and had a shape of a rectangular block with length, width, and height of 40 mm, 35 mm, and 11 mm, respectively. The difference was in sizes and shapes of hard inclusions enclosed in the samples. For the first sample type (ST1) the inclusion had a form of a spherical cap with base diameter 8 mm and height 2.4 mm oriented for palpation from the convex side. For the second sample type (ST2) the inclusion had a form of a spherical cap with a base diameter 4.7 mm and height 1.7 mm also oriented for palpation from the convex side. For the third sample type (ST3) the inclusions were the same as for ST2, but they were oriented for palpation from the flat side. For all sample types the inclusions were located in the center at height of approximately 3 mm. Thus, sample types were similar with a difference in either size or convexity of the inclusion. These samples simulated tissue with malignant neoplasms.

Totally 55 tactile examinations of the described samples were performed using MTEC. The contact angle was kept approximately equal to 90°, and inclusions were located close to the center of the operating head surface. We performed twenty-two, seventeen, and sixteen examinations for samples of ST1, ST2, and ST3 types, respectively. For each examination one tactile frame was selected, namely, the one with the largest standard deviation (SD) of values, and other tactile frames were disregarded. Visualization of tactile frames for each sample type is presented in Figures 2(a)–2(c).

Thus, each examination was associated with a point in $\mathbb{R}^{19}$, and the total number of points was 55. This set of points was clustered using the developed clustering algorithm, and the results were compared with the results of k-means clustering ($k = 3$, Euclidean distance; see, e.g., [1]), which was used as a reference. Scikit-learn implementation [18] of k-means algorithm was utilized. Adjusted and raw Rand indexes (clustering result versus original classes; see, e.g., [1]) were used as compared characteristics of the clustering quality. Note that both clustering algorithms use random initialization, so multiple runs were performed for clustering quality estimation (specifically, 100 runs were performed to estimate Rand index for each algorithm with given parameters).

The results produced by both the proposed algorithm and by the k-means algorithm were unsatisfactory. However, the poor quality of the resulting clustering was predictable as examining of a single sample can result in tactile frames that are essentially different with respect to representation by a point in $\mathbb{R}^{19}$ due to rotation and slight shifts of a tactile mechanoreceptor.

To get better results we mapped the data to the new 9-dimensional space of attributes. The new attributes included

(i) SD of all values in a tactile frame;

(ii) mean and SD of the values corresponding to 7 middle sensors;

(iii) mean and SD of the values corresponding to 12 outer sensors;

(iv) mean and SD of the values corresponding to sensors that belong to the main diagonals (3 diagonals each consisting of 5 sensors, 13 sensors in total; see Figure 2(d) for details);

(v) mean and SD of the values corresponding to sensors that belong to the secondary diagonals (6 diagonals each consisting of 4 sensors, 12 sensors in total; see Figure 2(d) for details).

These attributes are robust to rotations proportional to 60°. The values of mean and SD were computed after scaling the values to [0, 1] segment.

Transition to the new attribute space essentially improved the clustering quality, but our algorithm left 10–14 points as outliers ($c_{\min}$ was set equal to 8; the values of s, $s_{\min}$, and λ were set equal to 0.9, 0.3, and 0.8 respectively, and σ was set equal to 0.27 for all attributes). A representative result of one run is presented in Table 2. Then we placed outliers points to the obtained clusters by the k-nearest neighbors algorithm ($k = 8$, unweighted; see, e.g., [19]). A representative result for one run is presented in Table 3.

Table 2: Correspondence between the original classes and the clusters constructed by the proposed algorithm (with outliers).

