

# Constrained clustering with a complex cluster structure

Marek Śmieja<sup>1</sup> · Magdalena Wiercioch<sup>1</sup>

Received: 9 May 2014 / Revised: 6 May 2016 / Accepted: 10 May 2016 /

Published online: 24 May 2016

© The Author(s) 2016. This article is published with open access at Springerlink.com

**Abstract** In this contribution we present a novel constrained clustering method, Constrained clustering with a complex cluster structure (C4S), which incorporates equivalence constraints, both positive and negative, as the background information. C4S is capable of discovering groups of arbitrary structure, e.g. with multi-modal distribution, since at the initial stage the equivalence classes of elements generated by the positive constraints are split into smaller parts. This provides a detailed description of elements, which are in positive equivalence relation. In order to enable an automatic detection of the number of groups, the cross-entropy clustering is applied for each partitioning process. Experiments show that the proposed method achieves significantly better results than previous constrained clustering approaches. The advantage of our algorithm increases when we are focusing on finding partitions with complex structure of clusters.

**Keywords** Constrained clustering · Model-based clustering · Mixture of models · Pairwise equivalence constraints · Semi-supervised learning · Cross-entropy clustering

**Mathematics Subject Classification** Primary 68T Computer Science; Secondary 62H30

---

The work of the M. Śmieja was supported by the National Science Centre (Poland) Grant No. 2014/13/N/ST6/01832, while the work of the M. Wiercioch was supported by the National Science Centre (Poland) Grant No. 2014/13/B/ST6/01792.

---

✉ Magdalena Wiercioch  
magdalena.wiercioch@ii.uj.edu.pl

Marek Śmieja  
marek.smieja@ii.uj.edu.pl

<sup>1</sup> Faculty of Mathematics and Computer Science, Jagiellonian University,  
Łojasiewicza 6, 30-348 Kraków, Poland

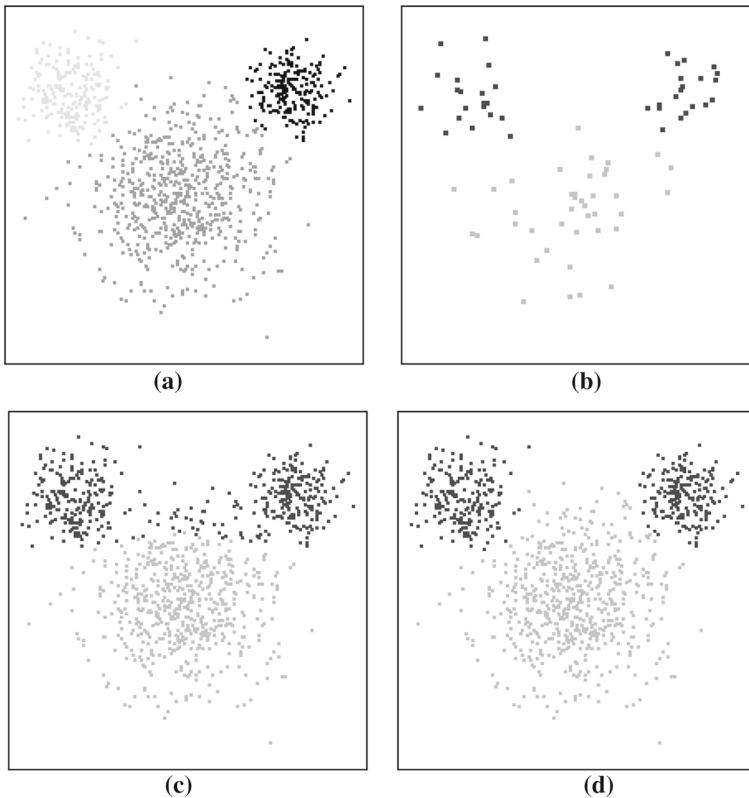
## 1 Introduction

Clustering is one of the most important and efficient tools for processing massive amounts of data (Xu and Wunsch 2005). For this reason, it is commonly used in many fields of computer science including data mining, pattern recognition, machine learning and data compression (Pavel 2002; Jain et al. 1999; Hruschka et al. 2009; Śmieja and Tabor 2015a). Clustering does not have an access to class labels and its results depend only on the values of features describing each object (Collingwood and Lohwater 2004; Xu and Wunsch 2009). In real applications, groups that we want to extract can be too complex to be discovered by strictly unsupervised algorithms (Bar-Hillel et al. 2003). Therefore, in order to support analysis or visualization of data, the user often provides additional information to indicate the crucial values, parts of a graph, etc. Consequently, clustering process is supposed to take an advantage of such background knowledge to provide better results.

Constrained clustering is a part of semi-supervised learning (Basu et al. 2002). It incorporates equivalence constraints between some pairs of elements to enforce which of them belong to the same group (positive constraints or must-link constraints) and which do not (negative constraints or cannot-link constraints) (Wagstaff et al. 2001). It has been widely used in various real-world applications like GPS-based map refinement (Wagstaff et al. 2001) or landscape detection from hyperspectral data (Lu and Leen 2004). On the other hand, in semi-supervised classification, learning involves the use of only a small amount of labeled data together with a large amount of unlabeled elements (Bennett and Demiriz 1998). Basically, the pairwise constraints used in clustering cannot be directly transformed into class labels, and it makes a conceptual difference between semi-supervised clustering and classification.

Numerous clustering algorithms have been modified to aggregate additional information from equivalence constraints. Most of adopted methods, including k-means (Wagstaff et al. 2001), Gaussian mixture model (Shental et al. 2004; Melnykov et al. 2015), hierarchical algorithms (Klein et al. 2002), spectral methods (Li et al. 2009), generate partitions, which are fully consistent with imposed restrictions (they define hard-type of constraints). It is worth to mention that there are also methods in which the constraints come in the form of suggestions that can occasionally be violated (Bilenko et al. 2004; Lu and Leen 2004; Wang and Davidson 2010), however in this paper we do not follow such an approach.

A version of the Gaussian mixture model (GMM) proposed by Shental et al. (2004) is one of the most interesting clustering approaches which impose hard restrictions. Equivalence constraints are used to gather points into chunklets, i.e. sets of elements that are required to be included into the same clusters. Chunklets may be obtained by applying the transitive closure to the set of positive constraints, which generate equivalence classes. The algorithm fits a mixture of Gaussians to unlabeled data together with constructed chunklets by summing over assignments which comply with constraints. Unfortunately, this method does not handle well a situation, presented in Fig. 1. The equivalence constraints, Fig. 1b, enforce that “ears elements” of mouse-like set belong to the same group. However, a direct application of constrained GMM assigns to them one Gaussian model which ultimately groups “ears elements” together with some of “head elements”, Fig. 1c.



**Fig. 1** Clustering of mouse-like set (a). The equivalence constraints (positive and negative) specified on 10 % of the data elements determine that “ears elements” are included in one group and “head elements” in the second (b). In contrast to the constrained version of GMM (c), our algorithm has discovered an expected partition of this dataset (d)

In this paper we propose a general constrained clustering algorithm, called Constrained clustering with a complex cluster structure (C4s), which incorporates equivalence constraints, both positive and negative, and deals well with the aforementioned problem, Fig. 1d. The idea of C4s relies on the observation that every chunklet can originate from complex model, e.g. a mixture of probability distributions. To find a detailed description of each chunklet, we cluster their elements individually at the initial stage of the algorithm, which yields small groups of data points. The obtained groups are used as the atomic parts of data for final clustering process. To ensure that none of negative constraints is violated during the clustering process, we formulate two strict conditions, see Theorems 2 and 3.

Various clustering algorithms can be applied to implement the C4s approach. However, since the number of components for each chunklet is not specified a priori, it is preferable to use an algorithm which detects the number of clusters automatically. Therefore, we combine our method with cross-entropy clustering (CEC) (Tabor and Spurek 2014; Spurek et al. 2013; Tabor and Misztal 2013; Śmieja and Tabor 2013, 2015b) which can be seen as a model-based clustering (McLachlan and Krishnan 2008;

Morlini 2012; Subedi and McNicholas 2014; Baudry et al. 2015) and determines the final number of groups.

In order to evaluate the performance of C4S, we applied it to a semi-supervised image segmentation and compared it with competitive constrained methods on several datasets. In particular, we used examples retrieved from UCI repository (Lichman 2013) as well as a real dataset including chemical compounds acting on central nervous system (Warszycki et al. 2013). Our experiments show that C4S gives comparable results to other constrained methods when the chunklets are represented by simple models. In the case of more complicated structure of restrictions, C4S significantly outperforms the competitive techniques.

The paper is organized as follows. The next section introduces a general C4S procedure (a combinatorial approach to constrained clustering with a complex cluster structure). In the third section, we show how to implement C4S algorithm with a use of CEC approach. The experimental results are presented in fourth section. Finally, a conclusion is given.

## 2 General C4S algorithm

In this section, we present a general form of proposed clustering procedure with equivalence constraints. We do not specify the clustering criterion here, e.g. cost function, dissimilarity measure, etc., but focus on defining generic steps which can be accomplished using any clustering method. In other words, we consider here the clustering problem as a combinatorial one, in which the goal is to construct clusters which comply with imposed restrictions. A specific implementation will be discussed in the next section.

### 2.1 Problem formulation

Let  $X = \{1, \dots, n\}$  be an  $n$ -element dataset of objects augmented by equivalence constraints between some pairs of its elements. Positive constraint enforces that two elements belong to the same group while negative constraint states that they are classified separately. The input restrictions are given in the form of a set of pairs  $(x, y) \in X \times X$ , where the following notation is used:

- $x \sim y$  denotes that a positive constraint is imposed on  $x$  and  $y$  (they have to be included into the same class),
- $x \approx y$  denotes that a negative constraint is imposed on  $x$  and  $y$  (they have to be included into diverse classes).

Let us observe that the set of positive (or negative) constraints determines a relation on  $X \times X$ . For this reason, we sometimes say that two elements are in positive (or negative) relation. In particular, the set of positive constraints defines an equivalence relation. In consequence, the input set of restrictions generates the equivalence classes (called chunklets by Shental et al. 2004). An equivalence class contains such elements of dataset, which are in positive relation (directly or transitively) and finally must be

grouped together. On the other hand, negative relation is not transitive, which makes it usually harder to incorporate them into clustering algorithms.

