Mach Learn (2015) 98:217–268
DOI 10.1007/s10994-013-5394-z

Unsupervised ensemble minority clustering

Edgar Gonzàlez · Jordi Turmo

Received: 16 May 2012 / Accepted: 8 June 2013 / Published online: 3 July 2013

Abstract Cluster analysis lies at the core of most unsupervised learning tasks. However, the majority of clustering algorithms depend on the all-in assumption, in which all objects belong to some cluster, and perform poorly on minority clustering tasks, in which a small fraction of signal data stands against a majority of noise.

The approaches proposed so far for minority clustering are supervised: they require the number and distribution of the foreground and background clusters. In supervised learning and all-in clustering, combination methods have been successfully applied to obtain distribution-free learners, even from the output of weak individual algorithms.

In this work, we propose a novel ensemble minority clustering algorithm, EWOCS, suitable for weak clustering combination. Its properties have been theoretically proved under a loose set of constraints. We also propose a number of weak clustering algorithms, and an unsupervised procedure to determine the scaling parameters for Gaussian kernels used within the task.

We have implemented a number of approaches built from the proposed components, and evaluated them on a collection of datasets. The results show how approaches based on EWOCS are competitive with respect to—and even outperform—other minority clustering approaches in the state of the art.

Keywords Clustering · Minority clustering · Ensemble clustering · Weak learning

Editors: Emmanuel Müller, Ira Assent, Stephan Günneman, Thomas Seidl, Jennifer Dy.

E. Gonzàlez (✉) · J. Turmo
TALP Research Center, Universitat Politècnica de Catalunya, c/Jordi Girona, 1, 08034 Barcelona, Spain
e-mail: egonzalez@lsi.upc.edu

J. Turmo
e-mail: turmo@lsi.upc.edu

Present address:
E. Gonzàlez
Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA

1 Introduction

The amount of data available in digital form is increasing every day. Given the expensive costs of human inspection (and annotation), unsupervised approaches to mining these data become more and more paramount.

Cluster analysis lies at the core of most unsupervised learning tasks. Jain et al. (1999) define clustering as "*the organization of a collection of patterns* [...] *into clusters based on similarity. Intuitively, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster*". In addition to *pattern*, each element to be clustered has also received the names of "*object, record, point, vector*, [...] *event, case, sample, observation, or entity*" (Tan et al. 2005, ch. 2). We will stick to the term *object* thorough this article.

In this the most common setting, it is assumed that all objects belong to some cluster. Even if several surveys have reviewed the vast literature on clustering methods (Dubes and Jain 1980; Jain et al. 1999; Xu and Wunsch 2005), so far they all have focused on this standard task, which can be named *all-in clustering*. Two of the most widely used methods to solve it are the distance-based k-means (MacQueen 1967) and the probabilistic-model-based Expectation-Maximization (Dempster et al. 1977) algorithms.

However, there is a number of situations in which the data are known not to fit neatly within this all-in assumption. In such cases, we know there is a fraction of data which are neither similar to one another nor to the data within the clusters. Often, these data will correspond to a certain form of *noise* and should hence be separated from the sought regular clusters, which constitute the *signal*. Within this alternative setting, a number of different tasks can be identified according to the characteristics of the data and the aim of the task itself.

In one of these tasks, the all-in clustering goal is preserved, but the data are known to contain a small fraction of noise. This has been called the *robust clustering* task (Davé and Krishnapuram 1997). To solve it, some authors have proposed changes to standard clustering methods to make them more robust to the presence of noise (Kaufman and Rousseeuw 2005; Peel and McLachlan 2000). Other approaches explicitly incorporate a noise cluster, often with different properties from the regular signal clusters (Davé 1991; Banfield and Raftery 1993; Guillemaud and Brady 1997; Biernacki et al. 2000). A last family is that of algorithms specifically devised for robust clustering, such as BIRCH (Zhang et al. 1996) or DBSCAN (Ester et al. 1996).

It is worth noticing that there is a number of related tasks which share this setting, such as *one-class classification* or *learning* (Moya et al. 1993; Schölkopf et al. 2001; Tax and Duin 2004) and *outlier detection* (Hodge and Austin 2004; Chandola et al. 2009). In both cases, there is also a dataset containing both signal and a fraction of noise objects. However, the focus of these tasks shifts away from that of clustering, becoming the estimation of a model which covers the signal objects in the former, and the detection of the objects that significantly deviate from the rest in the latter.

A different task is that in which there is only a minority of signal objects, standing against the majority of noise. Most often, the signal objects will be embedded within the noise ones, becoming respectively *foreground* and *background* objects, and the distinction between the former and the latter must be done on grounds of density criteria. In the literature, this task has been compared to "*clustering needles in a haystack*" (Ando 2007), and has received names such as *one-class clustering* (Crammer and Chechik 2004), *density-based clustering* (Gupta and Ghosh 2006) or *minority detection* (Ando and Suzuki 2006). As a catchall term, in this article we will refer to this setting and task as *minority clustering*. An example dataset for a minority clustering problem is depicted in Fig. 1.

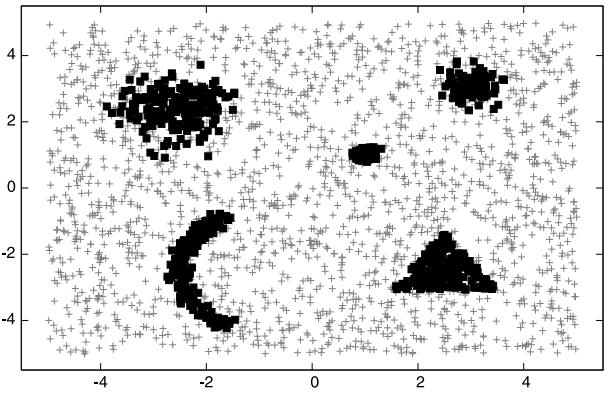


Fig. 1 Sample TOY minority clustering dataset

Even if this new task is related to the previously presented ones, the reversal of the signal-to-noise ratio can make existing approaches unsuitable. For instance, Crammer and Chechik (2004) give insights into why existing one-class classification approaches, which are tailored to finding large-scale structures, may be unable to identify small and locally dense regions embedded in noise. Empirical comparisons have also stated the low performance exhibited by all-in and robust clustering methods in the minority clustering task (Gupta and Ghosh 2006).

However, to the best of our knowledge, all the methods proposed so far require as an input the distribution of the foreground clusters or both the foreground clusters and the background noise, either in the form of a probability distribution or, equivalently, of a divergence metric.[1] This can become a significant issue when facing large amounts of data coming from a new and unexplored domain, whose distribution may be completely unknown.

With the aim of providing a way to obtain distribution-free methods, a number of combination methods have appeared for all-in clustering (e.g., Strehl and Ghosh 2002; Topchy et al. 2003, 2004; Gionis et al. 2005). Among them, Topchy et al. (2003) introduced the idea of using an ensemble of weak clusterings, which "*produce a partition, which is only slightly better than a random partition of the data*", to obtain a high-quality consensus clustering.

Ensemble clustering methods are known to offer a greater degree of flexibility with respect to individual algorithms. They allow the reusal of knowledge coming from multiple and heterogeneous sources, and can be used in a number of settings which are unfeasible using monolithic approaches, such as feature-distributed or privacy-preserving clustering (Strehl and Ghosh 2002). Moreover, most of them can be considered *embarrassingly parallel*, and as such can obtain significant speed-ups when deployed in distributed environments, using techniques such as Map/Reduce (Dean and Ghemawat 2004).

In this article, we make a three-fold proposal:

– First, we propose an unsupervised minority clustering approach, Ensemble Weak minOrity Cluster Scoring (EWOCS), based on weak-clustering combination. In it, a number of weak clusterings are generated, and the information coming from each one of them is combined to obtain a score for each object. A threshold separating foreground from background objects is then inferred from the distribution of these scores. We have been able

[1] A Bregman divergence induces a probability distribution of the exponential family (Banerjee et al. 2005).

to find a theoretical proof of the properties of the proposed method, and we consider a number of criteria by which the threshold value can be determined.
– Second, we propose Random Bregman Clustering (RBC), a weak clustering algorithm based on Bregman divergences, for use within EWOCS ensembles; as well as an extension of the Random Splitting (RSPLIT) weak clustering algorithm of Topchy et al. (2003).
– Third, we propose an unsupervised procedure to determine a set of suitable scaling parameters for a Gaussian kernel, to be used within RBC.

We have implemented a number of approaches built from the proposed components, and evaluated them on a collection of datasets. The results of the evaluation show how approaches based on EWOCS are competitive with respect to—and even outperform—other minority clustering approaches in the state of the art, in terms of F1 and AUC measures of the obtained clusterings.

The EWOCS algorithm has already been used in the real-world task of relation detection, which was reduced to a minority clustering problem (Gonzàlez and Turmo 2009). However, we now provide a formalization of the approach, as a minority clustering algorithm by itself, and a study of its theoretical properties, which were both missing from our previous work.

The rest of the article is organized as follows. Sect. 2 gives an overview of related work in the fields of minority clustering and clustering combination. Next, Sect. 3 contains a description of the EWOCS approach, particularly the derivation of a minority clustering algorithm whose properties are theoretically proved under a set of conditions. The obtained algorithm has a number of components which allow different implementations: Sects. 4 and 5 give details on the specific weak clustering algorithms and threshold score determination methods we have used, respectively. Sects. 6 and 7 contain the details and results of an empirical evaluation of the proposed approaches on synthetic and real-world data, respectively. Finally, Sect. 8 draws conclusions of our work.

2 Related work

One of the first works to identify the minority clustering task in opposition to that of one-class classification is that of Crammer and Chechik (2004). The authors formalize the problem in terms of the Information Bottleneck principle (IB) (Tishby et al. 1999), and provide a sequential algorithm to solve this one-class IB problem. Given a Bregman divergence as a generalized measure of object discrepancy, and a fixed radius value, the OC-IB method outputs a centroid for a single dense cluster. The foreground cluster consists of the objects which fall inside the Bregmanian ball of given radius centered around the given centroid. More recently, Crammer et al. (2008) propose a different algorithm for the same model, based in rate-distortion theory and the Blahut-Arimoto algorithm, and extend it to allow for more than one cluster.

In a different direction, Gupta and Ghosh (2005) reformulate the problem in terms of cost, defined as the sum of divergences from the cluster centroid to each sample within it, and extend the OC-IB method to avoid local minima. A triad of methods (HOCC, BBOCC and Hyper-BB) is proposed. However, the requirement of an a priori determination of the cluster radius (or equivalently, size) is not removed, and the output remains a single ball-shaped cluster.

To overcome this second limitation, Gupta and Ghosh (2006) propose Bregman Bubble Clustering (BBC), as a generalization of BBOCC to several clusters. However, the number of such clusters must still be given a priori, as well as the desired joint cluster size. The authors also propose a soft clustering version of BBC, as well as a unified framework between

all-in Bregman clustering (Banerjee et al. 2005) and BBC, in all their hard and soft versions. Ghosh and Gupta (2011) revisit all the theory of BBC, and present Density Gradient Enumeration (DGRADE), a procedure to determine the number of clusters as well as the initial centroids for BBC. However DGRADE introduces new parameters of its own, whose tuning requires a potentially expensive exhaustive search in the space of possible values.

The work of Ando and Suzuki (2006) is similar to previous ones in that it also uses the Information Bottleneck principle as a criterion to identify a single minority cluster. However, the method is more general in the sense that it allows arbitrary distributions, not only those induced by Bregman divergences, as foreground and background. Ando (2007) extends this last proposal, allowing multiple foreground clusters, and also provides a unifying framework of which not only the task of minority clustering, but also those of outlier detection and one-class learning, are particular cases.

A last line of research is that opened by Gupta et al. (2010), who propose Hierarchical Density Shaving (HDS). HDS is built upon the Hierarchical Mode Analysis algorithm (HMA) introduced forty years before by Wishart (1969), and can be seen as a generalization of the robust clustering algorithm DBSCAN (Ester et al. 1996). The algorithm produces a hierarchical clustering which is an approximation of the one which would be obtained by HMA. Dense clusters are then identified in the hierarchy using a heuristic criterion. The authors propose the AutoHDS framework, in which the parameters of the algorithm are manually tuned with the help of an interactive tool. The proposed application provides a visualization of the obtained minority clustering as the parameter values are updated.

Except for HDS, which is of a more heuristic nature, all the approaches discussed so far formalize the task of minority clustering as an optimization problem, and differ in the considered objective function and in the algorithm used to optimize it. In all cases, the formalization requires to make explicit the distribution of the sought clusters. In the case of HDS, a divergence function is required, and used throughout the algorithm. This is clearly a drawback, as the performance of these methods degrades if the distribution of the data does not match the one used by the model.

As discussed thoroughly in Sect. 3, EWOCS provides a different approach to the problem: we propose a procedure, based on aggregation of clustering ensembles, by which a score for each object can be found, and we show how these scores correlate with the fact of whether an object belongs to the foreground clusters or to the background. This alternative approach allows the use of much less informed (weak) clustering algorithms, and is the first one to our knowledge to use ensembles for the task. In addition to providing a distribution-free clusterer, the use of ensembles also brings practical benefits, as the algorithm becomes easily parallelizable: the individual clusterings can be found in a distributed fashion, and synchronization is only needed in batches to add up the object scores.

3 EWOCS

This section presents our Ensemble Weak minOrity Cluster Scoring (EWOCS) algorithm to solve the task of minority clustering.

As mentioned in the introduction, the aim of EWOCS is to leverage the information provided by clusterings in an ensemble, combining the evidence from each one of them to obtain a minority clustering of the dataset. The central idea in the algorithm is that of *object score*: from each individual clustering, objects are assigned a certain score, and these scores will be aggregated across the ensemble. In order to quantify their *density*, we propose the use of a score related to the size of the clusters each object is assigned to across the clusterings. In addition to being computationally cheap, we will be able to prove that this function

shows interesting theoretical properties in minority clustering scenarios. As a consequence of them, it will be possible to separate foreground from background objects using a threshold on their aggregated scores.

The following sections detail and formalize the intuitions presented in this overview. First Sect. 3.1 defines our setting for the task of minority clustering. Sect. 3.2 presents, from a theoretical point of view, the scoring scheme that lies at the core of our method. Sects. 3.3 and 3.4 then study the conditional probability distributions of the assigned scores: the first one on a single dataset; the second, across multiple dataset samplings. Next, Sect. 3.5 introduces the concept of *consistent clustering*, and shows how, when using clustering functions from a consistent family, an inequality on the score expectations for foreground and background objects can be established. This inequality will allow us to obtain as a corollary, in Sect. 3.6, a generic algorithmic procedure for minority clustering, based on the proposed scores. Finally, it is also possible to obtain a clustering model using this algorithm: its construction and application is described in the last Sect. 3.7.

3.1 Task setting

Our definitions of clustering are based on concepts from fuzzy set theory:

Definition 1 (Fuzzy set) A **fuzzy set** over an ordinary set $\mathcal{X}$ is a pair $\tilde{\mathcal{X}} = (\mathcal{X}, f_{\tilde{X}})$, where $f_{\tilde{X}} : \mathcal{X} \to [0, 1]$ is the **membership function** (or **characteristic function**) of $\tilde{\mathcal{X}}$. For $x_i \in \mathcal{X}$, $f_{\tilde{X}}(x)$ expresses the **grade** of membership of x_i to $\tilde{\mathcal{X}}$, and will often be denoted as grade$(x_i, \tilde{\mathcal{X}})$ (Zadeh 1965).

Definition 2 (Fuzzy c-partition) A **fuzzy c-partition** (or **fuzzy pseudopartition**) of an ordinary set $\mathcal{X}$ is a family of fuzzy sets $\Pi = \{\pi_1 \dots \pi_k\}$ over $\mathcal{X}$ such that

$$\forall x \in \mathcal{X} : \quad \sum_{\pi_c \in \Pi} f_{\pi_c}(x) = 1$$

$$\forall \pi_c \in \Pi : \quad 0 < \sum_{x \in \mathcal{X}} f_{\pi_c}(x) < \|\mathcal{X}\|$$

(Bezdek 1981; Klir and Yuan 1995).