	1st cluster 9 points	2nd cluster 13 points	3rd cluster 22 points	Unclustered 11 points
ST1 22 points	9 points	1 points	5 points	7 points
ST2 17 points	0 points	12 points	3 points	2 points
ST3 16 points	0 points	0 points	14 points	2 points

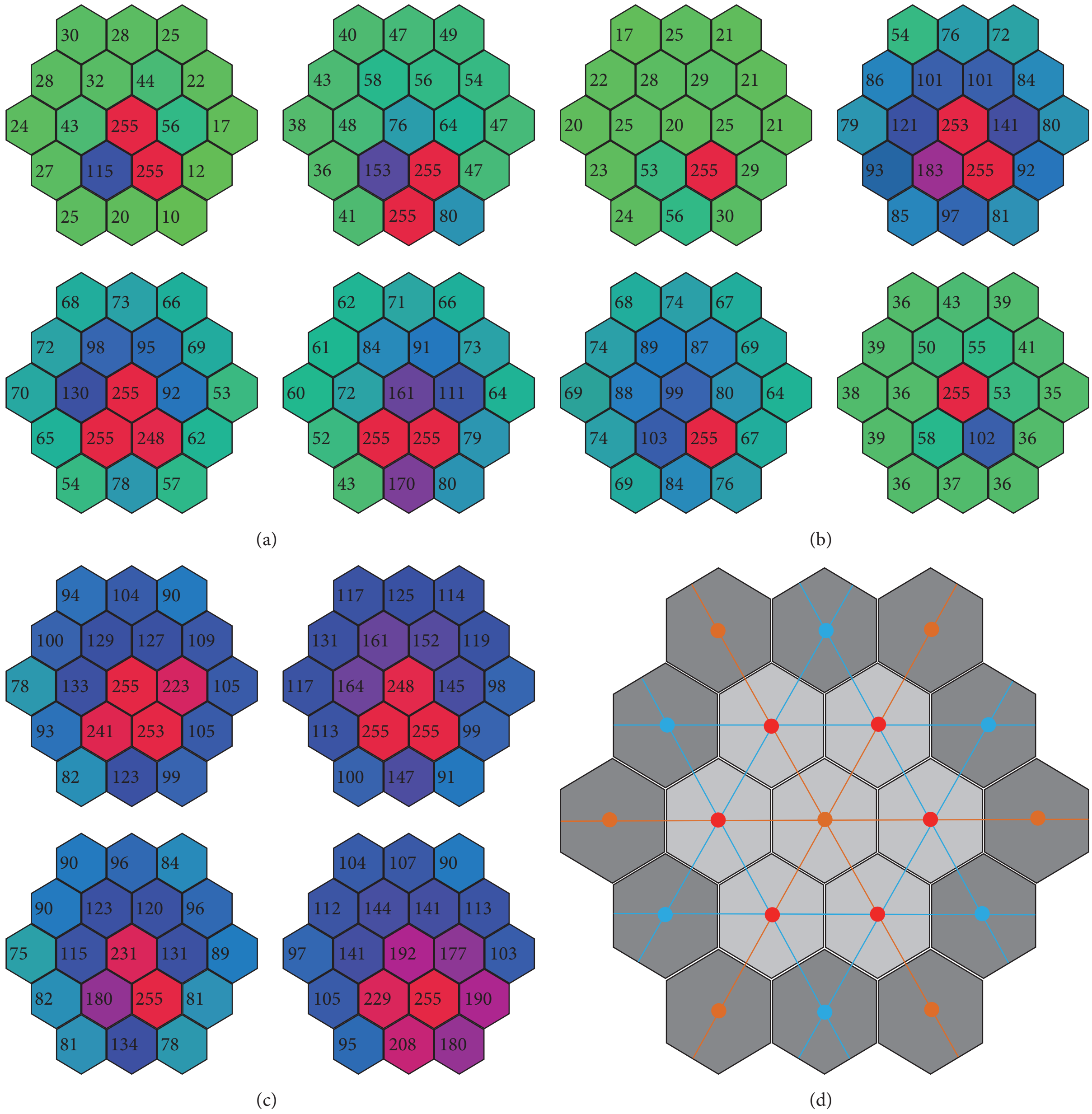
Table 3: Correspondence between the original classes and the clusters constructed by the proposed algorithm (no outliers).