We say that  $X_1, \dots, X_k$ , for  $k \in \mathbb{N}$ , is a *partition* of  $X$  if it is a family of pairwise disjoint subsets of  $X$  such that  $X = \bigcup_{i=1}^k X_i$ . Our goal is to construct such a partition of  $X$ , which complies with assumed constraints (the cardinality of partition is not specified). We assume that there exists at least one partition of  $X$ , which does not violate any constraint. This is the case when there is no pair of elements that belong to the same equivalence class, but have a negative relation.

In this paper we are interested in defining a flexible framework, which allows for finding the most appropriate clusters models for particular dataset and constraints. To illustrate this goal, we recall an example of the mouse-like set presented in Fig. 1. The application of standard unconstrained GMM results in detecting three spherically shaped clusters. However, after the specification of the equivalence relation the clustering method is supposed to discover such clustering configuration, which comply with the imposed restrictions and fits best to a dataset. In the case of model-based clustering we want to be able to automatically select two Gaussian components for describing the “ears of mouse” and one Gaussian for its “head”, see Fig. 1d.

We start with a description of the clustering process in the case of positive constraints only (Sect. 2.2). The entire procedure can be divided into four main steps:

1. *Aggregation of positive constraints* The elements with positive constraints are collected into equivalence classes, termed initial chunklets.
2. *Inner clustering* Every initial chunklet is clustered into smaller parts, called final chunklets.
3. *Global clustering* The set of final chunklets is partitioned into clusters containing similar final chunklets.
4. *Merging* Clusters (obtained in previous step) that contain elements in positive relation, are merged together.

The incorporation of negative constraints, requires a modification of a global clustering step, in which we have to create a grouping that is consistent with all constraints after the merging operation. We formulate two conditions, which allow to verify if a given partition constructed in global clustering can be merged into a partition which does not violate any constraint, see Theorems 2 and 3.

## 2.2 Incorporating positive constraints

In this section, we assume that a dataset  $X$  is augmented by positive constraints only. Proposed procedure relies on four steps, outlined in previous subsection and explained in the following paragraphs.

*Aggregation of positive constraints.* In the first step, we follow the arguments introduced by Shental et al. (2004) and consider the equivalence classes generated by the set of positive constraints. More precisely, for  $x \in X$ , we construct a set:

$$[x] = \{y \in X : x \sim y\}.$$

One element equivalence classes are created for the elements which were not concerned in any positive constraint.

Since for every  $x, y \in X$ , it holds that  $[x] = [y]$  or  $[x] \cap [y] = \emptyset$ , we have

$$X = C_1 \cup \dots \cup C_k, \text{ for } k \in \mathbb{N},$$

where  $(C_i)_{i=1}^k$  is a family of all pairwise disjoint equivalence classes in  $X$ . For a further use, each equivalence class  $C_i$  will hereafter be referred as *initial chunklet*.

*Inner clustering* Obviously, all elements associated with an initial chunklet should be finally included into the same cluster. However, every group can have very complex structure, e.g. its elements might not be generated from a single Gaussian probability distribution, but from a mixture of Gaussian distributions, as presented in Fig. 1. Roughly speaking, inner clustering attempts to discover atomic groups in  $X$  that are described by simple models.

For this reason, we want to construct a separate partition of every initial chunklet. As mentioned, we do not specify a type of clustering method here, but assume that its choice is an independent task, which will be discussed in the next section. As a result of inner clustering, every initial chunklet  $C_i$ , for  $i = 1, \dots, k$ , is split into

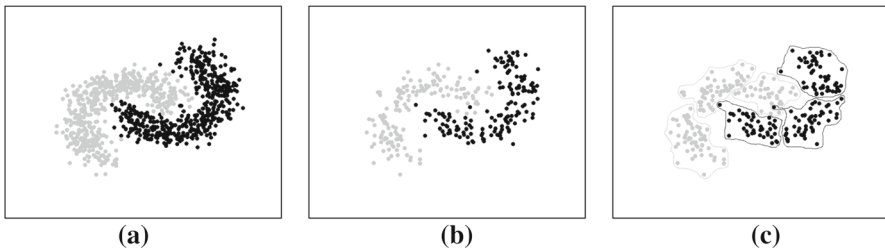
$$C_i = C_1^{(i)} \cup \dots \cup C_{k_i}^{(i)}, \text{ for specific } k_i \in \mathbb{N}.$$

We assume that the number of groups  $k_i$  created for  $i$ -th initial chunklet is not greater than its cardinality. In particular, for a set with only one element we obtain a trivial partition containing just one singleton class.

Retrieved groups  $C_j^{(i)}$  are referred as *final chunklets*. Moreover, we say that a *final chunklet*  $C_j^{(i)}$  is derived from an *initial chunklet*  $C_i$  if  $C_j^{(i)} \subset C_i$ . Observe that the set of final chunklets

$$\mathcal{C} = \{C_j^{(i)} : i = 1, \dots, k, j = 1, \dots, k_i\}$$

constitutes a partition of  $X$ .



**Fig. 2** Presentation of the chunklets' construction for the Banana-like set. The expected grouping of Banana-like set, (a). After imposing the positive and negative constraints on 30 % of its elements two initial chunklets are created, (b). The cross-entropy implementation of the introduced algorithm generates six final chunklets, (c)

In order to illustrate the inner clustering process, the cross-entropy implementation of C4s (presented in the next section) was applied on Banana-like set, Fig. 2. This is an example of data arranged around two parabolic manifolds. This kind of sets is becoming increasingly popular due to the manifold hypothesis which states that real world data embedded in high dimensional spaces are likely to concentrate in the vicinity of nonlinear sub-manifolds of lower dimensionality, see Cayton (2005); Narayanan and Mitter (2010). The clustering grants detection of manifolds. Interestingly, diverse shapes, such as Banana-like, often appear in medical sciences, e.g. in muscle injury determination system (Ding et al. 2011). In order to determine muscle injury from ultrasonic image of a healthy and unhealthy muscle, a specific shape of fiber has to be detected.

*Global clustering* Let  $\mathcal{C}$  be a family of all final chunklets discovered in the inner clustering stage. As a global clustering, we understand the process of constructing a partition  $\mathcal{P}$  of  $\mathcal{C}$ . In consequence, each cluster  $P \in \mathcal{P}$  will be a family of some final chunklets. If every final chunklet is described by one simple model, then we group similar chunklets together and in consequence, obtain clusters, which could follow the mixture models.

Let us observe that  $\mathcal{C}$  is a family of subsets of  $X$ . Therefore, the clustering algorithm has to be adapted to process a dataset that consists of some subsets of  $X$ . To facilitate such a procedure one can represent every final chunklet as a single element of  $X$  and apply a clustering algorithm in a classical way. The idea can be best illustrated with an example: in the case of k-means one can represent a final chunklet by its mean with a weight depending on the cardinality of chunklet. On the other hand, in the Gaussian mixture model approach every set is naturally related to a probability model characterized by its sample mean and covariance matrix. These adaptations allow to apply weighted versions of classical clustering algorithms, without referring to  $\mathcal{C}$  but its transformed form of representants. For more details, we refer the reader to the next section, in which CEC implementation is presented after such an adaptation.

*Merging* Let  $\mathcal{P}$  be a partition of  $\mathcal{C}$  created by a global clustering procedure, i.e., each  $P \in \mathcal{P}$  is a family of some final chunklets. This splitting might be inconsistent with some of positive constraints, e.g. it is possible to assign  $C_{i_1}^{(i)}$  to  $P$  and  $C_{i_2}^{(i)}$  to  $P'$ , for  $P, P' \in \mathcal{P}$  such that  $P \neq P'$ , whereas  $C_{i_1}^{(i)}$  and  $C_{i_2}^{(i)}$  are derived from the same initial chunklet  $C_i$ . In merging stage, we join two groups  $P, P' \in \mathcal{P}$  if they contain final chunklets, which are derived from the same initial chunklet.

For every  $P \in \mathcal{P}$ , we want to encode the information of clusters, which must be joined with  $P$  to ensure that none of positive constraints is violated. Let  $cl : \mathcal{P} \rightarrow 2^{\mathcal{P}}$  be a function such that:

- $P' \in cl(P)$ , for  $P' \in \mathcal{P}$ , if there exist two final chunklets  $C_{i_1}^{(i)} \in P$  and  $C_{i_2}^{(i)} \in P'$  that are derived from initial chunklet  $C_i$ .

A value  $cl(P)$  is a set of clusters  $P'$ , such that  $P, P'$  both contain final chunklets  $C_{i_1}^{(i)}$  and  $C_{i_2}^{(i)}$ , respectively, derived from the same  $C_i$ . In other words,  $P'$  is connected directly with  $P$  by some initial chunklet.

Let us observe that, if  $P' \in cl(P)$  and  $P'' \in cl(P')$ , for  $P, P', P'' \in \mathcal{P}$ , then  $P, P', P''$  have to be finally included into the same cluster, even if  $P'' \notin cl(P)$ . To

encode such a transitive relation a sequence of functions  $(cl^{(t)})_{t \in \mathbb{N}}$  is defined recursively by:

$$cl^{(t)}(P) := \begin{cases} cl(P) & \text{for } t = 1, \\ \bigcup \{cl(P') : P' \in cl^{(t-1)}(P)\} & \text{for } t > 1, \end{cases}$$

for  $P \in \mathcal{P}$ . We put

$$cl^\infty(P) := \bigcup_{t=1}^\infty cl^{(t)}(P), \text{ for } P \in \mathcal{P}$$

to denote all clusters that have to be joined together in the merge stage, see Fig. 3 for the illustration. One can interpret  $cl^\infty(P)$  as a closure of  $\{P\}$  with respect to the positive relation.