A *clustering* over a dataset $\mathcal{X} = \{x_1 \dots x_n\}$ of size n can now be defined as:

Definition 3 (Hard partitional clustering) A **hard (partitional) clustering** Π of dataset $\mathcal{X}$ is a partition $\Pi = \{\pi_1 \dots \pi_k\}$ of $\mathcal{X}$. Each one of the subsets $\pi_c \in \Pi$ is a **hard cluster**.

Definition 4 (Soft partitional clustering) A **soft (partitional) clustering** Π of dataset $\mathcal{X}$ is a fuzzy pseudopartition $\Pi = \{\pi_1 \dots \pi_k\}$ of $\mathcal{X}$. Each one of the fuzzy subsets $\pi_c \in \Pi$ is a **soft cluster**.

Remark 1 A hard clustering can be seen as a particular case of soft clustering where the grade of membership of a certain x_i to the π_c is zero for all but exactly one cluster, for which the grade is one.

Assume we have a finite set of $\hat{k}$ generative distributions or *sources* $\Psi = \{\psi_1 \dots \psi_{\hat{k}}\}$, with a priori probabilities $\{\alpha_1 \dots \alpha_{\hat{k}}\}$, from which the dataset $\mathcal{X}$ has been sampled. Each object x_i

will be generated by one of the sources in Ψ, and we can hence consider a set $\mathcal{Y}$ of hidden variables, with each $y_i \in \Psi$ containing the source which generated the corresponding x_i.

The setting presented so far is common to all-in clustering and minority clustering. However, in the latter we can make additional assumptions about the sources in Ψ. In particular, and without loss of generality, we can assume the first of those sources, ψ_1, to be a *background source*; and the objects generated by it, the *background objects*. The rest of sources and objects shall be named the *foreground sources* (whose set will be denoted as Ψ^+) and the *foreground objects*, respectively.

The relationship between background and foreground sources satisfies two additional assumptions which can be stated as follows:

Density: Foreground sources are *dense*, i.e., objects generated by the same foreground source are more similar to each other than to those generated by the background source.[2]

Locality: Foreground sources are *local*, i.e., objects generated by different foreground sources are as similar to each other as they are to those generated by the background source.

These two assumptions are similar to those in previous works, for instance, those of *atypicalness* and *local distribution* defined by Ando (2007), and are implicitly present in others (e.g., Gupta and Ghosh (2006) look for "*locally dense regions*"). In fact, we consider these assumptions to *define* our task. Thus, minority clustering (in contrast to, for instance, all-in clustering and robust clustering) can be defined as:

Definition 5 (Minority clustering) **Minority clustering** is the task of organizing a collection of objects based on similarity, when we can assume that a minority fraction of them are dense and local, and are embedded in a majority which are not.

3.2 Per-clustering scoring

Assume now we have a (possibly infinite) family of *clustering functions* F. From them, a sequence of functions $(f_1 \ldots)$ are independently drawn at random, with a certain probability density. When applied to the dataset, each f_r will produce a soft[3] clustering $\Pi_r = \{\pi_{r1} \ldots \pi_{rk_r}\}$ with a number k_r of clusters.

After clustering function f_r is applied, the *cluster size* and *object scores* can be calculated from the output clustering Π_r.

Definition 6 (Cluster size) The **size** of cluster π_{rc} is the sum of the grade of membership to the cluster of all objects in the dataset:

$$\text{size}(\pi_{rc}) = \sum_{x_i \in \mathcal{X}} \text{grade}(x_i, \pi_{rc}) \tag{1}$$

Definition 7 (Object score) The **score** of an object x_i by clustering function f_r is

$$s_{ri} = \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{size}(\pi_{rc}) \tag{2}$$

[2] This concept of *density* is related to traditional *probability density* in the sense that foreground clusters will correspond to regions in which the value of the global probability density function is higher than in their neighbourhood.

[3] The result is also valid for hard clustering families, being a particular case of soft clustering.

i.e., the sum of the sizes of the output clusters, weighted by the grade of membership of x_i to each one of them.

An additional concept will turn out to be of much importance later.

Definition 8 (Co-occurrence vector) The **co-occurrence vector** for object x_i and clustering function f_r is $\mathbf{c}_{ri} = [c_{ri1} \dots c_{rin}]^T$, where each component c_{rij} is

$$c_{rij} = \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{grade}(x_j, \pi_{rc}) \tag{3}$$

Remark 2 Using the co-occurrence vector, the score of object x_i by clustering function f_r can be written as

$$\begin{aligned} s_{ri} &= \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{size}(\pi_{rc}) \\ &= \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \sum_{x_j \in \mathcal{X}} \text{grade}(x_j, \pi_{rc}) \\ &= \sum_{\pi_{rc} \in \Pi_r} \sum_{x_j \in \mathcal{X}} \text{grade}(x_i, \pi_{rc}) \cdot \text{grade}(x_j, \pi_{rc}) \\ &= \sum_{x_j \in \mathcal{X}} \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{grade}(x_j, \pi_{rc}) \\ &= \sum_{x_j \in \mathcal{X}} c_{rij} \end{aligned}$$

From its definition, we can infer that the co-occurrence vector will satisfy the following property:

Proposition 1 *The values of the entries c_{rij} in the co-occurrence vector belong to the interval* $[0, 1]$.

Proof By the properties of fuzzy pseudopartitions, and hence of soft clusterings, we know that

$$\forall x_i : \quad \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) = 1$$

The product of two of these terms, which will also be equal to 1, can be expressed as

$$\begin{aligned} 1 &= \left(\sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \right) \cdot \left(\sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_j, \pi_{rc}) \right) \\ &= \sum_{\pi_{rc}, \pi_{rc'} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{grade}(x_j, \pi_{rc'}) \\ &= \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{grade}(x_j, \pi_{rc}) + \sum_{\substack{\pi_{rc}, \pi_{rc'} \in \Pi_r \\ \pi_{rc} \neq \pi_{rc'}}} \text{grade}(x_i, \pi_{rc}) \cdot \text{grade}(x_j, \pi_{rc'}) \\ &= c_{rij} + \nabla c_{rij} \end{aligned}$$

Given that the grade of membership is by definition non-negative, all pairwise products of grades will also be non-negative—and, being sums of pairwise products, both c_{rij} and $\bigtriangledown c_{rij}$ will at their turn be non-negative too: $0 \leq c_{rij}, \bigtriangledown c_{rij}$.

Finally, given that c_{rij} and $\bigtriangledown c_{rij}$ are two non-negative terms adding up to 1, it is clear that neither of them can exceed this value: $c_{rij}, \bigtriangledown c_{rij} \leq 1$. Hence, as we wanted to prove, $0 \leq c_{rij} \leq 1$. □

Rather than considering a single application of one clustering function $f_r \in F$ on $\mathcal{X}$, we will mainly be concerned with aggregating the results over a number R of repetitions of the process. In this context, we can define:

Definition 9 (Average co-occurrence vector) The sequence of **average co-occurrence vectors** for object x_i is $(\mathbf{c}^\star_{1i} \ldots)$, where each component of $\mathbf{c}^\star_{Ri} = [c^\star_{Ri1} \ldots c^\star_{Rin}]^T$ is

$$c^\star_{Rij} = \frac{1}{R} \sum_{r=1}^{R} c_{rij} \tag{4}$$

Definition 10 (Average score) The sequence of **average scores** of object x_i is $(s^\star_{1i}, s^\star_{2i} \ldots)$, where each $s^\star_{Ri}$ is

$$s^\star_{Ri} = \frac{1}{R} \sum_{r=1}^{R} s_{ri} \tag{5}$$

Remark 3 Using average co-occurrence vectors, the average score of object x_i can be expressed as

$$s^\star_{Ri} = \frac{1}{R} \sum_{r=1}^{R} s_{ri} = \frac{1}{R} \sum_{r=1}^{R} \sum_{x_j \in \mathcal{X}} c_{rij} = \sum_{x_j \in \mathcal{X}} \frac{1}{R} \sum_{r=1}^{R} c_{rij} = \sum_{x_j \in \mathcal{X}} c^\star_{Rij}$$

It is interesting to note that

Proposition 2 *The s_{ri} are linear transformations of $\mathbf{c}_{ri}$, and the $s^\star_{Ri}$ are linear transformations of $\mathbf{c}^\star_{Ri}$.*

Proof Using an all-ones vector,

$$s_{ri} = \mathbf{1}^T \cdot \mathbf{c}_{ri} = [1 \quad 1 \quad \cdots \quad 1] \cdot [c_{ri1} \quad c_{ri2} \quad \cdots \quad c_{rin}]^T = \sum_{x_j \in \mathcal{X}} c_{rij}$$

$$s^\star_{Ri} = \mathbf{1}^T \cdot \mathbf{c}^\star_{Ri} = [1 \quad 1 \quad \cdots \quad 1] \cdot [c^\star_{Ri1} \quad c^\star_{Ri2} \quad \cdots \quad c^\star_{Rin}]^T = \sum_{x_j \in \mathcal{X}} c^\star_{Rij}$$

□

3.3 Dataset-conditioned distribution

The dataset $\mathcal{X}$ and clustering function f_r uniquely determine the values for the co-occurrence vectors $\mathbf{c}_{ri}$, and hence for all other values considered in the previous section. However, as the selection of f_r is not deterministic, the c_{rij} can be regarded as random

variables, and their conditional distribution across clustering functions, given a certain dataset $\mathcal{X}$, can be considered.

As the selection of each f_r is independent from the others, the values of the c_{rij} for different r will also be. The $\mathbf{c}_{ri}$ for different r will hence be independent and identically distributed random vectors, with a common expectation vector $\boldsymbol{\mu}_i$ and covariance matrix Σ_i. We will refer to each element, μ_{ij}, of $\boldsymbol{\mu}_i$ as the *affinity* of x_i and x_j.

Definition 11 (Object affinity) The **affinity** of objects x_i and x_j is the conditional expectation of c_{rij} given $\mathcal{X}$,

$$\mu_{ij} = E[c_{rij} \mid \mathcal{X}] \tag{6}$$

Remark 4 Being the expectations of the c_{rij}, with $c_{rij} \in [0, 1]$, the affinities μ_{ij} will also fall in the [0, 1] interval.

We can additionally define

Definition 12 (Object expected score) The **expected score** of object x_i is the conditional expectation of s_{ri} given $\mathcal{X}$,

$$\mu_i = E[s_{ri} \mid \mathcal{X}] \tag{7}$$

It is then easy to successively prove that

Proposition 3 *The value of the expected score μ_i of object x_i is*

$$\mu_i = E[s_{ri} \mid \mathcal{X}] = \sum_{x_j \in \mathcal{X}} \mu_{ij} \tag{8}$$

Proof As s_{ri} is the sum of the c_{rij}, its conditional expectation is

$$\mu_i = E[s_{ri} \mid \mathcal{X}] = E\left[\sum_{x_j \in \mathcal{X}} c_{rij} \mid \mathcal{X}\right] = \sum_{x_j \in \mathcal{X}} E[c_{rij} \mid \mathcal{X}] = \sum_{x_j \in \mathcal{X}} \mu_{ij}$$

□

Remark 5 Being the sum of $n = |\mathcal{X}|$ terms within the interval [0, 1], the value of μ_i will fall in the interval $[0, n]$. In order to make scores across differently-sized datasets comparable, we will also consider a **normalized expected score** $\bar{\mu}_i$, defined as $\bar{\mu}_i = \mu_i / n$.

Proposition 4 *As the number of repetitions R increases, the conditional distributions of the average co-occurrence vectors $\mathbf{c}^{\star}_{Ri}$ approach a multivariate Gaussian distribution with expectation $\boldsymbol{\mu}_i$ and covariance matrix Σ_i / R.*

Proof As the c_{rij} are independent and identically distributed for different r, by the multivariate central limit theorem we know that the sequence

$$\sqrt{R}\left(\frac{1}{R}\sum_{r=1}^{R} \mathbf{c}_{ri} - \boldsymbol{\mu}_i\right) = \sqrt{R}\left(\mathbf{c}^{\star}_{Ri} - \boldsymbol{\mu}_i\right)$$

converges in distribution to a multivariate Gaussian distribution with expectation $\mathbf{0}$ and covariance matrix Σ_i. Hence, for large enough R,

$$\begin{aligned}\sqrt{R}\left(\mathbf{c}^{\star}_{Ri} - \boldsymbol{\mu}_i\right) &\approx \mathcal{N}(\mathbf{0}, \Sigma_i)\\ \mathbf{c}^{\star}_{Ri} - \boldsymbol{\mu}_i &\approx \mathcal{N}(\mathbf{0}, \Sigma_i/R)\\ \mathbf{c}^{\star}_{Ri} &\approx \mathcal{N}(\mu_i, \Sigma_i/R)\end{aligned}$$

□

Proposition 5 *As the number of repetitions R increases, the conditional distributions of the average scores $s^{\star}_{Ri}$ approach a Gaussian distribution with expectation μ_i.*

Proof Being linear transformations of random vectors $\mathbf{c}^{\star}_{Ri}$ approaching a multivariate Gaussian distribution, the $s^{\star}_{Ri}$ also approach a Gaussian distribution

$$s^{\star}_{Ri} = \mathbf{1}^T \cdot \mathbf{c}^{\star}_{Ri} \approx \mathcal{N}\left(\mathbf{1}^T \cdot \boldsymbol{\mu}_i, \left(\Sigma^{\star}_{Ri}\right)^2\right)$$

with a certain variance $(\Sigma^{\star}_{Ri})^2$. The conditional expectation of these variables hence converges to

$$\lim_{R\to\infty} E\left[s^{\star}_{Ri} \mid \mathcal{X}\right] = \mathbf{1}^T \cdot \boldsymbol{\mu}_i = \sum_{x_j \in \mathcal{X}} \mu_{ij} = \mu_i$$

□

3.4 Sampling distribution

We can now proceed to consider the distribution of the scores across multiple samplings of the dataset $\mathcal{X}$. In particular, we will first focus on the distribution of the affinity μ_{ij} between objects x_i and x_j, conditioned to their being respectively generated by a certain pair of sources ψ_s and ψ_t. We shall name this measure the *affinity* of the two sources, ζ_{st}.

Definition 13 (Source affinity) The **affinity** of sources ψ_s and ψ_t is the conditional expectation of the object affinity μ_{ij}, given that $y_i = \psi_s$ and $y_j = \psi_t$, across all datasets $\mathcal{X}$ sampled from Ψ:

$$\zeta_{st} = E[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t]$$

A particular case of affinity is that of $\psi_t = \psi_s$, which we shall name the *self-affinity* ζ_{ss} of source ψ_s.

We can now also consider the conditional expectation of the normalized expected scores $\bar{\mu}_i$ for objects from source ψ_s.