	1st cluster 11 points	2nd cluster 17 points	3rd cluster 27 points
ST1 22 points	11 points	3 points	8 points
ST2 17 points	0 points	14 points	3 points
ST3 16 points	0 points	0 points	16 points

Table 4 contains mean values and SDs for Rand indices and timing information.

As one can see, the proposed algorithm has an acceptable running time, and both our and k-means algorithm reach mean quality plateau already at 20 iterations.

The advantage of the proposed algorithm over the k-means algorithm with respect to the clustering quality was statistically significant. For example, for 20 iterations and adjusted Rand index the comparison of our algorithm with outliers and the k-means on 100 runs resulted in Mann–Whitney U-test two-tailed p value equal to $1.0 \cdot 10^{-10}$. As outliers are the points that are the most difficult for clustering, the advantage of our algorithm complemented by kNN-attributing of outliers to clusters over the k-means was lower but still firmly significant with Mann–Whitney U-test two-tailed p value equal to $9.5 \cdot 10^{-4}$.



(a)

(b)

(c)

(d)

Figure 2: (a–c) Examples of tactile frames for examinations of ST1 (a), ST2 (b), and ST3 (c) type samples. Pressure values are scaled to $[0, 255]$ segment and color-coded. (d) Correspondence between sensors and attributes from the new attribute space. Each hexagon represents one sensor. Middle sensors are colored in light-gray, outer sensors are colored in dark-gray. The main diagonals are shown by orange lines; the secondary diagonals are shown by blue lines. Centers of the hexagons that represent sensors belonging to both main and secondary diagonals are colored in red; belonging only to main diagonals, in orange; belonging only to secondary diagonals, in blue.

Interestingly, the transition to the new attribute space improved the quality of our algorithm more than the quality of the k-means clustering. For example, for 20 iterations, adjusted Rand index, and 100 runs, the comparison of the clustering quality for the initial attribute space and the new attribute space resulted in Mann–Whitney U-test two-tailed p values not exceeding 10^{-12} for both "with outliers" and "no outliers" versions of our algorithm, while for k-means the p value was 0.43.

6. Conclusions

In this paper we proposed a greedy clustering algorithm based on interval pattern concepts. The obtained theoretical estimate on algorithm complexity proved computational feasibility for high-dimensional spaces, and the validation on experimental data demonstrated high quality of the resulting clustering in comparison with conventional clustering algorithms such as k-means.

Table 4: Dependency of Rand index values and the running time for our and k-means clustering methods on number of iterations performed (100 program runs for each value). Values of Rand index are presented in terms of medians and interquartile ranges (IQR).

Number of iterations	Clustering method	Rand index median (adjusted/raw)	Rand index IQR (adjusted/raw)	Average running time (in seconds)
20	Our method (with outliers)	0.43/0.73	0.12/0.05	0.8
	Our method (no outliers)	0.39/0.73	0.10/0.04	0.8
	k-means	0.32/0.70	0.21/0.09	0.02
50	Our method (with outliers)	0.43/0.73	0.08/0.05	2.4
	Our method (no outliers)	0.39/0.73	0.06/0.03	2.5
	k-means	0.27/0.68	0.20/ 0.09	0.05
100	Our method (with outliers)	0.42/0.74	0.08/0.04	4.2
	Our method (no outliers)	0.39/0.73	0.06/0.03	4.3
	k-means	0.31/0.70	0.20/0.09	0.09

Particular results obtained during validation, such as a new attribute space for tactile frames registered by the Medical Tactile Endosurgical Complex, have individual significance as they provide new opportunities for the medical domain applications aimed at automated analysis of tactile images.