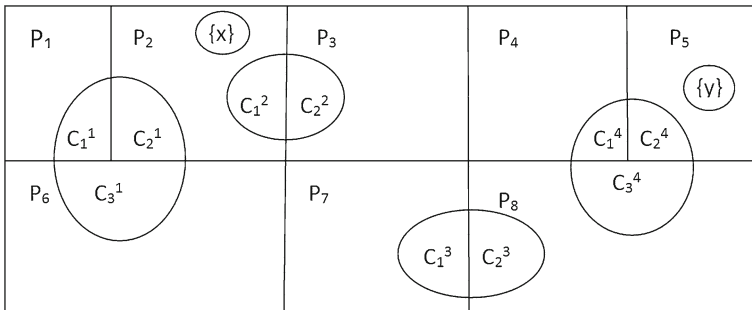
The above construction of  $cl^\infty$  provides that

$$\text{either: } (cl^\infty(P) = cl^\infty(P')) \text{ or } (cl^\infty(P) \cap cl^\infty(P') = \emptyset), \text{ for } P, P' \in \mathcal{P}.$$

In other words,  $cl^\infty$  generates equivalence classes on  $\mathcal{P}$ . If we denote by  $\mathcal{Q}$  the family of all different equivalence classes generated by  $cl^\infty$ , then  $\mathcal{Q}$  is a partition of  $\mathcal{C}$ . To obtain a partition  $\mathcal{X} = \{X_Q\}_{Q \in \mathcal{Q}}$  of  $X$ , which corresponds to  $\mathcal{Q}$ , we transform every  $Q \in \mathcal{Q}$  by:

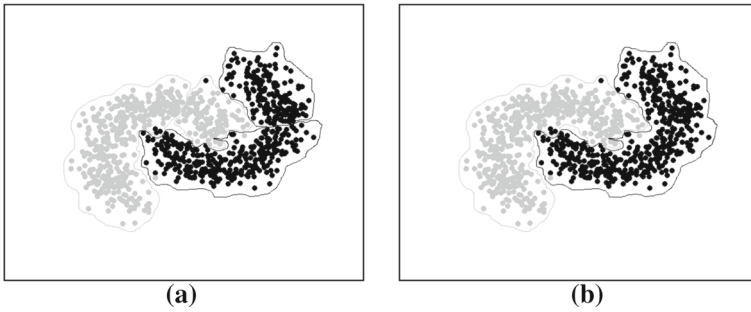
$$X_Q := \bigcup_{C \in Q} C.$$

We see that  $\mathcal{X}$  is consistent with all positive constraints, as outlined in the following theorem:



**Fig. 3** Partition of dataset into 8 clusters and final chunklets derived from 4 initial chunklets  $C_1, C_2, C_3, C_4, \{x\}, \{y\}$ . Since  $cl^\infty(P_1) = \{P_1, P_2, P_3, P_6\}$  and  $cl^\infty(P_8) = \{P_4, P_5, P_7, P_8\}$  then the merge operation will result in constructing 2 clusters. On the other hand,  $cl^\infty(P_1 \cup \{C_3^4\}) = \{P_1, P_2, P_3, P_4, P_5, P_6\}$  and  $\partial(C_3^4) = \{P_4, P_5\}$ . Therefore, we have  $\partial(C_3^4) = \{P_1, P_2, P_3, P_6\}$ . If  $x \approx y$  then the reassignment of  $C_3^4$  from  $P_8$  to  $P_1$  violates this constraints, see Theorems 2 and 3





**Fig. 4** Resulting groups of the global clustering (a) and merge process of Banana-like set (b) (constraints and chunklets are shown in Fig. 2)

**Theorem 1** *Let  $\mathcal{P}$  be a partition of the family of final chunklets  $\mathcal{C}$  and let  $\mathcal{Q}$  be a family of all equivalence classes generated by  $\text{cl}^\infty(P)$ , for  $P \in \mathcal{P}$ . Then  $\mathcal{X} = \{X_Q\}_{Q \in \mathcal{Q}}$  defined by:*

$$X_Q := \bigcup_{C \in Q} C$$

*is a partition of  $X$  which coincides with all positive constraints.*

Figure 4 demonstrates the results of global clustering and merging for the Banana-like set.

### 2.3 Incorporating negative constraints

In this section we assume that both positive and negative constraints are defined on selected pairs of  $X$ . The procedure proposed in previous subsection does not use the information contained in negative constraints. To apply this algorithm in the case of negative constraints one has to modify a global clustering step. For this purpose, we formulate two conditions, which allow to verify if a given partition obtained in the global clustering stage can be merged to a partition which is consistent with the negative constraints.

For a further convenience, we say that *two initial chunklets  $C_i, C_j \subset X$  are in negative relation*, which we denote by  $C_i \approx C_j$ , if there exist  $x \in C_i, y \in C_j$  such that  $x \approx y$ . In other words, since all the elements of any initial chunklet have to be finally included into a single cluster (after the merge stage), then the negative constraints can be propagated and verified on the set of initial chunklets. Moreover, we say that *a partition  $\mathcal{P}$  of  $\mathcal{C}$  is valid* if in the merging stage it generates a partition of  $X$  which is consistent with all negative constraints.

The following result shows how to verify the validity of a partition  $\mathcal{P}$  of  $\mathcal{C}$  based on the equivalence classes generated by  $\text{cl}^\infty$ :

**Theorem 2** *A partition  $\mathcal{P}$  of  $\mathcal{C}$  is not valid if and only if there exist  $P \in \mathcal{P}$  and final chunklets  $C_{i_1}^{(i)}, C_{j_1}^{(j)} \in \bigcup\{P' : P' \in \text{cl}^\infty(P)\}$  derived from initial chunklets  $C_i, C_j$ , respectively, such that  $C_i \approx C_j$ .*

*Proof* Let us first assume that a partition  $\mathcal{P}$  is not valid, i.e. a merge operation generates a partition  $\mathcal{X}$  of  $X$  which is not consistent with at least one constraint. Therefore, there exist a cluster  $Y \in \mathcal{X}$  and  $x, y \in Y$  such that  $x \approx y$ . One can find two final chunklets  $C_{i_1}^{(i)}, C_{j_1}^{(j)} \subset Y$  derived from initial chunklets  $C_i, C_j$ , respectively, such that  $x \in C_{i_1}^{(i)}, y \in C_{j_1}^{(j)}$  and  $C_i \approx C_j$ . Since both  $x$  and  $y$  belong to  $Y$  then there exists  $P \in \mathcal{P}$  such that  $C_{i_1}^{(i)}, C_{j_1}^{(j)} \in \bigcup\{P' : P' \in \text{cl}^\infty(P)\}$ .

On the other hand, let us assume that there exist  $P \in \mathcal{P}$  and final chunklets  $C_{i_1}^{(i)}, C_{j_1}^{(j)} \in \bigcup\{P' : P' \in \text{cl}^\infty(P)\}$  derived from  $C_i, C_j$ , respectively, such that  $C_i \approx C_j$ . Since  $C_{i_1}^{(i)}, C_{j_1}^{(j)} \in \bigcup\{P' : P' \in \text{cl}^\infty(P)\}$  then  $C_i \cup C_j$  will be joined together in the merge stage—it does not lead to a valid partition because  $C_i \approx C_j$ .  $\square$

In many clustering algorithms such as k-means, we start with a fixed partition and focus on its iterative refinement by switching the elements between clusters. Clearly, one could use Theorem 2 to verify if a given reassignment leads to a valid partition. Nevertheless, this operation might be computationally inefficient for this type of algorithms. Therefore, we formulate a condition that states when we are permitted to change the membership of a final chunklet from one cluster to another to preserve the validity of a partition.

Let  $\mathcal{P}$  be a fixed partition of  $\mathcal{C}$  and let  $C_j^{(i)} \in P'$ , for  $P' \in \mathcal{P}$ , be a final chunklet derived from an initial chunklet  $C_i$ . If we change the membership of  $C_j^{(i)}$  from  $P'$  to a cluster  $P \in \mathcal{P}$ , then  $\text{cl}^\infty(P)$  will change if only  $\text{cl}^\infty(P') \cap \text{cl}^\infty(P) = \emptyset$ . If  $\text{cl}^\infty(P) = \text{cl}^\infty(P')$  then such a reassignment has no effect on  $\text{cl}^\infty(P)$ . Therefore, at each attempt of reassigning  $C_j^{(i)}$  from  $P'$  to  $P$ , we have to verify if there is any pair of clusters held in  $(\text{cl}^\infty(P \cup \{C_j^{(i)}\}) \setminus \text{cl}^\infty(P)) \times \text{cl}^\infty(P)$  which breaks any negative constraint, i.e. contain final chunklets derived from initial chunklets which are in negative relation.

The following lemma will be useful to establish the form of  $\text{cl}^\infty(P \cup \{C_j^{(i)}\}) \setminus \text{cl}^\infty(P)$

**Lemma 1** *Let  $\mathcal{P}$  be a partition of a family of final chunklets  $\mathcal{C}$ . We consider  $P, P' \in \mathcal{P}$  and a final chunklet  $C_j^{(i)} \in P'$  derived from  $C_i$ . If  $\text{cl}^\infty(P') \cap \text{cl}^\infty(P) = \emptyset$  then*

$$\text{cl}^\infty(P \cup \{C_j^{(i)}\}) \setminus \text{cl}^\infty(P) = \bigcup\{\text{cl}^\infty(P'') : P'' \in \mathcal{P}, C_l^{(i)} \in P'' \text{ and } C_l^{(i)} \text{ is a final chunklet derived from } C_i, \text{ where } l \neq j\}.$$

otherwise  $\text{cl}^\infty(P \cup \{C_j^{(i)}\}) \setminus \text{cl}^\infty(P) = \emptyset$ .

*Proof* If  $\text{cl}^\infty(P') \cap \text{cl}^\infty(P) = \emptyset$  then we consider all final chunklets  $C_l^{(i)}$  which are derived from  $C_i$  such that  $C_l^{(i)} \neq C_j^{(i)}$ . If  $C_l^{(i)} \in P''$  then all clusters from  $\text{cl}^\infty(P'')$  belong to  $\text{cl}^\infty(P \cup \{C_j^{(i)}\})$  because both  $C_j^{(i)}, C_l^{(i)} \subset C_i$ . Moreover,

$cl^\infty(P'') \cap cl^\infty(P) = \emptyset$  since  $cl^\infty(P') \cap cl^\infty(P) = \emptyset$ . This proves the first part of theorem.

If  $cl^\infty(P') = cl^\infty(P)$  then the reassignment of  $C_j^{(i)}$  has no effect on  $cl^\infty(P)$  and in consequence  $cl^\infty(P \cup \{C_j^{(i)}\}) \setminus cl^\infty(P) = \emptyset$ . □

We put:

$$\partial(C_j^{(i)}) := \bigcup \{cl^\infty(P'') : P'' \in \mathcal{P}, C_l^{(i)} \in P'' \text{ and } C_l^{(i)} \text{ is a final chunklet derived from } C_l, \text{ where } l \neq j\},$$

which can be considered as a boundary of  $C_j^{(i)}$ . An illustrative explanation of the above definitions is presented in Fig. 3.