Definition 14 (Source normalized expected score) The **normalized expected score** of a source ψ_s is the conditional expectation of the normalized expected score $\bar{\mu}_i$ of objects x_i generated by ψ_s, across all datasets $\mathcal{X}$ sampled from Ψ:

$$\zeta_s = E[\bar{\mu}_i \mid y_i = \psi_s]$$

This newly defined score satisfies that:

Proposition 6 *The value of the normalized expected score ζ_s for a source ψ_s is*

$$\zeta_s = \sum_{\psi_t \in \Psi} \alpha_t \cdot \zeta_{st}$$

Proof The value of $\bar{\mu}_i$ is

$$\bar{\mu}_i = \frac{1}{n}\mu_i = \frac{1}{n}\sum_{x_j \in \mathcal{X}} \mu_{ij}$$

The conditional expectation of $\bar{\mu}_i$ across samplings of $\mathcal{X}$ for which $|\mathcal{X}| = n$ can then be found as

$$\begin{aligned} E\big[\bar{\mu}_i \mid y_i = \psi_s, |\mathcal{X}| = n\big] &= E\left[\frac{1}{n}\sum_{x_j \in \mathcal{X}} \mu_{ij} \mid y_i = \psi_s, |\mathcal{X}| = n\right] \\ &= \frac{1}{n} E\left[\sum_{x_j \in \mathcal{X}} \mu_{ij} \mid y_i = \psi_s, |\mathcal{X}| = n\right] \end{aligned}$$

Assuming the $x_j \in \mathcal{X}$ are independent and identically distributed, and using the law of total expectation, this can be expressed as

$$\begin{aligned} E\big[\bar{\mu}_i \mid y_i = \psi_s, |\mathcal{X}| = n\big] &= \frac{1}{n}\sum_{x_j \in \mathcal{X}} E\big[\mu_{ij} \mid y_i = \psi_s, |\mathcal{X}| = n\big] \\ &= \frac{1}{n}\sum_{x_j \in \mathcal{X}} \sum_{\psi_t \in \Psi} P(y_j = \psi_t) \cdot E\big[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n\big] \\ &= \frac{1}{n}\sum_{x_j \in \mathcal{X}} \sum_{\psi_t \in \Psi} \alpha_t \cdot E\big[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n\big] \\ &= \frac{1}{n}\sum_{\psi_t \in \Psi} \alpha_t \cdot E\big[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n\big] \cdot \sum_{x_j \in \mathcal{X}} 1 \\ &= \frac{1}{n}\sum_{\psi_t \in \Psi} \alpha_t \cdot E\big[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n\big] \cdot n \\ &= \sum_{\psi_t \in \Psi} \alpha_t \cdot E\big[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n\big] \end{aligned}$$

Finally, assuming independence of normalized expected scores and source affinities with respect to dataset size n, and plugging the definition of the latter into the above formula, we obtain the desired result:

$$\begin{aligned} E\big[\bar{\mu}_i \mid y_i = \psi_s, |\mathcal{X}| = n\big] &= \sum_{\psi_t \in \Psi} \alpha_t \cdot E\big[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n\big] \\ \zeta_s = E[\bar{\mu}_i \mid y_i = \psi_s] &= \sum_{\psi_t \in \Psi} \alpha_t \cdot E[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t] = \sum_{\psi_t \in \Psi} \alpha_t \cdot \zeta_{st} \end{aligned}$$

□

3.5 Consistent clustering

We will now impose some conditions on the used clustering families, with respect to how they preserve the density and locality of the sources in Ψ. We will start by considering the *detectability* of a source by a clustering family:

Definition 15 (Source detectability) Given a set of sources Ψ and a clustering family F, a foreground source $\psi_s \in \Psi^+$ is **detectable by** F if its normalized expected score ζ_s is larger than that ζ_1 of the background source ψ_1.

Proposition 7 (Detectability criterion) *Given a set of sources Ψ and a clustering family F, a foreground source $\psi_s \in \Psi^+$ is detectable by F if*:

$$\alpha_s \cdot (\zeta_{ss} - \zeta_{1s}) > \alpha_1 \cdot (\zeta_{11} - \zeta_{s1}) + \sum_{\substack{\psi_t \in \Psi^+ \\ \psi_t \neq \psi_s}} \alpha_t \cdot (\zeta_{1t} - \zeta_{st})$$

Proof From the definition of detectability and Proposition 6,

$$\begin{aligned}
\zeta_s &> \zeta_1 \\
\sum_{\psi_t \in \Psi} \alpha_t \cdot \zeta_{st} &> \sum_{\psi_t \in \Psi} \alpha_t \cdot \zeta_{1t} \\
\alpha_s \cdot \zeta_{ss} + \alpha_1 \cdot \zeta_{s1} + \sum_{\substack{\psi_t \in \Psi^+ \\ \psi_t \neq \psi_s}} \alpha_t \cdot \zeta_{st} &> \alpha_s \cdot \zeta_{1s} + \alpha_1 \cdot \zeta_{11} + \sum_{\substack{\psi_t \in \Psi^+ \\ \psi_t \neq \psi_s}} \alpha_t \cdot \zeta_{1t} \\
\alpha_s \cdot (\zeta_{ss} - \zeta_{1s}) &> \alpha_1 \cdot (\zeta_{11} - \zeta_{s1}) + \sum_{\substack{\psi_t \in \Psi^+ \\ \psi_t \neq \psi_s}} \alpha_t \cdot (\zeta_{1t} - \zeta_{st})
\end{aligned}$$

□

Remark 6 This arrangement of the terms in the difference $\zeta_s - \zeta_1$ is intended to capture the degree to which the clustering family captures the *density* and *locality* properties of the data in the minority clustering setting:

- For *dense* sources, self-affinity should be much larger than affinity to the background source. Therefore, the value of the left-side term should be large.
- For *local* sources, affinity to the background source and to other foreground sources should not be much different than their affinity to the background source itself. Therefore, the value of the right-side term should be small.

If a clustering family captures the density and locality of all foreground sources in a set, all of them will be detectable. In this case, the family is said to be consistent with the source set:

Definition 16 (Clustering family consistency) Given a set of sources Ψ, a clustering family F is **consistent with** Ψ if and only if all foreground sources $\psi_s \in \Psi^+$ are detectable by F.

The importance of detectable sources and consistent families lies in the fact that:

Theorem 1 *Given a dataset $\mathcal{X}$ sampled from a set of sources Ψ and a consistent clustering family F, for a sufficiently large number of repetitions R, the expected value of the average score s^*_{Ri} of objects x_i generated by a foreground source $\psi_s \in \Psi^+$ is larger than the expected value of the average scores s^*_{Rj} of objects x_j generated by the background source ψ_1.*

Proof Using $n = |\mathcal{X}|$, replacing the definitions of the different used quantities, and applying properties of the expectation, we know that, if ψ_s is detectable,

$$\begin{aligned} \zeta_s &> \zeta_1 \\ n \cdot \zeta_s &> n \cdot \zeta_1 \\ n \cdot E[\bar{\mu}_i \mid y_i = \psi_s] &> n \cdot E[\bar{\mu}_j \mid y_j = \psi_1] \end{aligned}$$

Assuming independence on the size of the dataset $\mathcal{X}$,

$$\begin{aligned} n \cdot E\big[\bar{\mu}_i \mid y_i = \psi_s, \big|\mathcal{X}'\big| = n\big] &> n \cdot E\big[\bar{\mu}_j \mid y_j = \psi_1, \big|\mathcal{X}'\big| = n\big] \\ n \cdot E\big[\mu_i/n \mid y_i = \psi_s, \big|\mathcal{X}'\big| = n\big] &> n \cdot E\big[\mu_j/n \mid y_j = \psi_1, \big|\mathcal{X}'\big| = n\big] \\ n \cdot E\big[E\big[s^\star_{Ri} \mid y_i = \psi_s, \mathcal{X}', \big|\mathcal{X}'\big| = n\big]\big]/n &> n \cdot E\big[E\big[s^\star_{Rj} \mid y_j = \psi_1, \mathcal{X}', \big|\mathcal{X}'\big| = n\big]\big]/n \\ E\big[s^\star_{Ri} \mid y_i = \psi_s, \mathcal{X}', \big|\mathcal{X}'\big| = n\big] &> E\big[s^\star_{Rj} \mid y_j = \psi_1, \mathcal{X}', \big|\mathcal{X}'\big| = n\big] \end{aligned}$$

which, assuming independence again, leads to

$$E\big[s^\star_{Ri} \mid y_i = \psi_s, \mathcal{X}\big] > E\big[s^\star_{Rj} \mid y_j = \psi_1, \mathcal{X}\big] \qquad \square$$

3.6 Algorithm

A corollary of this last Theorem 1 is

Corollary 1 *Given a dataset $\mathcal{X}$ sampled from a set of sources Ψ, and using a clustering family F which is consistent with Ψ, we can devise an algorithmic procedure to obtain a minority clustering of $\mathcal{X}$.*

Proof Given a dataset $\mathcal{X}$, we can apply a sequence of clustering functions f_r, drawn from F, and find the average score $s^\star_{Ri}$ for each object $x_i \in \mathcal{X}$. The expected value of the average scores of the background objects will be lower than that of the foreground ones. If a suitable threshold value is determined, we will be able to discriminate most foreground and background objects according to their score. $\square$

Remark 7 A *single* threshold suffices to separate background and foreground objects because Theorem 1 ensures the scores of the former will be lower than those of objects coming from *any* of the foreground sources.

Remark 8 It is important to note that, whereas this procedure will allow us to separate foreground and background objects, it will not find the different clusters formed by the foreground ones. A regular ensemble clustering algorithm, such as those of Ghosh et al. (2002) or Topchy et al. (2005), can be applied on the objects that have been deemed to belong to the foreground for that goal. We will hence focus on the foreground/background separation problem for the rest of the paper.

The resulting algorithm, which we have named Ensemble Weak minOrity Cluster Scoring (EWOCS), is described in Algorithm 1.

The first step of EWOCS is the initialization of an auxiliary array, which will contain the accumulated scores s_i^+ of all objects, to zero (line 1). The main loop is then entered

Algorithm 1 Ensemble Weak minOrity Cluster Scoring (EWOCS)

Input: A dataset $\mathcal{X}$
Input: A consistent clustering family F
Input: An ensemble size R
Output: A hard minority clustering Π of $\mathcal{X}$

1: Initialize the accumulated scores of all objects x_i to zero,

$$s_i^+ = 0$$

2: **For** $r = 1$ **to** R **do**
3: Draw a clustering function f_r at random from F,

$$f_r \in F$$

4: Apply f_r to obtain clustering Π_r,

$$\Pi_r = f_r(\mathcal{X})$$

5: Find cluster sizes,

$$\text{size}(\pi_{rc}) = \sum_{x_i \in \mathcal{X}} \text{grade}(x_i, \pi_{rc})$$

6: Update the accumulated scores of each object,

$$s_i^+ \leftarrow s_i^+ + s_{ri} = s_i^+ + \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{size}(\pi_{rc})$$

7: Find the final average scores of each object,

$$s_{Ri}^\star = \frac{s_i^+}{R}$$

8: Determine a threshold $s_{th}^\star$ separating the scores,

$$s_{th}^\star = \textbf{find\_threshold}(s_{R1}^\star \ldots s_{Rn}^\star)$$

9: Create the foreground and background clusters, π_f and π_b,

$$\pi_f = \left\{x_i \mid s_{Ri}^\star \geq s_{th}^\star\right\}$$
$$\pi_b = \left\{x_i \mid s_{Ri}^\star < s_{th}^\star\right\}$$

10: **Return** The minority clustering $\Pi = \{\pi_b, \pi_f\}$

(lines 2–6). The number of iterations of this loop, R, determines the ensemble size and is a user-supplied parameter. Larger values of R will yield better results, but at the expense of a larger computational cost.

At each iteration, a clustering function f_r is drawn at random from family F (line 3) and is then applied to dataset $\mathcal{X}$ to obtain a clustering Π_r (line 4). The size of each cluster π_{rc} in

clustering Π_r is then found (line 5), and then the score of each object, as defined in Eq. (2), is found and added to the accumulated score s_i^+ (line 6).

When the main loop is over, the final average score of each object, $s^\star_{Ri}$ is found from the final accumulated score s_i^+ and the ensemble size R (line 7). From the distribution of these scores $s^\star_{Ri}$, a threshold value $s^\star_{th}$ which separates the scores of the foreground and the background objects is inferred (line 8). At this point, the only steps that remain are separating the objects according to their scores into a foreground and a background cluster (line 9) and returning the resulting clustering (line 10).

The time complexity of the algorithm prior to the determination of the threshold is dominated by the R repetitions of the main loop. Inside it, if the number of clusters in the clusterings produced by the functions $f_r \in F$ is bounded and not dependent on the size of the dataset $\mathcal{X}$, the cost of each iteration is in the order of $O(|\mathcal{X}|)$. In order to keep an overall complexity of $O(R \cdot |\mathcal{X}|)$, linear with respect to the number of repetitions and the size of the dataset, it is thus necessary to use clustering families and threshold determination methods whose complexity is also a linear function of this size $|\mathcal{X}|$.

The obtained EWOCS algorithm has a number of components which allow different implementations: neither the consistent clustering function family F (line 3) nor the method for the determination of the threshold score separating foreground and background objects (line 8) are specified. As mentioned in the introduction, the following two sections, Sects. 4 and 5, give insights into each one of these two issues, respectively.

3.7 Clustering model

Some algorithms are only devised to build a clustering of an input dataset, and do not provide any device to determine the hypothetical assignments of new objects to one of the obtained clusters. This is the case, for instance, of most hierarchical (including HAC) and ensemble clustering (e.g., Ghosh et al. 2002; Gionis et al. 2005) algorithms. However, most popular partitional methods—starting with k- and c-means, and continuing with all probabilistic mixture algorithms—provide, as a byproduct of the clustering process, a *clustering model* which may then be later used as a classification model for new data, after identifying the obtained clusters with classes.

In the case of EWOCS, if the functions in the used family F provide models together with the clusterings when applied to dataset $\mathcal{X}$, these individual models can be extended to obtain an aggregated minority clustering model.

More specifically, if the application of $f_r \in F$ to $\mathcal{X}$ produces clustering Π_r and clustering model $\mathcal{M}_r$, after Algorithm 1, an EWOCS minority clustering model $\mathcal{M}^E$ can be constructed, containing:

- the inner clustering models $\mathcal{M}_r$,
- the size of each cluster π_{rc} in the clusterings Π_r,
- and the threshold value $s^\star_{th}$ which separates foreground and background objects.

The process of classifying a new object x_x using the obtained model $\mathcal{M}^E$ is described in Algorithm 2. It follows the main steps of the previous Algorithm 1, but replacing the application of new clustering functions $f_r \in F$, by that of the previously obtained clustering models $\mathcal{M}_r$ (line 3). After all models have been applied, the average score of the object is found (line 5), and the object is deemed to belong to the foreground or background cluster according to whether its score exceeds the previously found threshold (line 6).

Algorithm 2 Classification using an EWOCS clustering model

Input: An EWOCS minority clustering model $\mathcal{M}^E = (\{\mathcal{M}_r\}, \{\text{size}(\pi_{rc})\}, s^\star_{th})$
Input: An object x_x
Output: The cluster $\pi_x \in \{\pi_b, \pi_f\}$ to which x_x would belong

1: Initialize the accumulated score of the object x_x to zero,

$$s^+_x = 0$$

2: **For** $r = 1$ **to** R **do**
3: Apply the clustering model $\mathcal{M}_r$ to obtain the grade of membership of x_x to each π_{rc}

$$\big(\text{grade}(x_x, \pi_{r1}) \ldots \text{grade}(x_x, \pi_{rk_r})\big) = \mathcal{M}_r(x_x)$$

4: Update the accumulated score of the object

$$s^+_x \leftarrow s^+_x + s_{rx} = s^+_x + \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_x, \pi_{rc}) \cdot \text{size}(\pi_{rc})$$

5: Find the final average score of the object

$$s^\star_{Rx} = \frac{s^+_x}{R}$$

6: Assign the object to the foreground or background cluster, π_f or π_b, according to the relation between its average score and the separating threshold

$$\pi_x = \begin{cases} \pi_f & \text{if } s^\star_{Rx} \geq s^\star_{th} \\ \pi_b & \text{if } s^\star_{Rx} < s^\star_{th} \end{cases}$$

7: **Return** The object cluster π_x

4 Weak clustering

As stated in Sect. 3.5, the theoretical properties of the EWOCS algorithm depend only on the condition of the used clustering family being consistent. We believe that the requirements for being consistent, according to Definition 16, should be fairly loose—and that, hence, the EWOCS algorithm is suitable for use with weak clustering algorithms.

In this context, a clustering function family F is a clustering algorithm which includes elements of randomness. Each sequence of random values will determine a member function of the family. From a conceptual point of view, drawing a function f_r from the family F will hence correspond to drawing a sequence of random values to be later used by the algorithm. From a computational one, it can correspond, for instance, to choosing a seed value for the algorithm's internal random number generator.

The two weak clustering algorithms that are used in the work of Topchy et al. (2003) are based on either splitting the dataset using random hyperplanes, or on clustering projections of the data on random subspaces. We found the first of them particularly convenient for our purposes, and extended it. Sect. 4.1 reviews this our extension of the random splitting algorithm.