Data Access

Dataset of tactile frames used for the validation and the Python script that implements the developed clustering algorithm are available upon request from the authors.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank Dr. Alexey V. Galatenko and Dr. Vladimir V. Galatenko for valuable comments and discussions. The research was supported by the Russian Science Foundation (Project 16-11-00058 "The Development of Methods and Algorithms for Automated Analysis of Medical Tactile Information and Classification of Tactile Images").

References

[1] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster analysis*, Wiley Series in Probability and Statistics, John Wiley & Sons, Chichester, UK, Fifth edition, 2011.

[2] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer, Berlin, Germany, 1999.

[3] B. Ganter and S. O. Kuznetsov, "Pattern Structures and Their Projections," in *Conceptual Structures: Broadening the Base*, vol. 2120 of *Lecture Notes in Computer Science*, pp. 129–142, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[4] M. Kaytoue, S. Duplessis, S. O. Kuznetsov, and A. Napoli, "Two fca-based methods for mining gene expression data," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5548, pp. 251–266, 2009.

[5] M. Kaytoue, S. O. Kuznetsov, A. Napoli, and S. Duplessis, "Mining gene expression data with pattern structures in formal concept analysis," *Information Sciences. An International Journal*, vol. 181, no. 10, pp. 1989–2001, 2011.

[6] B. Aronov and S. Har-Peled, "On approximating the depth and related problems," *SIAM Journal on Computing*, vol. 38, no. 3, pp. 899–921, 2008.

[7] B. M. Chazelle and D. T. Lee, "On a circle placement problem," *Computing. Archives for Scientific Computing*, vol. 36, no. 1-2, pp. 1–16, 1986.

[8] C. M. de Figueiredo and G. D. da Fonseca, "Enclosing weighted points with an almost-unit ball," *Information Processing Letters*, vol. 109, no. 21-22, pp. 1216–1221, 2009.

[9] J. Matoušek, "Range searching with efficient hierarchical cuttings," *Discrete & Computational Geometry*, vol. 10, no. 1, pp. 157–182, 1993.

[10] J. Eckstein, P. L. Hammer, Y. Liu, M. Nediak, and B. Simeone, "The maximum box problem and its application to data analysis," *Computational Optimization and Applications. An International Journal*, vol. 23, no. 3, pp. 285–298, 2002.

[11] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[12] J. H. Freidman, L. B. Jon, and A. F. Raphael, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 209–226, 1977.

[13] D. T. Lee and C. K. Wong, "Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees," *Acta Informatica*, vol. 9, no. 1, pp. 23–29, 1977.

[14] V. Sadovnichy, R. Gabidullina, M. Sokolov, V. Galatenko, V. Budanov, and E. Nakashidze, “Haptic device in endoscopy,” in *Proceedings of the 21st Medicine Meets Virtual Reality Conference, NextMed/MMVR 2014*, pp. 365–368, USA, February 2014.

[15] V. Barmin, V. Sadovnichy, M. Sokolov, O. Pikin, and A. Amiraliev, “An original device for intraoperative detection of small indeterminate nodules,” *European Journal of Cardio-thoracic Surgery*, vol. 46, no. 6, pp. 1027–1031, 2014.

[16] R. F. Solodova, V. V. Galatenko, E. R. Nakashidze et al., “Instrumental tactile diagnostics in robot-assisted surgery,” *Medical Devices: Evidence and Research*, vol. 9, pp. 377–382, 2016.

[17] V. M. Staroverov, V. V. Galatenko, T. V. Zykova, Y. I. Rakhmatulin, D. V. Rukhovich, and V. E. Podol’skii, “Automated real-time correction of intraoperative medical tactile images: sensitivity adjustment and suppression of contact angle artifact,” *Applied Mathematical Sciences*, vol. 10, pp. 2831–2842, 2016.

[18] F. Pedregosa, G. Varoquaux, and A. Gramfort, “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[19] T. Mitchell, *Machine Learning*, McGraw Hill, New York, NY, USA, 1997.