The following result allows to check out if a given reassignment operation preserves the validity of partition:

**Theorem 3** *Let  $\mathcal{P}$  be a valid partition of  $\mathcal{C}$ . We assume that  $P, P' \in \mathcal{P}$  are fixed and we consider a final chunklet  $C_{i_1}^{(i)} \in P'$ . Let  $\mathcal{Q}$  be a partition generated from  $\mathcal{P}$  by changing the membership of  $C_{i_1}^{(i)}$  from  $P'$  to  $P$ , i.e.*

$$\mathcal{Q} = \{P \cup \{C_{i_1}^{(i)}\}, P' \setminus \{C_{i_1}^{(i)}\}\} \cup \bigcup \{P'' \in \mathcal{P} : P'' \neq P, P'' \neq P'\}.$$

Partition  $\mathcal{Q}$  is valid if one the following conditions is satisfied:

1.  $cl^\infty(P) = cl^\infty(P')$
2. for all pairs of final chunklets  $(C_{j_1}^{(j)}, C_{l_1}^{(l)})$  such that  $C_{j_1}^{(j)} \in \bigcup \{P'' : P'' \in cl^\infty(P)\}$  and  $C_{l_1}^{(l)} \in \bigcup \{P'' : P'' \in \partial(C_{j_1}^{(i)})\}$ , the initial chunklet  $C_j$  is not in negative relation with  $C_l$ , where  $C_{j_1}^{(j)}$  and  $C_{l_1}^{(l)}$  are derived from initial chunklets  $C_j, C_l$ , respectively.

*Proof* Clearly, if  $cl^\infty(P) = cl^\infty(P')$  then  $\mathcal{Q}$  is valid because  $\mathcal{P}$  is valid.

Let us suppose indirectly that condition 3 holds and partition  $\mathcal{Q}$  is not valid. Since  $\mathcal{P}$  is valid then for all  $P \in \mathcal{P}$  neither  $\bigcup \{P'' : P'' \in cl^\infty(P)\}$  nor  $\bigcup \{P'' : P'' \in cl^\infty(P')\}$  contain final chunklets which were derived from initial chunklets that are in negative relation. Therefore, there exist  $C_{j_1}^{(j)} \in \bigcup \{P'' : P'' \in cl^\infty(P)\}$  and  $C_{l_1}^{(l)} \in \bigcup \{P'' : P'' \in \partial(C_{i_1}^{(i)})\}$  derived from  $C_j, C_l$ , respectively, such that  $C_j \approx C_l$ . However, it is a contradiction to condition 3. □

If we perform a global clustering stage employing a clustering algorithm that iteratively switches elements of dataset between clusters, then Theorem 3 determines all acceptable reassignments. We start with any valid partition. At each reassigning step we verify if it leads to a valid partition and only then consider a possible change of membership. One can use the following pseudocode to perform the reassigning operation.

```

1: INPUT:
2:  $\mathcal{P}$ : partition of a family of all final chunklets  $\mathcal{C}$ 
3:  $P, P' \in \mathcal{P}$ : clusters
4:  $C \in P'$ : final chunklet
5: function reassign( $C, P, P'$ )
6: if  $\text{cl}^\infty(P) = \text{cl}^\infty(P')$  then
7:    $P = P \cup \{C\}$ 
8:    $P' = P' \setminus \{C\}$ 
9: else
10:   canBeChanged = TRUE
11:   for all  $Q \in \text{cl}^\infty(P)$  do
12:     for all  $C_{j_1}^{(j)} \in Q$  do
13:       for all  $Q' \in \partial(C)$  do
14:         for all  $C_{l_1}^{(l)} \in Q'$  do
15:           if  $C_j \approx C_l$  then
16:             canBeChanged = FALSE
17:             go to 2.3 line
18:           end if
19:         end for
20:       end for
21:     end for
22:   if canBeChanged then
23:      $P = P \cup \{C\}$ 
24:      $P' = P' \setminus \{C\}$ 
25:   end if
26: end if

```

To reduce the complexity of the above algorithm it is enough to collect in each cluster  $P \in \mathcal{P}$  the family

$$L(P) = \{C_{i_1}^{(i)} \in P : C_i \text{ has any negative constraint}\}$$

of all final chunklets which are derived from initial chunklets having any negative constraint. The iterations given in lines 2.3 and 2.3 are then performed only through final chunklets from  $L(P)$ . In consequence the reassignment of  $C \in P'$  to  $P$  takes  $(\sum_{P'' \in \text{cl}^\infty(P)} |L(P'')|) \cdot (\sum_{P'' \in \partial(C)} |L(P'')|)$  operations. Since the cardinalities of  $\text{cl}^\infty(P)$  and  $\partial(C)$  depend on the number of positive constraints while the cardinality of  $L(\cdot)$  is proportional the number of negative constraints, then one may say that the cost of verification the reassignment operation can be approximated by the total number of negative and positive constraints.

### 3 Implementation with use of model-based clustering and cross-entropy

In this section we present an implementation of proposed C4S procedure that employs cross-entropy clustering method (CEC). This is an a technique based on information

theoretical concepts, which has similar effects as classical model-based clustering. Moreover, it automatically detects the final number of groups, which is extremely important in our procedure due to the presence of inner and global clustering phases.

In this section we assume that  $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^N$  is a dataset of  $N$ -dimensional real-valued vectors. We start by presenting CEC method and its comparison with classical model-based clustering. Then, we show how to implement the C4s procedure with the help of CEC.

### 3.1 Mixture of Gaussian models

The idea of model-based clustering comes from Wolfe (1963) and has become increasingly popular across diverse applications (Bellas et al. 2013; McNicholas and Murphy 2010; Xiong et al. 2002; Samuelsson 2004). Although a variety of finite mixture models has been extensively studied and developed in the literature (Baudry et al. 2015; Subedi and McNicholas 2014; Lee and McLachlan 2013; Morris et al. 2013), the Gaussian case has received a special attention (Morlini 2012; Nguyen and McLachlan 2015; Hennig 2010; Scrucca and Raftery 2015).

Basically, model-based clustering focuses on a density estimation of a dataset  $X$  by the mixture of simple densities. It aims to find

$$p_1, \dots, p_k \geq 0 : \sum_{i=1}^k p_i = 1, \tag{1}$$

and  $f_1, \dots, f_k \in \mathcal{F}$ , where  $k \in \mathbb{N}$  is fixed and  $\mathcal{F}$  is a parametric (usually Gaussian) family of densities such that the convex combination

$$f = p_1 f_1 + \dots + p_k f_k$$

estimates unknown probability distribution on a dataset  $X$  (McLachlan and Krishnan 2008). This is a fuzzy-type clustering, where the probability of assigning  $x \in X$  to  $i$ -th clusters equals  $p_i f_i(x)$ . A locally optimal solution, which minimizes a negative log-likelihood function:

$$\text{EM}(f, X) = -\frac{1}{|X|} \sum_{j=1}^n \log (p_1 f_1(x_j) + \dots + p_k f_k(x_j)), \tag{2}$$

where  $|X| = n$  is a cardinality of  $X$ , can be found by applying the EM algorithm.

The goal of CEC is similar: it aims at finding numbers  $p_1, \dots, p_k$  that satisfy (1) and densities  $f_1, \dots, f_k \in \mathcal{F}$ , which minimize the following cost function:

$$\text{CEC}(f, X) = -\frac{1}{|X|} \sum_{j=1}^n \log (\max(p_1 f_1(x_j), \dots, p_k f_k(x_j))). \tag{3}$$

If  $\mathcal{F}$  is a family of Gaussian densities, then  $f = \max(p_1 f_1, \dots, p_k f_k)$  is not a density, but a subdensity, i.e.  $\int_{\mathbb{R}^N} f(x) dx \leq 1$ .

The formula (3) is known as the cross-entropy (Rubinstein and Kroese 2004) of dataset  $X$  with respect to  $f$ . If we define a partition  $X_1, \dots, X_k$  of  $X$  by

$$X_i := \{x \in X : p_i f_i(x) > \max_{j \neq i} (p_j f_j(x))\}$$

then (3) can be rewritten as:

$$\text{CEC}(f, X) = \sum_{i=1}^k \frac{|X_i|}{|X|} \cdot \left( -\log p_i - \frac{1}{|X_i|} \sum_{x \in X_i} \log f_i(x) \right).$$

One can understand the CEC formula as a mean cost of encoding a symbol from a dataset  $X$  by a model consisting of  $k$  encoders: the term  $-\log p_i$  is a code-length of encoder identifier while  $-\log f_i(x)$  determines a code-length of  $x$  when using  $i$ -th coding algorithm. An immediate consequence of the above formula is that the clusters do not “cooperate” one with each another to estimate a density of  $X$  (it is a hard-type of clustering) and as a result it is enough to define an optimal description for each cluster separately. Let us observe that, each cluster has set its individual cost given by  $-\log p_i$ , which allows to regularize a clustering model. While the introduction of one more group usually improves the likelihood function, it also increases the complexity of the model. This is the reason why CEC tends to reduce unnecessary clusters.

We assume that  $\mathcal{F}$  is a family of Gaussian densities  $\mathcal{N}(m, \Sigma)$ . Let  $X_1, \dots, X_k$  be a fixed partition of  $X$ . By  $\hat{m}_i$  and  $\hat{\Sigma}_i$  we denote the sample mean and covariance matrix calculated within a group  $X_i$  as:

$$\begin{aligned} \hat{m}_i &= \frac{1}{|X_i|} \sum_{x \in X_i} x, \\ \hat{\Sigma}_i &= \frac{1}{|X_i|} \sum_{x \in X_i} (x - \hat{m}_i)(x - \hat{m}_i)^T. \end{aligned}$$

The infimum of CEC cost function (3) for a partition  $X_1, \dots, X_k$  taken over all acceptable  $p_i$  and  $f_i$ , for  $i = 1, \dots, k$ , equals:

$$\sum_{i=1}^k p_i \cdot \left[ \frac{N}{2} \ln(2\pi \varepsilon) - \ln(p_i) + \frac{1}{2} \ln \det(\hat{\Sigma}_i) \right], \tag{4}$$

where  $p_i = \frac{|X_i|}{|X|}$ .