However, even if these methods have been proved to produce clusterings useful for combination within an ensemble, they both perform linear mappings of the data and, hence, are based on the notion of linear separation. Although non-linearly separable clusters can be successfully identified by linear separators, non-linear weak separators have not been thoroughly explored. Besides, linear methods depend on the data being expressible as feature vectors, and hence cannot directly deal with structured objects such as sequences or trees.

Our proposal in this direction is a new weak clustering algorithm based on Bregman divergences, which allows non-linear splitting boundaries and, through the use of kernels, can deal with structured data. This proposed Random Bregman Clustering is described in Sect. 4.2.

Later, Sect. 6.4.2 will provide an estimation of the consistency of the proposed clustering families over a number of datasets. The results shall provide an empirical assessment of the suitability of these two families for use within EWOCS.

4.1 Random splitting

The random splitting algorithm presented in Topchy et al. (2003) performs only binary bisections of the objects in the dataset. Our Random Splitting algorithm (RSPLIT) is a generalization of this algorithm, which allows an arbitrary number of clusters k.

For this algorithm we require the objects in dataset $\mathcal{X}$ to be expressible as z-dimensional real vectors (i.e., $\mathcal{X} \subset \mathbb{R}^z$). To account for multiple clusters, we have adopted the same representation of hyperplanes as in the Multi-Class Support Vector Machines of Crammer and Singer (2001): each splitting hyperplane is defined by a weight vector $\omega_c = (\omega_{c1} \ \ldots \ \omega_{cz})$ and an offset δ_c, and objects belong to the cluster (class in the original formulation) from whose hyperplane they are separated by the largest margin.

Similarly to Topchy et al., in a clustering ensemble setting, the number of clusters k does not need to be given a priori, but is rather drawn at random between 2 and a user-supplied value k_{max}.

This idea leads to the simple procedure described in Algorithm 3. The algorithm takes three sequential steps. The first of them is the selection of the effective number of clusters

Algorithm 3 Random Splitting (RSPLIT)

Input: A dataset $\mathcal{X}$
Input: A maximum number of clusters k_{max}
Output: A hard all-in clustering Π of $\mathcal{X}$

1: Draw a number of clusters k at random from the range $\{2 \ldots k_{max}\}$

$$k \in \{2 \ldots k_{max}\}$$

2: Generate a weight vector $\omega_c = [\omega_{c1} \ldots \omega_{cz}]$ and an offset δ_c at random for each $c \in \{1 \ldots k\}$

$$\omega_{c1} \ldots \omega_{cz}, \quad \delta_c \in [-1 \ldots 1]$$

3: Assign each object x_i to the cluster π_c whose hyperplane gives the largest margin

$$\pi_c = \left\{ x_i \in \mathcal{X} \mid \arg\max_q \omega_q \cdot x_i + \delta_q = c \right\}$$

4: **Return** The clustering $\Pi = \{\pi_1 \ldots \pi_k\}$

k (line 1). Any discrete distribution between 2 and k_{max}, such as the uniform distribution, can be used. For each cluster π_c, random weights ω_c and offsets δ_c (line 2) are then generated. Again, we have stuck to the uniform distribution from all the possible continuous distributions within the $[-1 \ldots 1]$ range.

Once these values are generated, the margin of each object x_i with respect to the hyperplanes is found as the dot product between the object x_i and the hyperplane's weight vector ω_c, shifted by the latter's offset δ_c. Each object is assigned to the cluster induced by the hyperplane to which its margin is maximal (line 3). The resulting clustering can then be returned (line 4).

The time complexity of this algorithm is dominated by the calculation of the margin in step (line 3), and is hence in the order of $O(k_{max} \cdot z \cdot |\mathcal{X}|)$.

The algorithm requires as input the maximum number of clusters k_{max} in each split. A part of Sect. 6.4.3 is devoted to the empirical study of the sensitivity of the results of EWOCS to its value.

We will henceforth refer to this algorithm as RSPLIT, and to its application within EWOCS as EW-RSPLIT.

4.2 Random Bregman clustering

As stated in the introduction to Sect. 4, two desirable properties of weak clustering algorithms, but to which few attention has been devoted so far, are, first, the ability to find non-linear boundaries in vectorial data, and, second, the possibility to deal with non-vectorial and/or structured data. Kernel methods have a long story of successes across a wide spectrum of machine learning tasks (Shawe-Taylor and Cristianini 2004) and, specifically, they are known for their capability to address both of these issues. The use of kernel functions allows to separate non-linearly separable classes, even with linear methods (Freund and Schapire 1999); and kernels have been devised and successfully applied for non-vectorial objects such as word sequences (Cancedda et al. 2003) or parse trees (Collins and Duffy 2002).

Kernel functions induce a distance metric between objects. Any kernel function K_ϕ is equivalent to an inner product in a high-dimensional space, onto which there will exist a certain mapping ϕ. Hence, if $\phi(x)$ and $\phi(y)$ are, respectively, the images of two objects x and y in this space, $K_\phi(x, y) = \phi(x) \cdot \phi(y)$. Their squared Euclidean distance on the mapped space, $D_\phi(x, y)$, can then be found as:

$$\begin{aligned} D_\phi(x, y) &= \big\|\phi(x) - \phi(y)\big\|^2 \\ &= \big(\phi(x) - \phi(y)\big) \cdot \big(\phi(x) - \phi(y)\big) \\ &= \phi(x) \cdot \phi(x) + \phi(y) \cdot \phi(y) - 2 \cdot \phi(x) \cdot \phi(y) \\ &= K_\phi(x, x) + K_\phi(y, y) - 2K_\phi(x, y) \end{aligned} \tag{9}$$

This transformation is the basis for existing kernel-based all-in clustering algorithms, such as kernel k-means (Girolami 2002). In our case, given that these squared Euclidean distances will be, by construction, Bregman divergences, we can join Mercer kernel theory and that of Bregman clustering and devise a weak all-in clustering procedure. The idea is to randomly select a number of objects which can act as *seeds* for the clustering, and then define clusters according to the divergence from these seeds of the objects in the dataset. The resulting Random Bregman Clustering (RBC) method is described in Algorithm 4.

Algorithm 4 Random Bregman Clustering (RBC)

Input: A dataset $\mathcal{X}$
Input: A Bregman divergence D
Input: A maximum number of clusters k_{max}
Output: A (hard or soft) all-in clustering Π of $\mathcal{X}$

1: Draw a number of clusters k at random from the range $\{2 \dots k_{max}\}$

$$k \in \{2 \dots k_{max}\}$$

2: Select a subset $\hat{\mathcal{X}}$ of k seeds from $\mathcal{X}$

$$\hat{\mathcal{X}} = \{\hat{x}_1 \dots \hat{x}_k\} \subset \mathcal{X}$$

3: **If** hard clustering desired **then**
4: Assign each object x_i to the cluster π_c induced by its nearest seed $\hat{x}_c$,

$$\pi_c = \left\{ x_i \in \mathcal{X} \mid \arg\min_{\hat{x}_q \in \hat{\mathcal{X}}} D(\hat{x}_q, x_i) = \hat{x}_c \right\}$$

5: **Else**
6: Find membership grade for each object x_i and cluster π_c,

$$\text{grade}(x_i, \pi_c) = \frac{e^{-D(\hat{x}_c, x_i)}}{\sum_{q=1}^{k} e^{-D(\hat{x}_q, x_i)}}$$

7: **Return** The clustering $\Pi = \{\pi_1 \dots \pi_k\}$

RBC is thus a seed-based algorithm. Given dataset $\mathcal{X}$, a Bregman divergence D and a maximum number of clusters k_{max}, the first step of RBC is selecting the effective number of clusters in the clustering, k (line 1). Any discrete distribution between 2 and k_{max}, such as the uniform distribution, can be used. A subset $\hat{\mathcal{X}}$ of size k is then selected at random from $\mathcal{X}$ (line 2). We shall name this subset the *seed subset*, and each one of their members will be a *seed*. Each seed will induce a cluster in the output clustering.

The output clustering is constructed following the theoretical framework provided by Bregman clustering (Banerjee et al. 2005). First, the distance of each object $x_i \in \mathcal{X}$ to the seeds $\hat{x}_c \in \hat{\mathcal{X}}$ is found. If a hard clustering is desired, each object is then assigned to the cluster induced by its nearest seed (line 4). If, instead, a soft clustering is desired, the grade of membership of each object to each cluster is proportional to the exponential of the negated divergence from the seed of the latter to the former (line 6). In both cases, the only remaining step is then returning the resulting (hard of soft) clustering (line 7).

The construction of the hard clustering is hence equivalent to a single assignment step of Bregman hard clustering; and that of the soft clustering is equivalent to a single expectation step of Bregman soft clustering, with a uniform *a priori* probability of membership to all clusters.

The time complexity of the RBC algorithm is dominated by the clustering construction step (line 4 or 6), and, as long as the kernel computation does not depend on the maximum number of clusters k_{max} or on the size of the dataset $|\mathcal{X}|$, it is in the order of $O(k_{max} \cdot |\mathcal{X}|)$. This is comparable to the cost of RSPLIT, so the increase in expressiveness of the algorithm

does not come at the expense of an increase in computational complexity. The algorithm hence remains inexpensive, and suitable for use in a weak clustering ensemble.

In addition to the particular divergence function used, the algorithm only takes as parameter the maximum number of clusters k_{max}, whose influence in EWOCS, as mentioned previously, will be considered in Sect. 6.4.3.

We will henceforth refer to the hard and soft versions of this algorithm as HRBC and SRBC, respectively, and to their application within EWOCS as EW-HRBC and EW-SRBC.

We have explored the use of the following families of Bregman divergences for two given objects x and y, at the core of the RBC algorithm:

Squared Euclidean Distance (EUC), widely used in a variety of domains because of its simplicity and good performance. It is simply:

$$D_E(x, y) = (x - y)^T (x - y) \tag{10}$$

Squared Mahalanobis Distance (MAH), which has specifically been reported to give the best results within previous approaches to minority clustering (Gupta and Ghosh 2005, 2006). It is a version of standard Euclidean distance normalized for a particular dataset:

$$D_M(x, y) = (x - y)^T \Sigma^{-1} (x - y) \tag{11}$$

where Σ is the covariance matrix of the considered dataset.

Gaussian-Kernel Distance (G(α,γ)), successfully applied in non-parametric (i.e., distribution-free) clustering algorithms, such as mean shift (Fukunaga and Hostetler 1975; Cheng 1995). The Gaussian kernel $K_\phi(x, y)$ between two objects x and y is defined as the exponential of the negated squared Euclidean distance between them, with two additional scaling parameters α and γ:

$$K_\phi(x, y) = \alpha \cdot e^{-\gamma \|x-y\|^2} \tag{12}$$

By Eq. (9), their induced squared Euclidean distance mapped space, $D_\phi(x, y)$, can be found as:

$$\begin{aligned} D_\phi(x, y) &= K_\phi(x, x) + K_\phi(y, y) - 2K_\phi(x, y) \\ &= \alpha + \alpha - 2\alpha \cdot e^{-\gamma \|x-y\|^2} \\ &= 2\alpha\left(1 - e^{-\gamma \|x-y\|^2}\right) \end{aligned} \tag{13}$$

Gaussian kernels locally map the Euclidean space around each point into a hypersphere of radius $\sqrt{2\alpha}$, and the rate at which neighbouring points are pushed apart towards the edge of the hypersphere increases with the value of parameter γ. If this Gaussian-kernel distance is used in RBC, small values of α lead to fuzzy boundaries between the clusters, whereas large values produce crisp ones. As a particular case, the limit of soft RBC as $\alpha \to \infty$ is equivalent to hard RBC using squared Euclidean distance.

4.3 Unsupervised tuning of Gaussian-kernel distance

The use of the presented Gaussian-kernel distance requires the choice of the values for α and γ, which model the degrees of fuzziness and locality of the output clustering, respectively. The determination of suitable values for α and γ can become a problematic issue, especially in unsupervised clustering settings.

Similar problems are to be addressed in all-in fuzzy clustering algorithms which depend on a parameter. The *degree of fuzziness* parameter, traditionally referred to as m, of the fuzzy c-means algorithm (Bezdek 1981) is probably the one whose tuning has received the most attention in the literature (Deer and Eklund 2003; Yu et al. 2004; Okeke and Karnieli 2006; Schwämmle and Jensen 2010).

In the approach of Schwämmle and Jensen (2010), the authors study the behaviour of the cluster centroids as the degree of fuzziness m increases, and find that, at a certain point, the clustering degrades and the clusters start collapsing on each other. This phenomenon can be detected by watching the minimum distance between centroids: the moment the degradation starts, the first two clusters collapse and this distance becomes close to zero. It is interesting to note that, according to the authors, this happens however many clusters are used, even if the number does not match the actual one.

Given that "*a large fuzzifier value suppresses outliers in data sets*", the authors consider that maximum fuzziness should be sought, and hence propose selecting the largest m value for which the minimum centroid distance still remains above a predefined threshold ϵ (set so as to reduce floating-point errors).

We have adapted the approach of Schwämmle and Jensen to determine the optimal values of α and γ for EW-SRBC. The method is particularly suitable to our needs: it does not depend on specific properties of FCM, nor requires knowledge of the exact number of clusters in the dataset. However, as the EW-SRBC method does not provide centroids for the found signal clusters, we have instead tuned the parameters with the SOFTBBC-EM algorithm of Gupta and Ghosh (2006). Given that the optimal divergence metric for clustering will be more dependant on the dataset than on the used algorithm, we believe that the parameters detected using SOFTBBC-EM will provide, at least, competitive performance when used within EW-SRBC.

For a given value of γ, the influence of α on the clustering is equivalent to that of m for FCM. When moving from $\alpha \to \infty$ to $\alpha \to 0$, the fuzziness of the clustering is increased from a completely crisp clustering to gradually fuzzier ones. At a certain point α_{th}, the clustering starts degrading, and each object is eventually assigned a uniform probability of belonging to any cluster.

On the flipside, for a given value of α, the influence of γ on the clustering gives rise to two turning points: for values larger than a certain γ_h, the distance between all pairs of objects tend to 2α; whereas for those smaller than a certain γ_l, they all tend to 0. Both phenomena degrade the clustering, and hence also lead to cluster collapse. However, there is an interaction between the values of α and γ: larger values of α force crisper decisions, and hence extend the feasible region for γ.

Hence, the (α, γ) plane will contain an approximately V-shaped curve on one of whose sides the value of the minimum centroid distance will fall below the floating-point-precision threshold ϵ. Following the criterion of Schwämmle and Jensen, we look for maximum fuzziness, and hence the algorithm should select the vertex of this curve. At this point, the value of α is the minimum one which still avoids degradation, and for it γ_h and γ_l have become equal.

We have empirically verified that such curves actually arise across a variety of datasets. For instance, Fig. 2 shows a contour plot of the minimum centroid distance of the clusterings obtained using SOFTBBC-EM on the TOY dataset. In it, the thicker curve denotes the contour level for a value of $\epsilon = 10^{-3}$, and the point at its vertex corresponds to the values of α and γ detected by the algorithm.

Given that the minimum centroid distance function has to be obtained by sampling, which introduces an amount of experimental noise, standard numerical methods for optimization

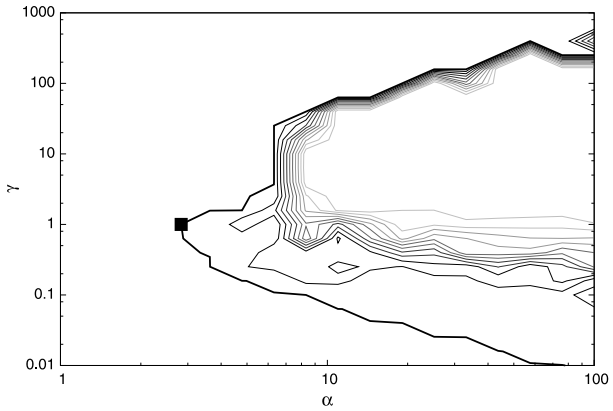


Fig. 2 Contour plot of minimum centroid distance (TOY data)

cannot be used, and minimization is instead performed using a recursive logarithmic grid search algorithm. This allows us to exponentially increase the precision in the detection of the optimal point, without an exponential increase of the computational burden.

We will henceforth refer to the distance induced by this automatically tuned Gaussian kernel as G(AUTO).

5 Threshold determination

The last step of the EWOCS algorithm is that of determining, from the sequence of scores $s_1^\star \dots s_n^\star$ found by the ensemble clustering process,[4] a threshold value $s_{th}^\star$ which separates foreground and background objects. We have considered the following procedures to perform this decision.

5.1 BEST

In BEST, the score for which the performance of the method is maximal according to a given measure is taken as threshold. From the metrics that we have used for our evaluation, we have chosen for our experiments the cutoff point to be the one that maximizes the F1 measure, which will be defined in Sect. 6.3. This criterion is informative as an upper bound of the performance of the other ones, and we have hence reported it for our experiments.

5.2 SIZE

Following other works in minority clustering (Gupta and Ghosh 2005, 2006; Ghosh and Gupta 2011), in SIZE the number of foreground objects is assumed to be known a priori. After sorting the objects by their score, it is this number of highest-scored objects that are taken to form the foreground cluster, whereas the rest are considered background objects. The score of the object in the cutoff point is taken as threshold.

However, the proposers of this criterion give no hints about how the number of foreground objects can be estimated, and we believe this limits its applicability for unsupervised

[4] For the sake of simplicity, we will be omitting in this section the R subindex from $s_{Ri}^\star$, as we believe there is no risk of confusion with other than the final scores.

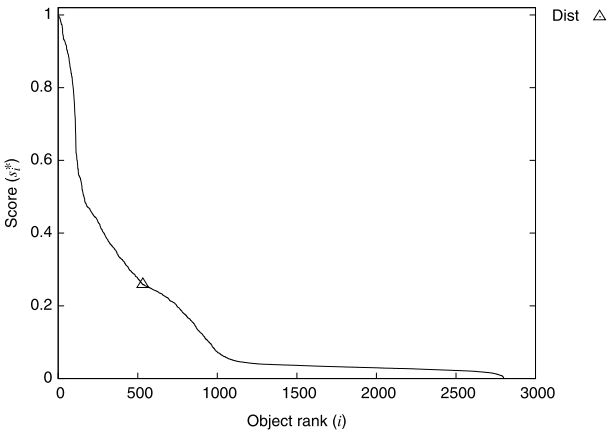


Fig. 3 Accumulated score distribution (EW-SRBC on TOY data)

minority clustering. We have nevertheless included it to allow a comparison to previous approaches which use it. For our experiments, we have assumed that the exact number of foreground objects is known, and used this value. Hence, the results for SIZE should also be regarded as an upper bound.

5.3 DIST

Following our previous work on relation detection (Gonzàlez and Turmo 2009), DIST arises from the observation of the distribution of the sorted sequence of scores of the clustered objects (see Fig. 3 for example). A small number of instances are assigned high scores whereas a large number are assigned low ones, presumably corresponding to foreground and background objects, respectively. The cutoff point should try to separate these two regions. Intuitively, this point will lie in the region of maximum convexity of the curve, and hence close to the lower left corner of the plot. An approximate but efficient way to determine the threshold is to minimize the distance from the origin in a normalized plot of the scores.

The first step in this criterion is hence sorting the objects $x_i \in \mathcal{X}$ by decreasing scores assigned to them by the EWOCS algorithm, so that, in the sequence $s_1^\star \ldots s_n^\star$, $\forall i : s_i^\star \geq s_{i+1}^\star$. These scores are then linearly mapped to the range $[0 \ldots 1]$, obtaining normalized versions $\bar{s}_i^\star$:

$$\bar{s}_i^\star = \frac{s_i^\star - \min s_j^\star}{\max s_j^\star - \min s_j^\star} \tag{14}$$

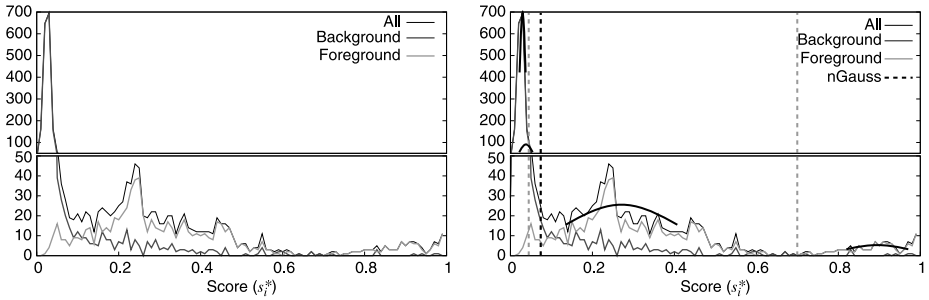
Then, the distance from the origin in the normalized plot is found for each object, and that at the minimum distance is selected as cutoff object x_{th}:

$$\mathbf{dist}(x_i) = \sqrt{(\bar{s}_i^\star)^2 + (i / \max i)^2} \tag{15}$$

$$x_{th} = \arg\min_{x_i \in \mathcal{X}} \mathbf{dist}(x_i) \tag{16}$$

5.4 NGAUSS

The theoretical analysis of the EWOCS method presented in Sect. 3 provides us a new approach to automatically determine the threshold score. In particular, we can much benefit



(a) Score histogram (b) Fitting and threshold detection (nGauss)

Fig. 4 Ew-SRbc on Toy data

from the result stated in Proposition 5: the conditional distributions of the average scores $s_i^\star$ approach a Gaussian distribution with expectation μ_i. If we assume that the value of μ_i depends mainly on the source ψ_s which produced x_i, we can try to approximate the overall distribution of average scores $s_i^\star$ by a mixture of Gaussian components, one for each one of the sources generating the dataset.

As an example, the histogram of scores generated by the same run of Ew-SRbc on the Toy data is shown in Fig. 4a. As well as the joint distribution of scores (labeled All), the separate histograms for objects from the foreground and background sources are also plotted. Two Gaussian peaks are easily identifiable around the scores of 0.05 and 0.25, and we could expect another minor Gaussian component to explain the probability mass around the score of 0.9.

The key to threshold selection is thus determining the number of mixtures, identifying them, and finding the boundaries between them. The cutoff points must lie at one of these boundaries. There is a wide spectrum of methods to solve this task, and among them we have chosen Expectation-Maximization (EM), being by far the most popular one. The determination of the number of mixtures reduces to discovering the number of clusters and hence to a model selection problem. Given that the score distribution will always be one-dimensional (for whichever dimension of the input dataset), and one-dimensional EM is fast, we have used the usual approach of running EM for increasing numbers of clusters and then using a model-selection criterion to select the best one (Fraley and Raftery 1998). More specifically, we have used the Bayesian Information Criterion (Schwartz 1978). In Fig. 4b, the arcs denote the mean, variance and a priori probabilities of the identified components.

Proposition 5 states than only the mixture with the lowest mean should contain the background objects. However, it is empirically observed that the selection criterion often chooses models which split this (and/or other) source into several components (this can be observed, for instance, in Fig. 4b). It is hence necessary to separate the found components into those corresponding to the background source and those from the foreground ones. More specifically, if k components $\hat{\psi}_1 \dots \hat{\psi}_k$ have been identified (sorted by increasing means $\hat{\mu}_1 > \cdots > \hat{\mu}_k$), for each $c \in \{1 \dots k-1\}$, the possibility that $\hat{\psi}_1 \dots \hat{\psi}_c$ contain background objects and $\hat{\psi}_{c+1} \dots \hat{\psi}_k$ contain foreground ones needs to be considered.

The set of cutoff point candidates is hence built from the boundary scores for each $c \in \{1 \dots k-1\}$, i.e., the scores $s_c^\star$ for which:[5]

$$p\left(s_c^\star \in \bigcup_{d=1}^{c} \hat{\psi}_d \mid s_c^\star\right) = p\left(s_c^\star \in \bigcup_{d=c+1}^{k} \hat{\psi}_d \mid s_c^\star\right)$$

Moreover, and as stated in Sect. 5.3, the small number of foreground instances are assigned high scores whereas the large number of background instances are assigned low scores. As a result, the variances of the scores of the former will differ significantly from those of the latter, being much larger.

This last fact provides us with a heuristic criterion to choose a single threshold score from the candidate set: being $\hat{\sigma}_1^2 \dots \hat{\sigma}_k^2$ the variances of the found components $\hat{\psi}_1 \dots \hat{\psi}_k$, we select the boundary score that maximizes the difference between the average component variances at both sides:

$$s_{th}^\star = \arg\max_{s_c^\star} \left| \frac{1}{c} \sum_{i=1}^{c} \hat{\sigma}_i^2 - \frac{1}{k-c} \sum_{i=c+1}^{k} \hat{\sigma}_i^2 \right| \tag{17}$$

We will refer to this criterion as nGauss+Var. As an upper bound of its performance, we will also consider a nGauss+Best criterion, which selects the boundary score $s_c^\star$ which maximizes the F1 measure. In Fig. 4b, the possible cutoff points are depicted by dashed vertical rules. The score selected as threshold by both nGauss+Best and nGauss+Var is emphasized in black.

Remark 9 It is important to note here that need not be a one-to-one correspondence between mixture components and sources (as mentioned, we have often found the scores from a single source to be split across several components in the chosen mixture model), so we do not expect the number components selected at this step to be the number of sources in the data. We have devised this procedure for threshold determination only, and cannot ascertain how well correlated the number of components and the number of sources will be.

A slightly different alternative to overcome the foreground and background component separation problem is that of simplifying the possible models and performing EM with only 2 clusters. In this case, there is no ambiguity in the choice of the background and foreground components, as there must be one of each. We have named this simplified Gaussian modeling approach 2Gauss.

Finally, as a last and implementation-related detail, we have found that using the linearly mapped scores $\bar{s}_i^\star$ as defined in Eq. (14) as input to the EM algorithm for model fitting, instead of the actual scores $s_i^\star$, reduces the floating point rounding error and improves the quality of the detected threshold.

6 Evaluation on synthetic data

In order to validate the proposed Ewocs algorithm and to assess the performance of Ewocs-based approaches, we have performed a series of experiments on synthetic data.

[5] If several such scores exist for a given c, we have taken the largest value for which, in addition, the probability of the foreground mixtures is increasing.

Table 1 Parameter range for synthetic dataset generation

Number of dimensions	2, 3, 5, 8
Data range	$[-2.0 \ldots +2.0]$
Number of background samples	5400 ... 12000
Number of foreground sources	3 ... 8
Number of foreground samples	700 ... 1800
Variance within foreground sources	0.125 ... 0.25
Minimum distance between foreground sources	0.75

In a preliminary stage, the consistency (in the sense of Definition 16) of the different used weak clustering algorithms has been empirically assessed. Later, a full-fledged comparison of the performance of EWOCS-based approaches to other methods in the state of the art has been carried out.

Next sections give details about the evaluation procedure. Sect. 6.1 describes the used datasets and Sect. 6.2 enumerates the different approaches to be evaluated or employed as reference. Next Sect. 6.3 describes the evaluation protocol, including the considered metrics, and, finally, Sect. 6.4 exposes and discusses the obtained results.

6.1 Data

The first dataset we have used for our experiments is the sample data plotted in Fig. 1. It is a simple 2-dimensional dataset in which five foreground sources, with different shapes and variances, are scattered against a background filled with a uniform distribution. Even though evaluation on a single dataset such as TOY scarcely possesses any statistical significance, "*for a 2-dimensional dataset, graphical verification is an intuitive and reliable validation of clustering*" (Ando 2007), and we believe this can be useful as an illustration of most of the concepts in our work.

For a more serious evaluation, we have prepared a number of synthetic datasets where foreground Gaussian sources are embedded within a set of uniformly distributed background objects. Several parameters, such as the number of sources, the number of foreground and background objects and the means and variances of the Gaussian sources, were chosen at random for each dataset. A summary of the ranges of these parameters can be found in Table 1. In total, 160 such datasets have been generated.[6] We will refer to this collection as SYNTH.

Additionally, in order to perform the preliminary experiments on method consistency, for each dataset in SYNTH, 9 additional samplings using the same source parameters were generated. The whole 10-dataset groups have been used for consistency estimation.

6.2 Approaches

We have implemented the EWOCS algorithm using each one of the weak clusterers proposed in Sect. 4.

EW-RSPLIT: EWOCS using the RSPLIT algorithm of Sect. 4.1.
EW-HRBC: EWOCS using the hard RBC algorithm, HRBC, of Sect. 4.2.
EW-SRBC: EWOCS using the soft RBC algorithm, SRBC, of Sect. 4.2.

[6] Available at http://www.lsi.upc.edu/~egonzalez/data/ml-synth.tar.gz.

The notation EW-RSPLIT/$R \times k$ (resp., EW-HRBC/$R \times k$ and EW-SRBC/$R \times k$) will be used to refer to the results obtained by EWOCS with an ensemble of R clusterings, each one produced by RSPLIT (resp., HRBC and SRBC) with $k_{max} = k$.

In order to assess the performance of EWOCS-based approaches with respect to the state of the art, we have implemented five existing methods for minority clustering:

BBOCC: as proposed by Gupta and Ghosh (2005). We have used the actual number of foreground objects as the *desired clustering size* parameter.

BBCPRESS: as proposed by Gupta and Ghosh (2006). Similarly to BBOCC, we have used the actual number of foreground objects as the *desired clustering size* parameter. The number of clusters, however, has been assumed to be given a priori, and by BBCPRESS/k we will refer to the runs of this algorithm with a number of clusters k.

DGRADE: as proposed by Ghosh and Gupta (2011). Again, the actual number of foreground objects has been used as *number of dense points*, to be classified into clusters. Among the three strategies sketched by the authors, we have implemented the only one not requiring the number of clusters k or a maximum *stability* parameter from the user. This strategy has been preferred, despite its greater computational cost, because of its much lower degree of supervision. Finally, following the original paper, the output of DGRADE has been refined using the BBC algorithm.

AUTOHDS: as proposed by Gupta et al. (2010).[7] The tuning of the *smoothing* and *particle threshold* parameters of the algorithm using the interactive approach proposed by the authors is not feasible in our case (for the SYNTH corpus, it would require the manual tuning of 160 sets of parameters). We have instead considered a setting in which a single set of parameter values is used across all datasets. Thus, by AUTOHDS/n_{eps}–n_{part} we will refer to the runs of this algorithm with a smoothing parameter n_{eps} and a particle threshold n_{part}.

kMD: as proposed by Ando (2007). The implementation tries to mimic to the maximum extent that of the original paper: we have used Gaussian distributions for the foreground clusters and a uniform distribution for the background. The clusters have been initialized by selecting fixed-size sets of most similar points to a randomly chosen one. To refer to the runs of this algorithm with a certain parameter tuning, we will use the notation kMD/$R \times s_0$–s_{min}, where R refers the number of cluster detection iterations, and s_0 and s_{min} refer to the *initial* and *required cluster size* parameters.

For the divergence-based approaches (i.e., all but kMD), MAH has been used as metric.

It is important to note that these methods, as well as, to our knowledge, all other existing minority clustering methods proposed so far, include critical elements of supervision, in the form of parameters such as the number of foreground objects, the number of foreground clusters, and/or the foreground cluster sizes.

Additionally, we have considered three pseudo-systems for reference, to give lower and upper bounds of the performance of the actual systems:

RANDOM: A random clusterer, which assigns foreground and background clusters according to a Bernoulli distribution. We have taken the one among such clusterers which assigns the labels according to the actual source size ratio in the data.

ALLFG: A blind clusterer, which assigns all objects to the foreground cluster.