To find a partition, which minimizes (4), one can adopt an iterative Hartigan algorithm, which is commonly used in the case of k-means (Jain 2010). The idea of the Hartigan method is to proceed over all elements of  $X$ , switching the membership of particular elements to those clusters which would maximally decrease the cost function (Telgarsky and Vattani 2010; Hartigan and Wong 1979). It can be proven that this algorithm refines a given partition and finally finds a locally optimal solution.

The entire procedure can be summarized in the following steps:

1. Let  $X_1, \dots, X_k$  be an initial partition of  $X$ . In the simplest case it can be a random grouping.

2. Iterate over all  $x \in X$  and execute the following steps until no cluster membership has been changed:
  - (a) Find a membership of  $x \in X_i$  to this cluster  $X_j$  for which the decrease of the cost function (4) is maximal. To evaluate the change of the cost function after the reassignment from  $X_i$  to  $X_j$ , we have to temporally recalculate the probabilities, means and covariances of these clusters.
  - (b) If an optimal cluster membership  $X_j \neq X_i$ , then switch  $x$  from  $X_i$  to  $X_j$  and update the parameters of these clusters permanently. Otherwise, no reassignment is performed.
  - (c) Reduce a cluster  $X_i$ , if  $p_i < \varepsilon$  (for most application  $\varepsilon \leq 2\%$  provides satisfactory results) and assign its elements to different clusters according to point (a);

Though it may seem that the recalculation of the cluster model in the above procedure involves high computational complexity, it does not. The following formulas show that the time of these updates does not depend on the cardinality of data, but only on the dimension of dataset. Every cluster has only to remember its actual sample mean and covariance.

**Observation 1** (Tabor and Spurek 2014, Theorem 4.3) *Let  $U, V$  be two subsets of  $X \subset \mathbb{R}^N$  with sample means  $\hat{m}_U, \hat{m}_V$ , covariance matrices  $\hat{\Sigma}_U, \hat{\Sigma}_V$  and associated prior probabilities  $p(U), p(V) \geq 0$  such that  $p(U) + p(V) \leq 1$  [the role of  $p(U), p(V)$  is the same as numbers  $p_i$  in (3)].*

- *If we assume that  $U \cap V = \emptyset$  then the sample mean and the covariance of  $U \cup V$  equals:*

$$\begin{aligned}
 - \hat{m}_{U \cup V} &= p_U \hat{m}_U + p_V \hat{m}_V, \\
 - \hat{\Sigma}_{U \cup V} &= p_U \hat{\Sigma}_U + p_V \hat{\Sigma}_V + p_U p_V (\hat{m}_U - \hat{m}_V)(\hat{m}_U - \hat{m}_V)^T,
 \end{aligned}$$
*where  $p_U = \frac{p(U)}{p(U)+p(V)}, p_V = \frac{p(V)}{p(U)+p(V)}$ .*
- *If we assume that  $V \subset U$  then the sample mean and covariance of  $U \setminus V$  equals:*

$$\begin{aligned}
 - \hat{m}_{U \setminus V} &= q_U \hat{m}_U - q_V \hat{m}_V, \\
 - \hat{\Sigma}_{U \setminus V} &= q_U \hat{\Sigma}_U - q_V \hat{\Sigma}_V - q_U q_V (\hat{m}_U - \hat{m}_V)(\hat{m}_U - \hat{m}_V)^T,
 \end{aligned}$$
*where  $q_U = \frac{p(U)}{p(U)-p(V)}, p_V = \frac{p(V)}{p(U)-p(V)}$ .*

### 3.2 Cross-entropy C4s

Let  $X \subset \mathbb{R}^N$  be a dataset augmented by the set of positive and negative equivalence constraints. We discuss the application of CEC in C4s algorithm, in particular in realizing the inner and the global clustering stages. We consider a Gaussian version of CEC, i.e. every cluster is modeled as Gaussian probability distribution. For a convenience, we use a notation and a terminology introduced in Sect. 2.

*Inner clustering* In the inner clustering we extract atomic parts of every initial chunklet. In the case of CEC, we try to discover final chunklets represented by Gaussian distributions. To run CEC, the maximum number of groups  $gr_{\max}^{(i)}$ , for  $i = 1, \dots, k$ , must be specified for each initial chunklet. Since the constraints usually cover a small number of examples then  $gr_{\max}^{(i)}$  should also be small.

The application of CEC to every initial chunklet  $C_i$ , for  $i = 1, \dots, k$ , produces a partition into final chunklets  $\mathcal{C} = \{C_j^{(i)} : i = 1, \dots, k, j = 1, \dots, k_i\}$ . Each final chunklet  $C_j^{(i)}$  is represented by a Gaussian density function  $f_j^{(i)} \sim \mathcal{N}(\hat{m}_j^{(i)}, \hat{\Sigma}_j^{(i)})$ , with a sample mean and a covariance matrix calculated within this chunklet as well as the associated weight  $p_j^{(i)}$ .

*Global clustering* An input to global clustering is a family of final chunklets  $\mathcal{C}$  and the maximum number of groups  $gr_{\max}$ . To adapt a clustering method to process such a dataset, we assume that each final chunklet is represented by a probability model calculated during inner clustering stage. More precisely, every final chunklet  $C_j^{(i)}$  is identified by a Gaussian probability density  $f_j^{(i)} \sim \mathcal{N}(\hat{m}_j^{(i)}, \hat{\Sigma}_j^{(i)})$ . Moreover, every model has an attached weight  $p_j^{(i)} = \frac{|C_j^{(i)}|}{|X|}$  proportional to the cardinality of the chunklet. For one element chunklet  $C_j^{(i)} = \{x\}$ , we put  $\hat{m}_j^{(i)} = x$ ,  $\hat{\Sigma}_j^{(i)} = 0$  and  $p_j^{(i)} = \frac{1}{|X|}$ . This does not define a Gaussian model, but a Dirac measure condensed at  $x$ . Nevertheless, we keep the symbol  $f_j^{(i)}$  to denote such a probability model. In consequence, we focus on clustering a set of probability models

$$\mathcal{M}(\mathcal{C}) = \{(p_j^{(i)}, f_j^{(i)}) : i = 1, \dots, k, j = 1, \dots, k_i\}.$$

To evaluate the CEC cost function (4) of a partition  $\mathcal{P}$  of  $\mathcal{M}(\mathcal{C})$  (which is now interpreted as a set of probability models), one has to know the covariance and probability coefficient of each cluster. This can be calculated using Observation 1. More precisely, the sample covariance matrix of a union of two final chunklets  $C_{i_1}, C_{i_2}$  is directly given by Observation 1. The sample covariance matrix of the union of  $l$  final chunklets  $C_{i_1}, \dots, C_{i_l}$  is calculated with use of a recursive formula:

$$\begin{aligned} \hat{\Sigma}_{\mathcal{P}} &= \hat{\Sigma}_{C_{i_1} \cup (C_{i_2} \cup \dots \cup C_{i_l})} \\ &= p_{i_1} \hat{\Sigma}_{i_1} + p_{i_2, \dots, i_l} \hat{\Sigma}_{i_2, \dots, i_l} + p_{i_1} p_{i_2, \dots, i_l} (\hat{m}_{i_1} - \hat{m}_{i_2, \dots, i_l})(\hat{m}_{i_1} - \hat{m}_{i_2, \dots, i_l})^T, \end{aligned}$$

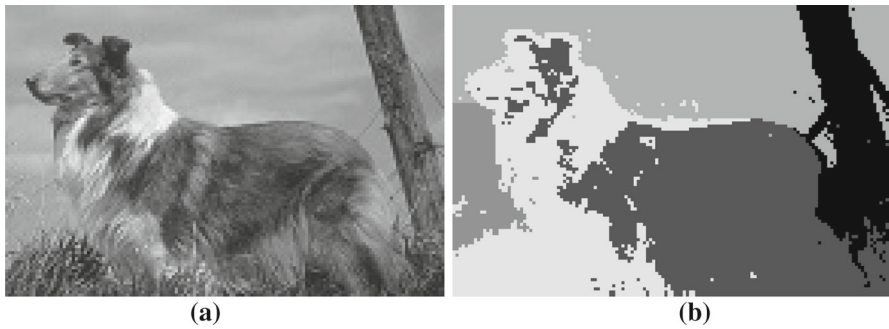
where we assume that  $\hat{m}_{i_2, \dots, i_l}$  and  $\hat{\Sigma}_{i_2, \dots, i_l}$  are known.

If any negative constraint is introduced, then the global clustering must additionally preserve the validity of partition. Since CEC relies on iterative switching the elements between clusters, then it is sufficient to verify conditions given in Theorem 3 and succeeding pseudocode. It is enough to incorporate this pseudocode to step 3.1 of the CEC procedure described in previous subsection.

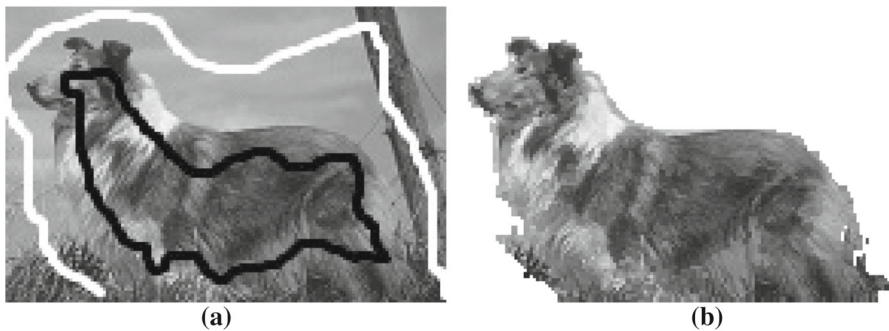
## 4 Experimental results

In this section the proposed C4S method is examined on the several datasets. First, it will be applied in a semi-supervised image segmentation, then it will be compared with other constrained clustering methods on selected examples retrieved from UCI repository and one real dataset of chemical compounds.





**Fig. 5** Image of the dog (a) and unsupervised CEC clustering (b)



**Fig. 6** Constraints specification (a) and the effect of our algorithm (b)—the dog was discovered properly

A demonstration version of the C4s is publicly available from the link: <http://ww2.ii.uj.edu.pl/~wiercioc/C4s/>. Please contact the second author for further information.