CONVEX: An oracle clusterer for the SYNTH dataset, which detects as foreground objects those objects that lie within the convex hull of the actual foreground sources. The out-

[7] For our experiments, we have used GeneDiver, an implementation of the method made available by its authors at http://www.ideal.ece.utexas.edu/~gunjan/genediver.

put of this CONVEX clusterer will hence detect all foreground objects, but include some background ones as false positives.

6.3 Protocol

In the preliminary evaluation of clustering consistency, for each one of the 10 samplings of the datasets in the SYNTH collection, 25 runs of every weak clustering algorithm were performed, and the source affinities have been estimated from the co-occurrence matrices of these 250 clusterings. We have then reported the fraction of datasets with which the considered methods are consistent (Cons), as well as, more precisely, the fraction of sources which are detectable by them—both macro- (M-Det) and micro-averaged (μ-Det) by dataset.

In order to assess and compare the performance of the different approaches in the full minority clustering evaluation, we have used the well-known measures of precision (Prc), recall (Rec) and F1, which have been previously employed for the evaluation of minority clustering (Ando 2007). The use of percentages when printing values of these metrics is customary.

Additionally, to evaluate the performance of the scoring phase, isolating it from that of threshold selection, we have also included information about Receiver Operator Characteristic (ROC) curves, more specifically, the Area Under the ROC Curve (AUC) (Fawcett 2006). The relation of dominance between ROC curves has been proved equivalent to that of precision/recall curves (Davis and Goadrich 2006), and they are less sensitive to variances of the class skew.

To reduce the impact of randomness, we have carried out 5 different runs for each method, configuration and dataset, and reported the average measures.

Finally, to compare the performance of the different methods across the synthetic datasets, we have used the Bergmann-Hommel non-parametric hypothesis test (Bergmann and Hommel 1988). Being non-parametric, the test judges the relative performances of the different methods with respect to each other, rather than their absolute scores or score differences. Recently, works such as that of Demšar (2006) have advocated for non-parametric tests to assess significance in machine learning tasks, as the assumption of metric commensurability across datasets, required by usual parametric tests such as Student or ANOVA, is often broken. The use of the Bergmann-Hommel test in particular has been recommended by García and Herrera (2008).

The graphical presentation of the results is that introduced by Demšar (2006): methods are placed along the horizontal axis according to their average ranks across datasets, and those for which no statistically significant difference can be found are joined by thick bars.

6.4 Results

The first Sect. 6.4.1 presents the results of the full experiments on the TOY dataset. The next two sections, 6.4.2 and 6.4.3, detail the results obtained over the SYNTH collection.

6.4.1 Clustering on the TOY *dataset*

A graphical depiction of the output of a representative subset of the compared approaches on the TOY dataset is shown in Fig. 5. The plots correspond to the parameter configurations achieving the best results.

The BBOCC method is unable to detect the multiple foreground sources and instead creates a single cluster covering two of them. Similarly, the BBCPRESS method, despite

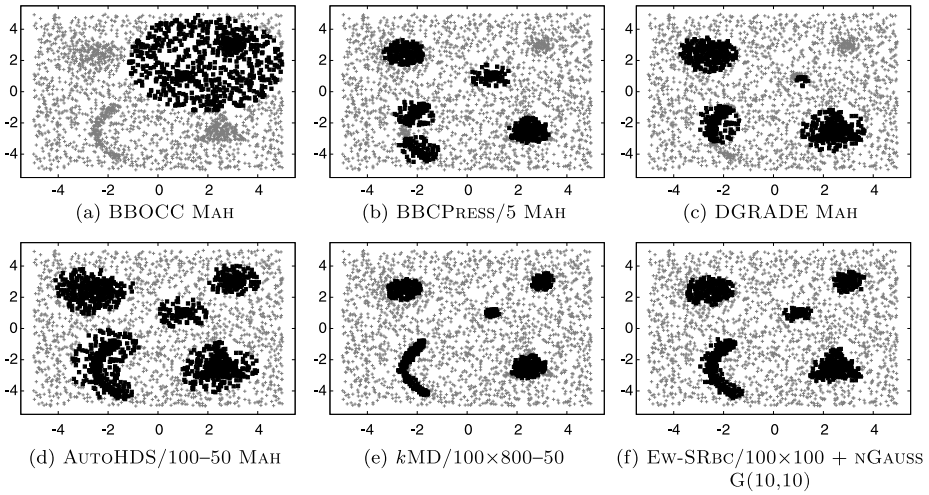


Fig. 5 System output for the compared methods (TOY data)

being given the correct number of sources, fails to recognize the half-moon-shaped one and instead splits it into two clusters, and rounds the triangle-shaped one. As a result, the top right source to be missed. The limitations of these two methods are well-known, and come from the fixed number and shape (hyperelliptical) of clusters they look for. Seeding BBC using DGRADE does not work in this case, either.

On the flipside, the AUTOHDS, *k*MD and EW-SRBC methods are able to recognize the variously shaped foreground sources. AUTOHDS seems to include too many background objects into the clusters, whereas the classification of the two other methods is more accurate. For this TOY dataset, *k*MD produces tighter clusters, favouring precision over recall, whereas for EW-SRBC this tendency is reversed.

The ROC curves for these approaches are plotted in Fig. 6. *k*MD and AUTOHDS do not provide an adjustable decision threshold; instead, their output is a fixed crisp boundary, and hence their ROC curve is composed of two straight segments. On the contrary, EW-SRBC, as all other EWOCS-based approaches, assigns a continuous score to all objects, and the separation between foreground and background ones is based on a threshold. Hence, its ROC curve, as a function of this threshold, is much smoother. For this reason, even if the differences in precision, recall and F1 score between the methods are small (see Fig. 6b), the curves for AUTOHDS and *k*MD are missing a large fraction of the AUC, which that of EW-SRBC is able to enclose. The fact will also be relevant to the evaluation on SYNTH.

Regarding the proposed threshold determination approaches, Fig. 7 shows the precision, recall and F1 curves for the output of EW-SRBC on TOY, according to the number of objects clustered as foreground. The cutoff points for the different criteria are plotted above the F1 curve. For this particular case, NGAUSS+VAR finds the same cutoff point as NGAUSS+BEST, and they are both plotted as NGAUSS.

The threshold values found by SIZE, NGAUSS and 2GAUSS are quite close to the optimal one, BEST. It is only the threshold found by DIST which falls somehow behind, trading in this case too much recall for precision.

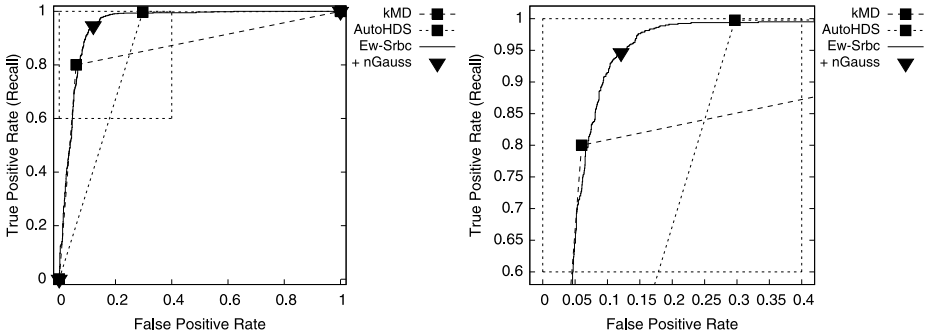


Fig. 6 ROC curves for AUTOHDS, *k*MD and EW-SRBC (TOY data)

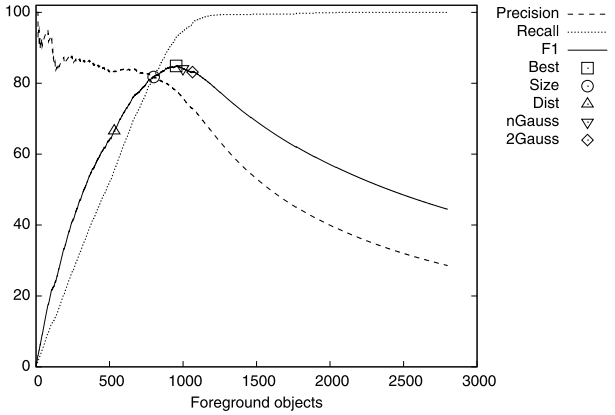


Fig. 7 Precision, Recall and F1 curves, and cutoff point determined by different threshold detection criteria (EW-SRBC on TOY data)

6.4.2 Consistency on the SYNTH *dataset collection*

Table 2 contains the values of consistency and averaged source detectability of the different weak clustering algorithms, estimated over all SYNTH datasets. Given that more dimensional data will exhibit a larger degree of sparsity which may render the results not comparable with those of lower dimensional datasets, we have opted to present the results segregated by the number of dimensions in the datasets.

Our hypothesis that weak clustering algorithms are consistent with data generated by dense and local sources seems corroborated by the empirical evidence coming from these experiments. The property holds in *all* tested datasets for 3-, 5- and 8-dimensional data. Only for 2-dimensional datasets, the algorithms, especially RSPLIT and SRBC using the MAH distance, fail to detect some of the sources—up to 7.45 % of them in the case of SRBC with MAH. Overall, for these two methods full consistency is only achieved in three fourths of the datasets; and HRBC fulfills the property in 91.67 % of the cases. On the flipside, the performance of SRBC using G(10, 10) is remarkable, as it obtains perfect consistency even in these harder cases. The results also confirm the intuition that 2-dimensional datasets, being less sparse, are harder to deal with.

However, even if perfect consistency is not achieved, the fact that, in the worst of the cases, more than 94% of the sources are detectable suggests that the lack of full consistency

Table 2 Consistency of the proposed weak clustering algorithms (SYNTH data)

			2 Dimensions			3 Dimensions		
			Cons	M-Det	μ-Det	Cons	M-Det	μ-Det
RSPLIT	×2	–	81.82	96.10	94.48	100.00	100.00	100.00
	×50	–	78.79	95.82	95.71	100.00	100.00	100.00
HRBC	×100	MAH	93.94	99.13	98.77	100.00	100.00	100.00
SRBC	×100	MAH	84.85	94.81	95.71	100.00	100.00	100.00
		G(10, 10)	100.00	100.00	100.00	100.00	100.00	100.00
			5 Dimensions			8 Dimensions		
			Cons	M-Det	μ-Det	Cons	M-Det	μ-Det
RSPLIT	×2	–	100.00	100.00	100.00	100.00	100.00	100.00
	×50	–	100.00	100.00	100.00	100.00	100.00	100.00
HRBC	×100	MAH	100.00	100.00	100.00	100.00	100.00	100.00
SRBC	×100	MAH	100.00	100.00	100.00	100.00	100.00	100.00
		G(10, 10)	100.00	100.00	100.00	100.00	100.00	100.00

does not necessarily hamper the actual performance of the EWOCS algorithm. The study of the clustering results over the same SYNTH collection in next section will shed light on this issue.

6.4.3 Clustering on the SYNTH *dataset collection*

Table 3 contains the AUC values for the compared methods across all datasets in the SYNTH collection, as well as their achievable precision, recall and F1 values, using the BEST threshold selection criterion.[8] As mentioned before, the degree of sparsity increases with the number of dimensions, and this simplifies the clustering task, and the results across datasets with different dimensionality may not be commensurable. For this reason, we have again opted to split the results according to dataset dimensionality.

For reasons of brevity, only the parameter configurations which achieve the best results for each method are included. Later in this same section, experiments studying the sensitivity of each method to the tuning of their parameters will be presented.

Finally, Fig. 8 contains a graphical representation of the outcome of Bergmann-Hommel tests on the F1 and AUC measures across all datasets in SYNTH. As mentioned previously, the position on the line indicates average rank across datasets (with 1 corresponding to a method consistently obtaining the highest score); and methods without a statistically significant difference between them are joined by thick bars.

In these experiments, EWOCS-based approaches are able to obtain results in the state of the art for minority clustering, and particularly, EW-SRBC is able to outperform the existing approaches for the task, achieving a performance close to the upper bound, given by CONVEX. We believe this is an excellent result, and one which confirms the validity of the EWOCS algorithm.

[8] For the AUTOHDS and kMD methods, which do not provide an adjustable decision boundary, the BEST results correspond directly to the clustering produced by the algorithm.

Table 3 Results for SYNTH data

			2 Dimensions				3 Dimensions			
			AUC	BEST			AUC	BEST		
				Prc	Rec	F1		Prc	Rec	F1
RANDOM	–		**0.500**	14.5	14.5	**14.5**	**0.500**	14.5	14.5	**14.5**
ALLFG	–		**0.500**	14.5	100.0	**24.9**	**0.500**	14.5	100.0	**24.9**
BBOCC	–	MAH	**0.752**	40.9	69.4	**44.5**	**0.841**	61.6	62.3	**56.9**
BBCPRESS	7	MAH	**0.849**	55.6	68.1	**60.7**	**0.934**	79.0	76.8	**77.4**
DGRADE	–	MAH	**0.897**	68.0	70.4	**68.2**	**0.969**	85.2	85.1	**84.9**
AUTOHDS	200–30	MAH	**0.871**	54.3	85.6	**64.5**	**0.875**	82.6	77.4	**78.1**
KMD	100 × 800–50		**0.808**	82.0	63.7	**68.5**	**0.945**	93.6	90.0	**91.6**
EW-RSPLIT	500 × 2	–	**0.843**	41.1	78.0	**52.1**	**0.911**	55.9	77.2	**63.6**
	500 × 50	–	**0.862**	45.0	75.9	**54.5**	**0.950**	66.0	83.9	**73.1**
EW-HRBC	100 × 100	MAH	**0.896**	59.1	71.5	**63.6**	**0.971**	76.1	85.0	**79.9**
EW-SRBC	100 × 100	MAH	**0.799**	37.1	73.3	**47.5**	**0.901**	53.6	78.2	**62.5**
		G(10, 10)	**0.958**	66.4	85.9	**74.6**	**0.991**	85.3	94.9	**89.7**
		G(AUTO)	**0.937**	64.5	83.7	**72.5**	**0.986**	83.7	93.2	**88.1**
CONVEX	–		**0.957**	67.6	100.0	**79.3**	**0.996**	95.4	100.0	**97.6**

			5 Dimensions				8 Dimensions			
			AUC	BEST			AUC	BEST		
				Prc	Rec	F1		Prc	Rec	F1
RANDOM	–		**0.500**	14.5	14.5	**14.5**	**0.500**	14.5	14.5	**14.5**
ALLFG	–		**0.500**	14.5	100.0	**24.9**	**0.500**	14.5	100.0	**24.9**
BBOCC	–	MAH	**0.942**	87.4	75.7	**79.9**	**0.993**	94.5	93.8	**94.0**
BBCPRESS	7	MAH	**0.983**	92.2	89.2	**90.2**	**0.996**	96.3	97.0	**96.6**
DGRADE	–	MAH	**0.984**	93.7	93.3	**93.4**	**0.998**	98.7	97.6	**98.1**
AUTOHDS	200–30	MAH	**0.855**	86.3	83.7	**78.6**	**0.843**	90.7	78.5	**76.8**
KMD	100 × 800–50		**0.983**	98.9	96.7	**97.8**	**0.991**	99.8	98.1	**99.0**
EW-RSPLIT	500 × 2	–	**0.961**	77.2	84.7	**80.1**	**0.991**	91.8	91.3	**91.3**
	500 × 50	–	**0.986**	87.4	91.2	**89.0**	**0.998**	96.0	97.6	**96.8**
EW-HRBC	100 × 100	MAH	**0.985**	86.7	90.4	**88.3**	**0.993**	91.6	94.2	**92.7**
EW-SRBC	100 × 100	MAH	**0.966**	79.9	85.9	**82.1**	**0.992**	91.2	94.2	**92.3**
		G(10, 10)	**0.999**	98.3	99.4	**98.8**	**0.996**	99.9	99.4	**99.6**
		G(AUTO)	**0.972**	90.4	96.5	**91.4**	**0.987**	96.1	99.7	**96.8**
CONVEX	–		**1.000**	100.0	100.0	**100.0**	**1.000**	100.0	100.0	**100.0**

BBOCC is the weakest approach among the compared ones. Even if its results are above the RANDOM and ALLFG baselines, the limitation to a single hyperelliptical cluster produces clusterings with a lower precision than those from other approaches. The differences are statistically significant in terms of both F1 and AUC.

Regarding EW-RSPLIT, the extension from 2 to a larger number of hyperplanes improves the performance of the RSPLIT algorithm within the ensemble. However, the algorithm favours too much recall over precision, and even if this allows it to achieve a good AUC measure, its values of F1 are lower than other methods which exhibit a similar performance,

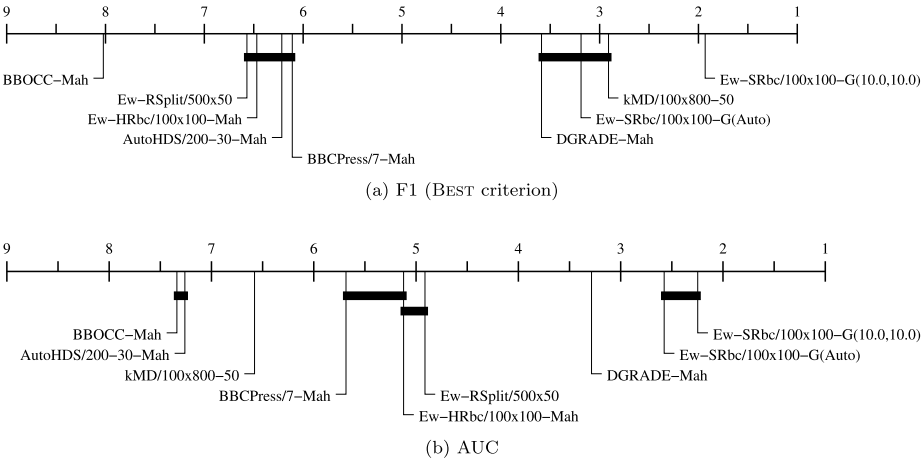


Fig. 8 Bergmann-Hommel tests for the compared approaches (SYNTH data)

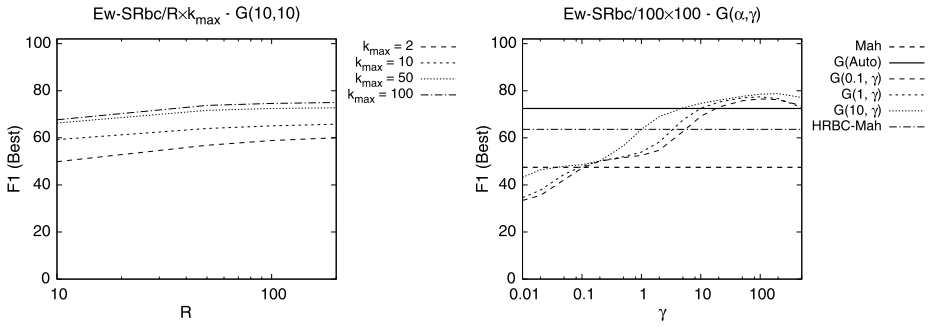
such as EW-HRBC and BBCPRESS. These too approaches trade some of the recall of EW-RSPLIT for precision, thus obtaining lower AUC but higher F1. The differences between the three systems, nevertheless, are deemed not significant by the Bergmann-Hommel test, and can hence be considered similar in terms of minority clustering power.

The results of AUTOHDS are also comparable in terms of F1 to those of these three methods. Hypothesis testing finds no statistically significant differences among them, either. However, the method seems unable to benefit from the increasing sparsity present in higher-dimensional datasets, obtaining the lowest F1 scores among all methods in the 8-dimensional ones. We believe the high sensitivity of the method to the tuning of its parameters (which we will consider below) can be an explanation for these poor results: it is unlikely that the same parameters produce good clusterings across all datasets in SYNTH. This seems a major drawback of the approach, and one which we think seriously reduces its utility in unsupervised minority clustering scenarios.

Finally, concerning DGRADE, kMD and EW-SRBC, their performance is significantly better than that of the other methods in terms of F1, and that of EW-SRBC is also better in terms of AUC. This is true for EW-SRBC not only when using the G(10,10) distance, which achieves the best results on SYNTH with a significant difference from the competing methods, but also when using the unsupervised one G(AUTO). The results for EW-SRBC using G(AUTO) are only slightly below those of kMD in terms of F1, and slightly below those obtained using G(10,10) in terms of AUC. In both cases the differences are not statistically significant. Taking into account that the determination of G(AUTO) is completely unsupervised, we believe we can qualify these results as really encouraging.

However, the results using the MAH distance within EW-SRBC fall much below those obtained with the G(α, γ) family. One reason for this behaviour may lie in the fixed degree of fuzziness allowed by MAH: the standardized scale that this distance provides may not always give the most suitable fuzziness. The greater versatility offered by the G(α, γ) distances is thus a valuable property.

Note that the high F1 score of kMD comes from its elevate precision, which is particularly high, for instance, in 3-dimensional datasets; whereas EW-SRBC tends to favour recall over precision and DGRADE seems to find more balanced solutions. These tendencies agree with the ones observed in the TOY dataset.



(a) Effect of R and k_{max} on F1 (b) Effect of α and γ on F1

Fig. 9 Effect of parameters on EW-SRBC (2-dimensional SYNTH data)

Finally, the values of AUC for AUTOHDS and kMD are lower than for all other methods except BBOCC. The difference comes, as mentioned in Sect. 6.4.1, from the lack of an adjustable threshold in their output.

At the light of these results, we can assert that EWOCS-based approaches perform competitively with respect to the state of the art in the minority clustering task, in terms of AUC and F1 of the obtained clusterings. Ensemble clustering methods have hence been proven to be useful for this task.

Moreover, the fact that the EW-SRBC method is able to outperform all other compared approaches when using the manually tuned Gaussian-kernel distance, and most of them when using the automatically tuned one, leads us to believe that, on the one hand, kernel-based distances are a serious alternative to other similarity measures used in clustering tasks; and that, on the other, the proposed RBC algorithm can be successfully employed to construct individual clusterings suitable for combination within a clustering ensemble.

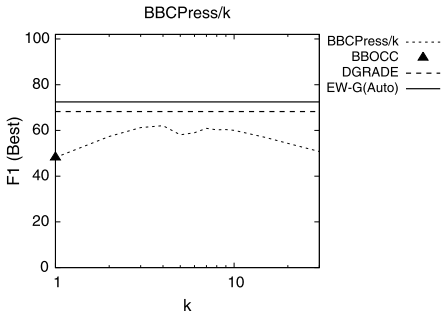
However, these conclusions require an evaluation of the sensibility to parameter tuning of the compared approaches.

Parameter sensitivity A number of experiments have been performed to assess the relevance of parameter tuning on the different approaches, in terms of the impact these parameters have in their performance on the minority clustering task.

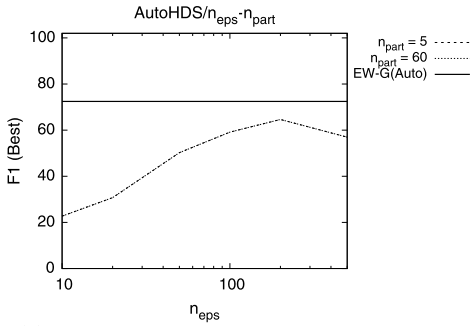
Figure 9 provides two plots of the BEST F1 score as a function of the parameters in EW-SRBC: the ensemble size R, the maximum number of clusters in each individual clustering k_{max}, and the Gaussian-kernel distance scaling factors α and γ. The plots correspond to the 2-dimensional subset of the SYNTH collection, being the datasets where the difference in performance between approaches is the largest.

First Fig. 9a plots the curves of F1 for a fixed distance function G(10,10). It can be seen how a change in any of the two parameters does influence the F1 score. However, the difference in performance is small, and, more importantly, the value stabilizes with increasing values of both R and k_{max}. In particular, the ensemble size R controls the convergence of the object scores to the source affinities. Higher values will provide more accurate clusterings, with the drawback of an increased computational cost. In these experiments, a value such as that of $R = k_{max} = 100$ we have used, produces good quality clusterings across a wide range of situations. Nevertheless, we will revisit their influence on clustering performance in the evaluation on real world collections.

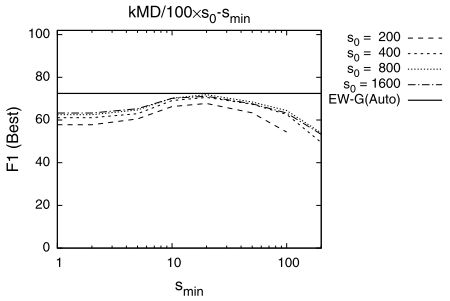
However, the plot in Fig. 9b, which shows the curves of F1 for fixed values of $R = k_{max} = 100$, presents a different picture. The scaling parameters of the Gaussian-kernel distance



(a) Effect of k on F1 of BBCPRESS



(b) Effect of n_{eps} and n_{part} on F1 of AUTOHDS



(c) Effect of s_0 and s_{min} on F1 of kMD

Fig. 10 Effect of parameters on BBCPRESS, AUTOHDS and kMD (2-dimensional SYNTH data)

also have an impact on the F1 of the clusterings produced by EW-SRBC, but in this case the values do not stabilize. Moreover, the curves for different α present a maximum around $\gamma = 10$, and lower values of F1 are obtained at either side of these maxima. The score using $G(\alpha, \gamma)$ distances can exceed significantly that obtained using MAH (both with EW-SRBC and EW-HRBC), but it can also eventually drop much below.

The selection of the suitable values for α and γ seems indeed a crucial issue when using EW-SRBC, as intuited in Sect. 4.3. Nevertheless, the plot in Fig. 9b also shows how the value of F1 obtained using the automatically tuned G(AUTO) distance provides an approximation to the optimum. We hence believe that G(AUTO) can be used to perform the minority clustering task satisfactorily, even if we must also admit that fine tuning can improve the overall results.

Regarding non-EWOCS-based approaches, Fig. 10 contains plots of the BEST F1 score for BBCPRESS, AUTOHDS and kMD, as a function of their various parameters. For reference, the plots also include the value obtained by EW-SRBC/100 $\times$ 100 using G(AUTO).[9]

DGRADE provides an effective alternative to BBCPRESS to obtain a suitable set of parameters and seeds for BBC (Fig. 10a). However, their computational cost limits its applicability for large collections. Concerning AUTOHDS and kMD (Figs. 10b and 10c), the latter seems more robust to the choice of its parameters. However, no method was proposed to automate the tuning of either method, other than interactive trial-and-error. EW-SRBC hence has as an advantage over the compared approaches, because of the automatic tuning

[9]For space reasons, the name is shortened to EW-G(AUTO).

Table 4 Results for 2-dimensional SYNTH data

			BEST			SIZE			DIST		
			Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
RANDOM	–		14.5	14.5	**14.5**	–	–	**–**	–	–	**–**
ALLFG	–		14.5	100.0	**24.9**	–	–	**–**	–	–	**–**
BBOCC	–	MAH	40.9	69.4	**44.5**	35.4	35.4	**35.4**	–	–	**–**
BBCPRESS	7	MAH	55.6	68.1	**60.7**	58.3	58.3	**58.3**	–	–	**–**
DGRADE	–	MAH	68.0	70.4	**68.2**	66.1	66.1	**66.1**	–	–	**–**
AUTOHDS	200–30	MAH	54.3	85.6	**64.5**	–	–	**–**	–	–	**–**
KMD	100 × 800–50		82.0	63.7	**68.5**	–	–	**–**	–	–	**–**
EW-RSPLIT	500 × 2	–	41.1	78.0	**52.1**	41.2	41.2	**41.2**	29.1	85.9	**42.2**
	500 × 50	–	45.0	75.9	**54.5**	48.0	48.0	**48.0**	30.2	87.9	**43.8**
EW-HRBC	100 × 100	MAH	59.1	71.5	**63.6**	61.2	61.2	**61.2**	37.8	86.3	**51.0**
EW-SRBC	100 × 100	MAH	37.1	73.3	**47.5**	38.1	38.1	**38.1**	25.0	82.6	**37.2**
		G(10,10)	66.4	85.9	**74.6**	71.2	71.2	**71.2**	45.2	97.4	**60.6**
		G(AUTO)	64.5	83.7	**72.5**	68.7	68.7	**68.7**	56.6	81.8	**63.4**
CONVEX	–		67.6	100.0	**79.3**	–	–	**–**	–	–	**–**

			NGAUSS+BEST			NGAUSS+VAR			2GAUSS		
			Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
EW-RSPLIT	500 × 2	–	39.5	77.7	**50.3**	39.2	58.7	**35.9**	34.2	81.2	**45.8**
	500 × 50	–	42.0	78.5	**51.8**	25.6	89.3	**32.2**	35.7	85.2	**46.8**
EW-HRBC	100 × 100	MAH	56.0	72.2	**60.5**	41.2	80.0	**45.6**	38.5	89.9	**48.7**
EW-SRBC	100 × 100	MAH	35.7	74.1	**46.3**	24.3	88.8	**31.7**	32.8	76.1	**43.1**
		G(10,10)	62.0	88.0	**70.9**	50.3	94.2	**64.2**	49.3	96.4	**64.6**
		G(AUTO)	58.5	87.2	**68.7**	63.7	60.1	**51.4**	51.6	90.2	**63.4**

procedure of the proposed G(AUTO) distance. Moreover, the results obtained using EW-SRBC and G(AUTO) are better than those of the compared approaches in terms of AUC and F1.

We believe the existence of such a tool is a significant difference with respect to other approaches, and that this makes EW-SRBC suitable for completely unsupervised minority clustering tasks.

Threshold determination Table 4 contains the values of precision, recall and F1 obtained when applying the different criteria to the output of each minority clustering method. Again, for brevity the table contains only the results across the 2-dimensional datasets of SYNTH. Concerning the statistical significance of the differences, Fig. 11 contains the graphical representation of the outcome of Bergmann-Hommel tests on precision, recall and F1 across all (not only 2-dimensional) datasets in SYNTH.

The results show there is still a gap between the maximum achievable F1 score (criterion BEST) and that obtained using the different criteria. There is another gap between the F1 scores of the criteria that contain some element of supervision (SIZE and NGAUSS+BEST) and those of the completely unsupervised ones (DIST, NGAUSS+VAR and 2GAUSS). These differences are present in a consistent way across all the EWOCS-based approaches.

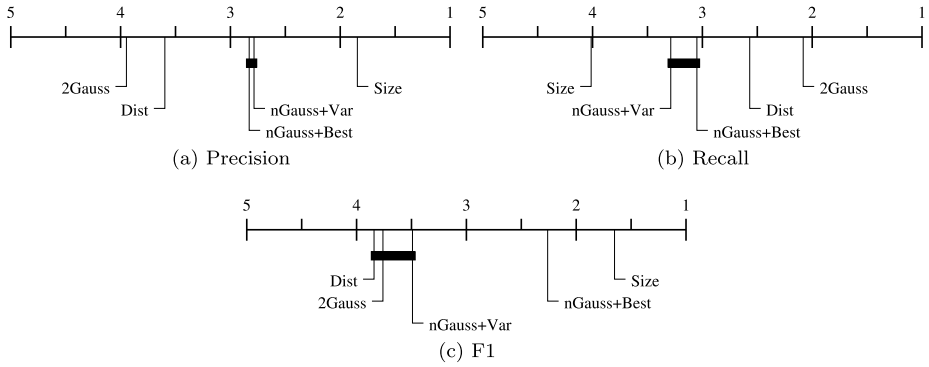


Fig. 11 Bergmann-Hommel tests for the compared criteria (EW-SRBC/100 × 100 using G(10,10) on SYNTH data)

Criterion SIZE is the one to obtain results closest to BEST in terms of F1, and that to obtain the best figures for precision, but at the cost of being the one which gives the least recall. All differences are statistically significant.

However, the upper bound achievable using Gaussian modelling of the scores, that of NGAUSS+BEST, lies quite close to the output of SIZE. For the EW-SRBC/100 × 100 method using G(10,10) on 2-dimensional data, the difference is only a 0.3 % in terms of F1. NGAUSS+BEST also shifts the bias towards recall instead of precision, which is much closer to the region where the optimal threshold (that of BEST) lies.