#### 4.1 Image segmentation

Let us consider a dog's image (Arbelaez et al. 2011) presented in the Fig. 5a. A natural question of the image segmentation is to discover dog's shape. As it can be seen in the Fig. 5b, the adaptation of classical CEC to images (Śmieja and Tabor 2013) produces five parts—two of them form the dog's shape.

Since it is difficult to perform an unsupervised segmentation which detects only two parts—the dog's shape and the background, we ask for examples of elements which should be grouped together. Figure 6a presents graphically the imposed constraints—pixels marked by hand in one color are restricted to be in the same group.

C4s reads these restrictions and in the first stage clusters individually elements with the same constraint. As a result, two groups from the first initial chunklet (elements marked in black) and three groups from the second initial chunklet (elements marked in white) were obtained. Then, the algorithm takes these five final chunklets and the rest of a dataset and performs the global clustering. Figure 6b shows the effect—the dog was discovered very well. All the clustering procedures were started with ten initial groups.

This semi-supervised scenario, in which a user indicates examples of objects to be extracted, often appears in real situations. In medical sciences a video capsule

**Table 1** Datasets used in the experiments

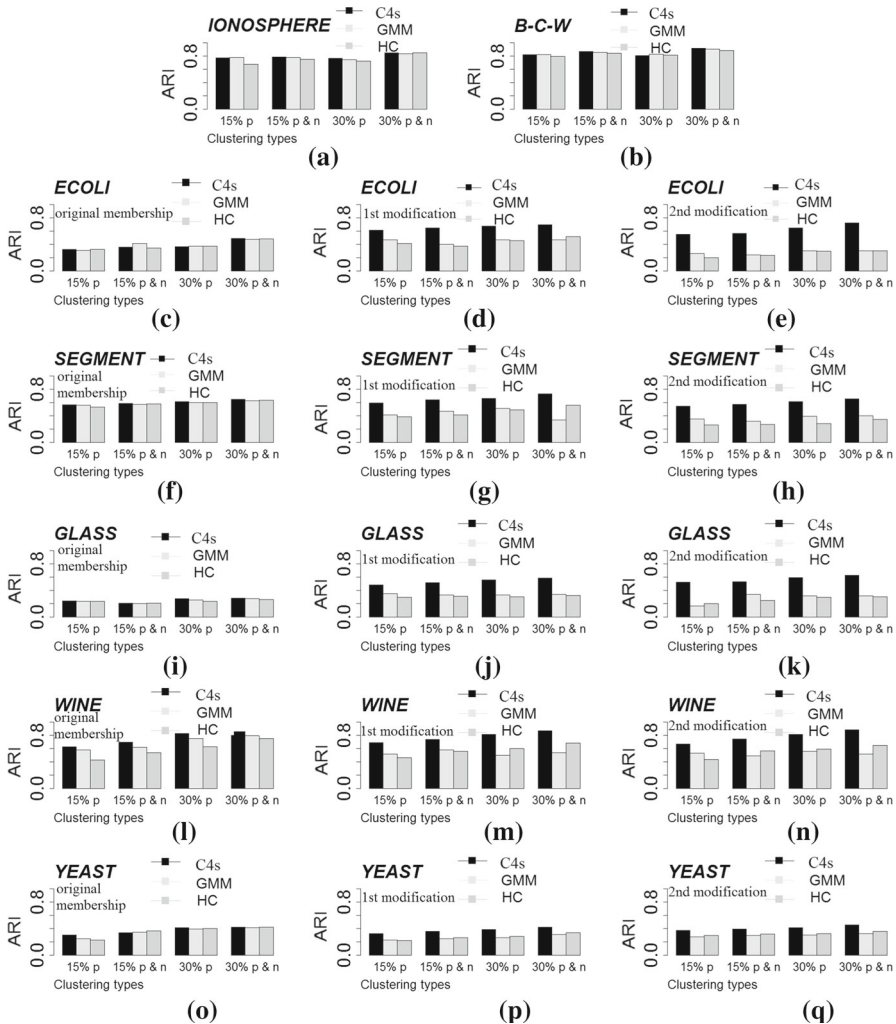
Dataset	Attributes	Instances	Classes
Ionosphere—original	34	351	{1}, {2}
B-C-W—original	10	699	{1}, {2}
<i>E. coli</i> —original	7	336	{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}
<i>E. coli</i> —1st modification	7	336	{1}, {2}, {3, 7}, {4}, {5}, {6, 8}*
<i>E. coli</i> —2nd modification	7	336	{1}, {2, 3}, {4}, {5, 7}, {6}, {8}*
Segment—original	19	2310	{1}, {2}, {3}, {4}, {5}, {6}, {7}
Segment—1st modification	19	2310	{1, 3}, {2, 4}, {5}, {6}, {7}*
Segment—2nd modification	19	2310	{1}, {2, 6}, {3, 4}, {5}, {6}*
Glass—original	9	214	{1}, {2}, {3}, {4}, {5}, {6}, {7}
Glass—1st modification	9	214	{1, 3}, {2}, {4}, {5, 7}, {6}*
Glass—2nd modification	9	214	{1}, {2, 3, 6}, {4}, {5}, {7}*
Wine—original	13	178	{1}, {2}, {3}
Wine—1st modification	13	178	{1, 3}, {2}*
Wine—2nd modification	13	178	{1}, {2, 3}*
Yeast—original	8	1484	{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}
Yeast—1st modification	8	1484	{1}, {2, 6}, {3}, {4}, {5, 8}, {7}, {9}, {10}*
Yeast—2nd modification	8	1484	{1}, {2}, {3, 9}, {4, 6}, {5}, {7}, {8}, {10}*

For each example from UCI repository we consider three variants of reference membership—original membership and two modified ones. Fourth column contains the reference classes. The classes marked with \* are constructed from the original ones

endoscopy is an examination where thousands of pictures are taken from inside of a gastrointestinal tract (Vyas et al. 2014). A doctor is not able to check manually an entire video of all patients in a search of lesions. In consequence, it might be more preferably to mark only a few interesting examples from pictures and then let the application to discover the rest of them. On the other hand, biologists must analyze a great amount of microscopic images of cells which might be impossible in practice (Wu et al. 2008). They often use computer tools, like ilastic (Sommer et al. 2011), which can perform automated semi-supervised image segmentation. Finally, pairwise equivalence constraints facilitate the detection of a person walking or tracking missiles as they are carried on a moving vehicle (in the army).

## 4.2 UCI repository

To compare the performance of C4s with the constrained versions of GMM and hierarchical clustering (HC) (Shental et al. 2004; Klein et al. 2002), we have tested it on several examples of datasets selected from the UCI repository (Lichman 2013). The results were evaluated with a use of adjusted Rand index (ARI) (Hubert and

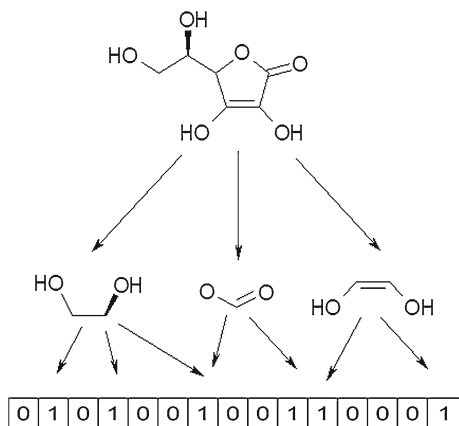


**Fig. 7** Adjusted Rand index of C4s, constrained GMM and HC over seven datasets from the UCI repository: Ionosphere, Breast Cancer Wisconsin, *E. coli*, Segment, Glass, Wine, Yeast for three types of reference partitions for each set (except the first two sets). The results are shown for four cases: using 15 % of the data points in positive constraints (15 % p), using 15 % of the data points in both positive and negative constraints (15 % p & n), using 30 % of the data points in positive constraints (30 % p), using 30 % of the data points in both positive and negative constraints (30 % p & n)

Arabic 1985) which is a well-known measure of agreement between two partitions. ARI assumes its maximum value 1 in the case of ideal agreement while for completely independent partitions it gives value 0.

In order to obtain side information, a teacher was employed. A teacher is given a random selection of  $M$  elements from a dataset and is then asked to partition this set of retrieved points into equivalence classes which are used as equivalence constraints. We carried out experiments with two criteria—when approximately 15 % of data points are constrained, and when approximately 30 % of data points are constrained.

**Fig. 8** Exemplary topological fingerprint of chemical compounds. Value 1 means presence, whereas value 0 means absence of predefined structural patterns



We tested all methods in two modes:

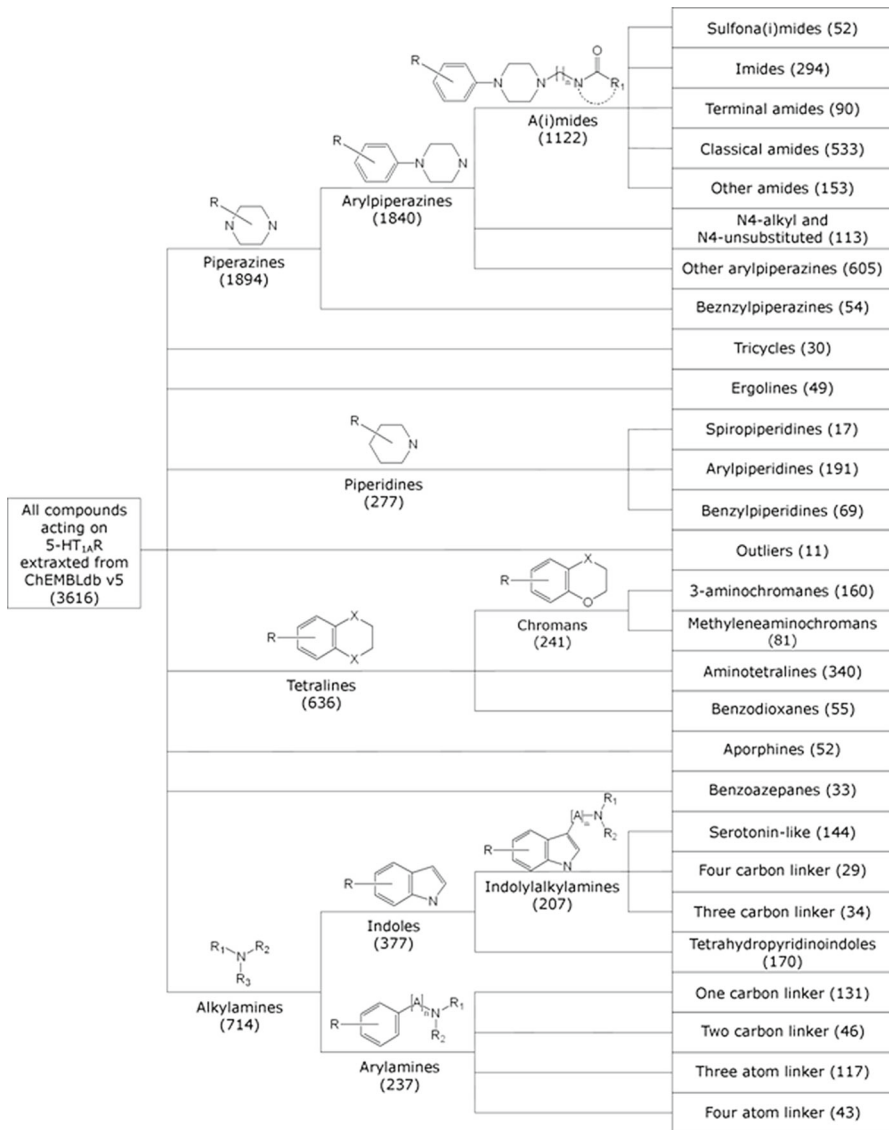
- using only positive equivalence constraints;
- using both positive and negative equivalence constraints.

Since GMM and C4s are nondeterministic we ran each of them 10 times and choose a result with the lowest value of cost function.

As mentioned in the previous section, our algorithm is intended to perform a clustering which discovers groups that are possibly generated from the mixture of models. In the present experiment, we consider three variants of the reference membership of each dataset: the first one is the original membership (defined by UCI) while the other two are modified by merging selected groups (except Ionosphere and B-C-W, where the partitions contain only two groups). In the second case some of the original groups are joined in order to obtain a reference partition where clusters are described by complex probability distributions. Table 1 provides detailed information connected with datasets and their modifications used in the simulations.

The following values were assumed as CEC parameters:  $gr = 3 * k$ , where  $k$  is a correct number of clusters,  $gr_i = 4$ ,  $\varepsilon = \varepsilon_i = 1\%$ . It should be noted that GMM and HC algorithms do not detect the correct number of clusters. For this reason, the number of clusters for these methods in certain mode equaled the number of clusters returned by C4s. Several observations follow from the results reported in Fig. 7:

- According to Fig. 7a–c, f, i, l, o the performance of all algorithms checked on the original reference partitions is almost identical.
- The advantage of C4s method is evident in the case of modified reference memberships (see Fig. 7d, e, g, h, j, k, m, n, p, q). The internal structure of each group becomes too complex after joining the clusters to be described just by one model. In consequence, C4s provides significantly higher ARI than constrained GMM and HC.
- After incorporating 30 % of random constraints, C4s gives the best value of agreement. Furthermore, in most cases adding negative constraints makes an improvement over results received when using only positive constraints.



**Fig. 9** Hierarchical structure of reference partition (Warszycki et al. 2013)

- Apart from that, the proposed algorithm detects quite precisely the right number of regions.

### 4.3 Chemical compounds

This experiment relies on grouping the selected set of chemical compounds with respect to their structural features. The set of compounds acting on central nervous system CNS (5-HT<sub>1A</sub> receptor ligands) was chosen for this example (Olivier et al.

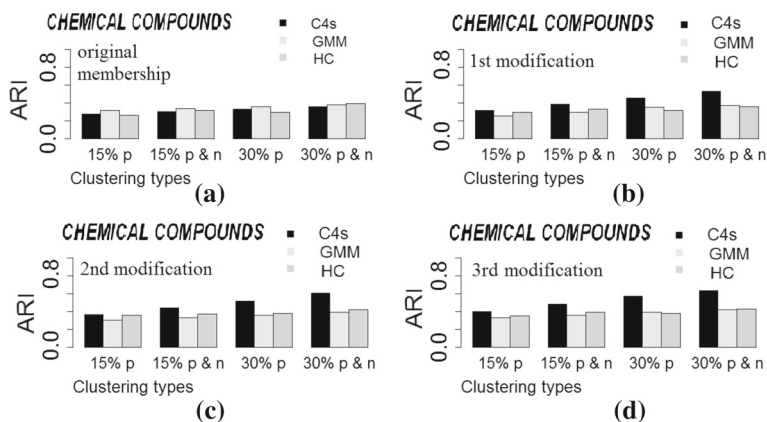
**Table 2** Four cases of reference grouping

Dataset	Attributes	Instances	Merged classes
Klekota Roth—original	4860	3696	–
Klekota Roth—1st modification	4860	3696	{3-aminochromanas, Methyleneaminochromans, Aminotetralines, Benzodioxanes}, {Aporhines, Benzoazepanes}
Klekota Roth—2nd modification	4860	3696	{Serotonin-like, Four carbon linker, Three carbon linker, Tetrahydropyridinoindoles}, {Aporhines, Benzoazepanes}, {3-aminochromanas, Methyleneaminochromans, Aminotetralines, Benzodioxanes}, {Tricycles, Ergolines}
Klekota Roth—3rd modification	4860	3696	{Four atom linker, Three atom linker, Two carbon linker, One carbon linker}, {Sulfona(i)mides, Imides, Terminal amides, Classical amides, Other amides, <i>N</i> <sub>4</sub> -alkyl and <i>N</i> <sub>4</sub> -unsubstituted, Other arylpiperazines, Benzylpiperazines}, {Serotonin-like, Four carbon linker, Three carbon linker, Tetrahydropyridinoindoles}, {Serotonin-like, Four carbon linker, Three carbon linker, Tetrahydropyridinoindoles}

Fourth column shows which groups of the lowest level of the hierarchy (Fig. 9) are merged to obtain reference partitions

1999; Śmieja and Warszycki 2016). The results were compared to the partition created manually by the experts (Warszycki et al. 2013).

Chemical compounds are usually represented by binary strings called fingerprints. The bit “1” means the presence of particular feature of compound while “0” denotes its absence (see Fig. 8). There are a lot of fingerprint representations since various features can be taken into account (Willett 2005). Our experiment uses Klekota–Roth



**Fig. 10** Adjusted Rand index of C4s, GMM and HC over dataset of chemical compounds acting on central nervous system. The results are shown for four cases: using 15 % of the data points in positive constraints (15 % p), using 15 % of the data points in both positive and negative constraints (15 % p & n), using 30 % of the data points in positive constraints (30 % p), using 30 % of the data points in both positive and negative constraints (30 % p & n)

fingerprint which provides reasonably good description of compound (it contains 4860 bits) (Klekota and Roth 2008).

The reference partition has a hierarchical structure (Fig. 9). One can decide how many groups should be taken into account. In the experiment, four different levels of the hierarchy were chosen and therefore four different reference groupings were obtained. In consequence, partitions into 28, 24, 18 and 12 groups were considered. Table 2 shows which groups from the lowest level of the reference hierarchy were merged in order to create a reference partition.

As in the example of UCI, the cases of 15 and 30 % of constrained points (both positive as negative) were examined and similar values of parameters for C4s were used. Moreover, the number of groups returned by C4s was assumed as the input to constrained GMM and HC.

The results shown in Fig. 10 clearly indicate that the advantage of proposed method increases when a reference partition contains more complex groups. When a reference clustering into 28 groups is assumed, all three examined methods provide similar values of ARI (Fig. 10a). The more groups were combined into larger clusters, the higher differences between C4s and the two other ones were observed (Fig. 10b–d). Moreover, introduced method gives significantly better results for a greater number of elements with constraints. It follows from the fact that the inner clustering processes are performed on sets of elements with the same positive constraints; i.e. the more elements are taken for the inner clustering, the more accurate results are obtained.

## 5 Conclusion and future work

In this paper we proposed a novel semi-supervised clustering technique, C4s, which incorporates equivalence constraints. The idea of introduced method was indebted to

work of Shental et al. (2004) who applied Gaussian mixture model to a clustering with strict pairwise constraints. The conceptual difference between these two algorithms lies in the number of components used to describe a cluster. C4S enables to understand a cluster as a complex structure which elements are generated by a mixture of simple models which is a novel concept in constrained clustering.

This reasoning is motivated by real-life examples where data is often classified in a hierarchical structure. Groups defined at the lowest level of hierarchy represent simple models, while their mixtures are used to describe clusters at higher levels. As an example one can consider an expert classification of chemical compounds (see Fig. 9).

The numerical results were consistent with an assumed theoretical model and confirmed that the proposed method is more suitable for data clustering when pairwise constraints suggest a complex structures of groups. It outperformed constrained GMM (Shental et al. 2004) as well as hierarchical clustering with equivalence constraints (Klein et al. 2002).

As mentioned in the paper, the introduced general algorithm can be implemented in combination with various clustering methods. This study assumed the existence of clusters described by Gaussian mixtures and applied CEC method. In the future, we plan to consider different techniques which are suited for particular form of data. Moreover, it would be also a challenge to modify this procedure to the case of soft constraints which can be occasionally violated during grouping (Bilenko et al. 2004; Lu and Leen 2004; Wang and Davidson 2010).

**Acknowledgements** We are grateful to anonymous reviewers and the Editor for their important comments and critics of this paper. The authors would also like to thank Prof. Jacek Tabor for constructive discussions and invaluable suggestions. We are also thankful to Dawid Warszycki for providing access to the cheminformatics data.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Arbelaez P, Maire M, Fowlkes C, Malik J (2011) Contour detection and hierarchical image segmentation. *IEEE Trans Pattern Anal Mach Intell* 33(5):898–916. doi:10.1109/TPAMI.2010.161
- Bar-Hillel A, Hertz T, Shental N, Weinshall D (2003) Learning distance functions using equivalence relations. In: *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, August 21–24, 2003. DC, USA, AAAI Press, Washington, pp 11–18
- Basu S, Banerjee A, Mooney RJ (2002) Semi-supervised clustering by seeding. In: *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002)*, July 8–12, 2002. Australia, Morgan Kaufmann, Sydney, pp 27–34
- Baudry JP, Cardoso M, Celeux G, Amorim M, Ferreira A (2015) Enhancing the selection of a model-based clustering with external categorical variables. *Adv Data Anal Classif* 9(2):177–196. doi:10.1007/s11634-014-0177-3
- Bellas A, Bouveyron C, Cottrell M, Lacaille J (2013) Model-based clustering of high-dimensional data streams with online mixture of probabilistic PCA. *Adv Data Anal Classif* 7:281–300
- Bennett KP, Demiriz A (1998) Semi-supervised support vector machines. In: *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, pp 368–374



- Bilenko M, Basu S, Mooney RJ (2004) Integrating constraints and metric learning in semi-supervised clustering. In: Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), July 4–8, 2004, Banff, Alberta, Canada, ACM, New York, NY, USA, p 11. doi:[10.1145/1015330.1015360](https://doi.org/10.1145/1015330.1015360)
- Cayton L (2005) Algorithms for manifold learning. University of California at San Diego Tech Rep, pp 1–17
- Collingwood EF, Lohwater AJ (2004) The theory of cluster sets. Cambridge University Press, Cambridge
- Ding JJ, Wang YH, Hu LL, Chao WL, Shau YW (2011) Muscle injury determination by image segmentation. In: Visual Communications and Image Processing (VCIP), 2011 IEEE, pp 1–4. doi:[10.1109/VCIP.2011.6115925](https://doi.org/10.1109/VCIP.2011.6115925)
- Hartigan JA, Wong MA (1979) Algorithm AS 136: a k-means clustering algorithm. *J R Stat Soc Ser C (Appl Stat)* 28(1):100–108
- Hennig C (2010) Methods for merging Gaussian mixture components. *Adv Data Anal Classif* 4(1):3–34. doi:[10.1007/s11634-010-0058-3](https://doi.org/10.1007/s11634-010-0058-3)
- Hruschka ER, Campello RJGB, Freitas AA, De Carvalho ACPLF (2009) A survey of evolutionary algorithms for clustering. *IEEE Trans Syst Man Cybern C Cybern* 39(2):133–155
- Hubert L, Arabie P (1985) Comparing partitions. *J Classif* 2:193–218
- Jain AK (2010) Data clustering: 50 years beyond k-means. *Pattern Recogn Lett* 31(8):651–666. doi:[10.1016/j.patrec.2009.09.011](https://doi.org/10.1016/j.patrec.2009.09.011)
- Jain AK, Murty NM, Flynn PJ (1999) Data clustering: a review. *ACM Comput Surv* 31:264–323
- Klein D, Kamvar SD, Manning CD (2002) From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), July 8–12, 2002. Australia, Morgan Kaufmann, Sydney, pp 307–314
- Klekota J, Roth FP (2008) Chemical substructures that enrich for biological activity. *Bioinformatics* 24(21):2518–2525
- Lee S, McLachlan G (2013) On mixtures of skew normal and skew t-distributions. *Adv Data Anal Classif* 7(3):241–266. doi:[10.1007/s11634-013-0132-8](https://doi.org/10.1007/s11634-013-0132-8)
- Li Z, Liu J, Tang X (2009) Constrained clustering via spectral regularization. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp 421–428. doi:[10.1109/CVPR.2009.5206852](https://doi.org/10.1109/CVPR.2009.5206852)
- Lichman M (2013) UCI machine learning repository. <http://archive.ics.uci.edu/ml>
- Lu Z, Leen TK (2004) Semi-supervised learning with penalized probabilistic clustering. In: NIPS
- McLachlan G, Krishnan T (2008) The EM algorithm and extensions, Wiley series in probability and statistics, 2nd edn. Wiley, Hoboken
- McNicholas PD, Murphy TB (2010) Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics* 26(21):2705–2712
- Melnykov V, Melnykov I, Michael S (2015) Semi-supervised model-based clustering with positive and negative constraints. *Adv Data Anal Classif* 1–23. doi:[10.1007/s11634-015-0200-3](https://doi.org/10.1007/s11634-015-0200-3)
- Morlini I (2012) A latent variables approach for clustering mixed binary and continuous variables within a Gaussian mixture model. *Adv Data Anal Classif* 6(1):5–28
- Morris K, McNicholas P, Scrucca L (2013) Dimension reduction for model-based clustering via mixtures of multivariate t-distributions. *Adv Data Anal Classif* 7(3):321–338. doi:[10.1007/s11634-013-0137-3](https://doi.org/10.1007/s11634-013-0137-3)
- Narayanan H, Mitter S (2010) Sample complexity of testing the manifold hypothesis. In: Advances in Neural Information Processing Systems, pp 1786–1794
- Nguyen HD, McLachlan GJ (2015) Maximum likelihood estimation of Gaussian mixture models without matrix operations. *Adv Data Anal Classif* 9(4):371–394
- Olivier B, Soudijn W, van Wijngaarden I (1999) The 5-ht<sub>1A</sub> receptor and its ligands: structure and function. In: Jucker E (ed) Progress in Drug Research, Progress in Drug Research, vol 52, pp 103–165
- Pavel B (2002) Survey of clustering data mining techniques. Technical report, Accrue Software
- Rubinstein RY, Kroese DP (2004) The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-Carlo Simulation (Information Science and Statistics). Springer-Verlag New York Inc, Secaucus, NJ, USA
- Samuelsson J (2004) Waveform quantization of speech using Gaussian mixture models. In: Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, vol 1, pp I–165–8, vol 1. doi:[10.1109/ICASSP.2004.1325948](https://doi.org/10.1109/ICASSP.2004.1325948)

- Scrucca L, Raftery AE (2015) Improved initialisation of model-based clustering using Gaussian hierarchical partitions. *Adv Data Anal Classif* 9(4):447–460
- Shental N, Bar-Hillel A, Hertz T, Weinshall D (2004) Computing Gaussian mixture models with EM using equivalence constraints. *Adv Neural Inf Process Syst* 16(8):465–472
- Śmieja M, Tabor J (2013) Image segmentation with use of cross-entropy clustering. In: *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*. Springer, *Advances in Intelligent Systems and Computing*, pp 403–409
- Śmieja M, Tabor J (2015a) Entropy approximation in lossy source coding problem. *Entropy* 17(5):3400–3418
- Śmieja M, Tabor J (2015b) Spherical Wards clustering and generalized Voronoi diagrams. In: *Data Science and Advanced Analytics (DSAA), 2015*. 36678 2015. *IEEE International Conference on*, IEEE, pp 1–10
- Śmieja M, Warszycki D (2016) Average information content maximization—a new approach for fingerprint hybridization and reduction. *PLoS One* 11(1):e0146666
- Sommer C, Strähle C, Köthe U, Hamprecht FA (2011) ilastik: Interactive Learning and Segmentation Toolkit. In: *Eighth IEEE International Symposium on Biomedical Imaging (ISBI)*. *Proceedings*, pp 230–233. doi:[10.1109/ISBI.2011.5872394](https://doi.org/10.1109/ISBI.2011.5872394)
- Spurek P, Tabor J, Zając E (2013) Detection of disk-like particles in electron microscopy images. In: *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, Springer, pp 411–417
- Subedi S, McNicholas P (2014) Variational Bayes approximations for clustering via mixtures of normal inverse Gaussian distributions. *Adv Data Anal Classif* 8(2):167–193. doi:[10.1007/s11634-014-0165-7](https://doi.org/10.1007/s11634-014-0165-7)
- Tabor J, Misztal K (2013) Detection of elliptical shapes via cross-entropy clustering. In: *Pattern Recognition and Image Analysis*, Springer, Berlin 7887:656–663
- Tabor J, Spurek P (2014) Cross-entropy clustering. *Pattern Recogn* 47(9):3046–3059. doi:[10.1016/j.patcog.2014.03.006](https://doi.org/10.1016/j.patcog.2014.03.006)
- Telgarsky M, Vattani A (2010) Hartigan’s method: k-means clustering without Voronoi. In: Teh YW, Titterton DM (eds) *AISTATS, JMLR.org, JMLR Proceedings*, vol 9, pp 820–827
- Vyas R, Gao J, Cheng L, Du P (2014) An image-based model of the interstitial cells of cajal network in the gastrointestinal tract. In: Goh J (ed) *The 15th International Conference on Biomedical Engineering, IFMBE Proceedings*, vol 43, Springer International Publishing, pp 5–8
- Wagstaff K, Cardie C, Rogers S, Schrödl S (2001) Constrained k-means clustering with background knowledge. In: *Machine Learning, Proceedings of the Eighteenth International Conference (ICML 2001)*, June 28–July 1, 2001. Williams College, Williamstown, MA, USA, Morgan Kaufmann, pp 577–584
- Wang X, Davidson I (2010) Flexible constrained spectral clustering. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, KDD ’10, pp 563–572. doi:[10.1145/1835804.1835877](https://doi.org/10.1145/1835804.1835877)
- Warszycki D, Mordalski S, Kristiansen K, Kafel R, Sylte I, Chilmonczyk Z, Bojarski AJ (2013) A linear combination of pharmacophore hypotheses as a new tool in search of new active compounds—an application for 5-HT<sub>1A</sub> receptor ligands. *PLoS One* 8(12):e84,510. doi:[10.1371/journal.pone.0084510](https://doi.org/10.1371/journal.pone.0084510)
- Willett P (2005) Searching techniques for databases of two- and three-dimensional chemical structures. *J Med Chem* 48(13):4183–4199. doi:[10.1021/jm0582165](https://doi.org/10.1021/jm0582165)
- Wolfe J (1963) *Object cluster analysis of social areas*. University of California
- Wu Q, Merchant FA, Castleman KR (2008) *Microscope image processing*. Elsevier/Academic Press, Amsterdam
- Xiong Z, Chen Y, Wang R, Huang T (2002) Improved information maximization based face and facial feature detection from real-time video and application in a multi-modal person identification system. In: *Multimodal Interfaces, 2002*. *Proceedings. Fourth IEEE International Conference on*, pp 511–516. doi:[10.1109/ICMI.2002.1167048](https://doi.org/10.1109/ICMI.2002.1167048)
- Xu R, Wunsch D (2009) *Clustering*. Wiley-IEEE Press, Hoboken
- Xu R, Wunsch I (2005) Survey of clustering algorithms. *IEEE Trans Neural Netw* 16:645–678