Finally, regarding the three unsupervised criteria, NGAUSS+VAR seems the one which comes closest in terms of performance to NGAUSS+BEST. Even if this does not hold for the particular subset of 2-dimensional data, overall NGAUSS+VAR gives higher precision and lower recall than NGAUSS+BEST. These differences are not statistically significant, but overall the one in F1 score is. DIST and 2GAUSS show a strong bias for recall, particularly the latter, and fall much below NGAUSS+BEST in precision. They perform worse in terms of F1 than the other proposed approaches. However, from the statistical point of view, the difference is not significant between them and NGAUSS+VAR.

Taking these and all obtained results into account, we can affirm that, even if elements of supervision improve the results in the task of minority clustering, the proposed EWOCS algorithm allows us to obtain competitive results using an unsupervised[10] approach: the results obtained by EW-SRBC/100 × 100 using G(AUTO) and one of DIST, NGAUSS+VAR or 2GAUSS are above those obtained by other supervised approaches, such as BBOCC or BBCPRESS.

Regarding the elements of supervision introduced by each one of the criteria, it is remarkable that the use of NGAUSS+BEST, which would require an a posteriori selection of the number of background Gaussian components from a small number of them, suffices for EW-SRBC/100 × 100 using G(AUTO) to outperform all other approaches, including kMD, which requires careful tuning of three parameters R, s_0, s_{min}.

Even if manual determination of the most suitable G(α, γ) distance, or more informed (i.e., supervised) threshold detection criteria, such as SIZE or BEST, allow further increases in the F1 scores obtained by EW-SRBC, we believe that the fact that, using no or little

[10] As observed in the previous section, the EW-SRBC method is robust to the tuning of R and k_{max}, so we can consider it unsupervised.

supervision, EW-SRBC outperforms supervised minority clustering approaches in the state of the art is an excellent result, and one which proves the validity of the whole minority clustering framework introduced by the EWOCS algorithm.

7 Evaluation on real-world data

We have carried out a number of additional experiments with the goal of extending our conclusions to larger and higher-dimensional collections coming from real-world problems.

The first dataset we have used belongs to the text classification domain. Specifically, we have used a subset of the Reuters-21578 corpus,[11] which is a popular benchmark for the task. The Reuters corpus was previously used by Ando (2007) to evaluate their minority clustering algorithm.

Our second collection of datasets comes from the area of information extraction, within which, as mentioned previously, Gonzàlez and Turmo (2009) introduced the EWOCS algorithm. More specifically, the authors considered the problem of unsupervised relation detection (i.e., learning which pairs of *entities* mentioned in a document collection are linked by some *relation* without resorting to annotated data), and proposed a reduction of the problem to minority clustering. EWOCS was then used to find foreground objects, which in the context of the task corresponded to pairs of related entities.

The experiments presented in the following sections extend those of the previous work, and compare the results of EWOCS not only to other relation detection approaches, as in the original paper, but also to other minority clustering approaches. Sect. 7.1 describes the used corpora and data generation procedure. Sect. 7.2 reviews the approaches that we have considered for the task. Finally, Sect. 7.3 presents the obtained results.

7.1 Data

As previously mentioned, the Reuters corpus has been used by other authors to evaluate minority clustering algorithms. Specifically, Ando (2007) assembled a dataset which contained the documents in topics *oilseed*, *money-supply*, *sugar* and *gnp* as foreground objects, and those in *acq* as background.

However, in preliminary experiments we found this partition not to provide a real minority clustering problem—but rather an all-in clustering one with unequal cluster sizes. For this reason, we decided to use a different subset of the collection, in which clusters have to be determined on the grounds of density. The documents belonging to the largest category, *earn*, have been taken as foreground objects, and the rest of documents as background ones. To reduce the density of the background, only a random 60 % of its documents has been kept. The resulting dataset[12] has a total of 3987 and 10507 documents belonging to each one of the two classes, respectively.

Similarly to other works in document clustering (Zhao and Karypis 2004), the text in each document has been tokenized, and numbers and stop words have been removed. Last, the remaining tokens have been stemmed using the method of Porter (1980), and the *tf-idf* vectors have been found (Spärck-Jones 1972). We will refer to the obtained dataset as REUTERS.

[11] Available at http://www.daviddlewis.com/resources/testcollections/reuters21578/.

[12] Available at http://www.lsi.upc.edu/~egonzalez/data/ml-reuters.tar.gz.

Table 5 ACE entity types

FAC	Facility	PER	Person
GPE	Geo-Political	ORG	Organization
LOC	Location	VEH	Vehicle

Table 6 Size of APW-ACE datasets

	Objects		Dims.		Objects		Dims.
	APW	ACE			APW	ACE	
FAC-GPE	57917	3394	765	GPE-VEH	14713	1241	647
FAC-LOC	9766	630	574	LOC-PER	74286	5110	834
FAC-PER	149677	7098	877	ORG-PER	844331	30558	950
GPE-LOC	39758	4203	738	ORG-VEH	12929	731	636
GPE-ORG	273209	13055	910	PER-VEH	37856	2597	760
GPE-PER	576566	39730	943				

Regarding the relation detection datasets, and following Gonzàlez and Turmo (2009), a hold-out evaluation scheme has been used: minority clustering is first performed on the objects generated from a large document collection, and the obtained clustering models are then applied on additional objects from new documents, where performance is measured. We have used the year 2000 subset of the Associated Press section of the AQUAINT Corpus to perform the clustering (Graff 2002), and the hold-out datasets are generated from the annotated corpora used in the Relation Detection and Recognition task of the ACE evaluation (ACE 2008), for which ground truth is available. Specifically, we used the training data of ACE evaluations for years 2003, 2004 and 2008. The corpora add up to almost 29 million and over half a million words, respectively.

Each dataset in the collection will contain binary feature vectors which capture syntactical properties of the contents of pairs of entities of two considered types (e.g., for ORG-PER, one of the two entities will be an organization and the other one a person). We have considered for evaluation the 11 entity type pairs that were already used by Gonzàlez and Turmo (2009). Table 5 contains a quick overview of the entity types annotated in the corpus.[13]

The set of syntactic features that have been used to generate the binary vectors to be clustered is the part-of-speech–based one used in the original paper.[14] Features occurring in less than ten objects are filtered. The number of objects and dimensions in the resulting datasets are listed in Table 6. We will refer to this dataset collection as APW-ACE.

7.2 Approaches

In order to assess the generality of the results on the SYNTH collection over real-world data, we have used the same set of methods presented in Sect. 6.2 for this new set of experiments. Thus, the BBOCC, BBCPRESS, DGRADE, AUTOHDS and kMD algorithms have been applied over REUTERS, as well as the RANDOM and ALLFG baselines. For the

[13] We refer to the ACE annotation guidelines for details about the entity classification scheme (ACE 2008).

[14] We refer to Gonzàlez (2012) for a language-processing oriented analysis of more complex features.

divergence-based approaches, we have resorted to EUC rather than MAH because of the highest computational cost of the latter as the number of dimensions grows.

Regarding *k*MD, multinomial distributions have been used for both the foreground and background clusters. Additionally, for this particular algorithm, instead of using a *tf-idf* representation of the REUTERS dataset, we have employed the unsupervised feature selection scheme of Slonim and Tishby (2000): documents are represented using raw term frequencies, but, to reduce data dimensionality, only the 200 stems that contribute the most to the mutual information between stems and documents are selected. This configuration mimics that used by Ando (2007) on the same corpus.

The much larger sizes of the datasets in APW-ACE renders impossible the use of some of the previous methods, namely DGRADE and AUTOHDS, because of their cubic computational complexity. Nevertheless, we have kept the rest of approaches in the comparison; the only change has been the use of Bernoulli rather than multinomial distributions in *k*MD, the former being more suitable for binary feature vectors.

Moreover, in order to compare the performance of minority clustering approaches with respect to other relation detection methods, we have included an additional method in our comparison:

GRAMS: as proposed by Hassan et al. (2006). The method uses a combination of n-gram models and graph-based mutual reinforcement to generate POS-based patterns, sorted by confidence, which can then be applied on new data. The approach requires no additional external resources and acquires patterns which can be applied to hold-out data, and thus allows a fair comparison within the present setting.

The authors of the GRAMS method do not provide a way to determine a threshold value for the confidence of the patterns so, similarly to other methods, we are taking the BEST value in terms of obtained F1 score. Thus, the results displayed for GRAMS are an upper bound of the performance of the method.

7.3 Results

Next two sections expose and analyze the results of the experiments on the two considered real-world scenarios: Sect. 7.3.1 deals with those on the REUTERS dataset, and Sect. 7.3.2 details the outcome of the evaluation on APW-ACE.

7.3.1 Clustering on the REUTERS *dataset*

Table 7 contains the AUC values for the compared methods on the REUTERS dataset, as well as the precision, recall and F1 values obtained using the different threshold selection criteria. Similarly to Sect. 6.4.3, only the configurations which achieve the best results for each method are included.

Strikingly, the results obtained by *k*MD are well below those of the baseline RANDOM and ALLFG methods—contrary to the excellent performance shown on the SYNTH datasets. In particular, the obtained recall is extremely poor, below 1 %, and precision barely reaches 25 % for the *k*MD/100 × 800–5 setting, which is the one to obtain the best results among those tried. *k*MD/100 × 800–50, which was used by Ando (2007) on Reuters documents, achieves even lower precision, down to 18 %. Overall, F1 remains around 1.5 %, clearly pointing that the feature selection scheme, or the multinomial distribution modelling used, or both, are not suitable for the task at hand.

Table 7 Results for REUTERS data

			AUC	BEST			SIZE			DIST		
				Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
RANDOM	–		**0.500**	27.5	27.5	**27.5**	–	–	**–**	–	–	**–**
ALLFG	–		**0.500**	27.5	100.0	**43.1**	–	–	**–**	–	–	**–**
BBOCC	–	EUC	**0.971**	87.2	88.3	**87.7**	87.6	87.6	**87.6**	–	–	**–**
BBCPRESS	4	EUC	**0.834**	74.6	59.7	**66.2**	65.0	65.0	**65.0**	–	–	**–**
DGRADE	–	EUC	**0.887**	79.6	66.0	**72.2**	71.3	71.3	**71.3**	–	–	**–**
AUTOHDS	200–15	EUC	**0.507**	27.8	100.0	**43.5**	–	–	**–**	–	–	**–**
KMD	100 × 800–5		**0.499**	24.5	0.6	**1.3**	–	–	**–**	–	–	**–**
	100 × 800–50		**0.497**	18.0	0.8	**1.5**	–	–	**–**	–	–	**–**
EW-SRBC	100000 × 50	EUC	**0.902**	76.8	71.8	**74.1**	73.7	73.7	**73.7**	97.0	51.7	**67.4**

			AUC	NGAUSS+BEST			NGAUSS+VAR			2GAUSS		
				Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
EW-SRBC	100000 × 50	EUC	**0.902**	85.1	64.2	**71.0**	94.5	56.8	**71.0**	59.2	83.3	**69.2**

AUTOHDS does also perform poorly on this dataset, assigning almost all objects to the foreground clusters. Its results are hence virtually indistinguishable from those of the ALLFG baseline.

Regarding the BBOCC, BBCPRESS, DGRADE and EW-SRBC methods, their results are placed high above the baselines. In fact, the best results for the task are achieved with BBOCC (equivalently, BBCPRESS/1) which gives an AUC value of 0.971 and F1 of 87.7 %. The values clearly exceed the AUC of 0.902 and F1 of 74.1 % achievable by EW-SRBC using the BEST threshold. It is surprising how this method, the simplest one after the baselines, is also the one to obtain the best results, exceeding the proposed EWOCS method by such a margin.

However, there is a number of factors to take into account concerning the generality of this statement. First, by the construction of REUTERS dataset, the problem is well-suited for methods looking for one (and only one) dense foreground cluster surrounded by a sparse background. One proof of this is that the second best AUC and F1 values obtained by BBCPRESS are those with $k = 4$, lying much below those for $k = 1$, and also those of EW-SRBC. There is thus a strong sensitivity to the value of parameter k. In this sense, DGRADE continues to provide a better (and less supervised) starting model for BBC-style clustering, even if its results are still slightly below those of EW-SRBC.

Moreover, methods BBOCC and BBCPRESS use the SIZE threshold selection criterion, and hence require the number of foreground objects to be known a priori. Figure 12 shows the precision, recall and F1 values achieved by BBOCC as a function of the provided number of foreground objects (expressed as a ratio of the total dataset size). The F1 value obtained by EW-SRBC using NGAUSS+VAR—a completely unsupervised approach—is included for reference: a bad estimation of the foreground cluster size causes the precision/recall balance to break, and the F1 scores to fall from the optimal ones, located around the actual foreground ratio value of 27.5 %.

There is thus an inherent brittleness in the fitting of parameters for BBOCC, BBCPRESS and, even if to a lesser extent, DGRADE (despite being able to determine the number of foreground clusters, it does require the number of foreground objects as input) on

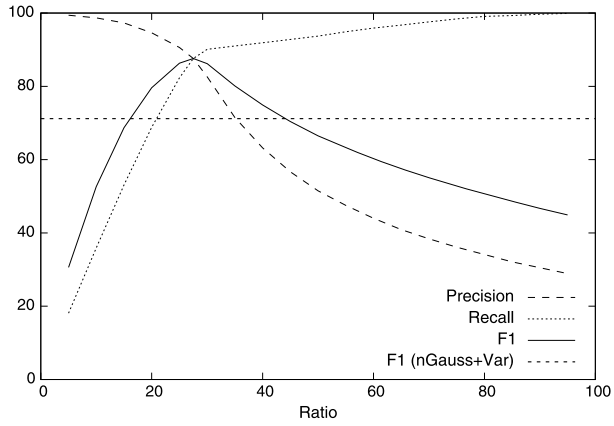
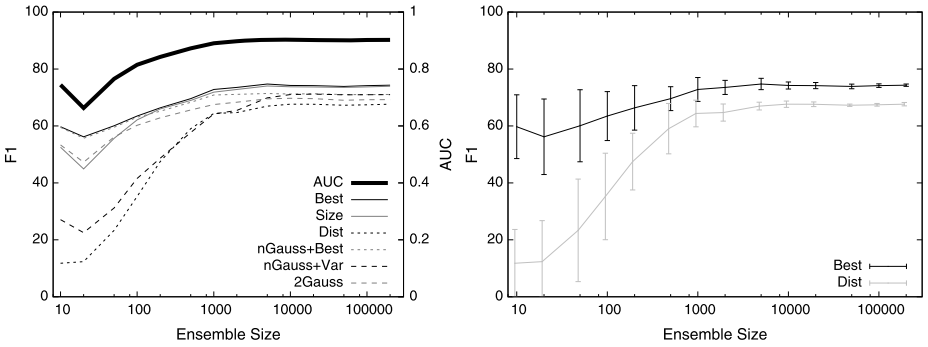


Fig. 12 Effect of foreground size ratio on BBOCC performance (SIZE criterion on REUTERS data)



(a) AUC/F1 (b) Standard deviation of F1

Fig. 13 Effect of R on EW-SRBC/$R \times 50$ performance (using EUC on REUTERS data)

REUTERS—in the same way as we had found it for SYNTH—and this can become an important drawback if the dataset characteristics change.

Concerning the different threshold selection criteria available for EW-SRBC, it is encouraging to see how the REUTERS dataset allows a much better identification by part of the unsupervised methods—namely DIST, NGAUSS+VAR and 2GAUSS. The value of F1 achieved by NGAUSS+VAR matches the upper bound of Gaussian-based criteria NGAUSS+BEST, and fall only 3 points below the upper bound value obtained with BEST. The gap between the results obtained with DIST and 2GAUSS and those with the supervised thresholds BEST and SIZE is also smaller in this case than it was for 2-dimensional SYNTH datasets (Table 4). It is also worth noting how, for this dataset, NGAUSS+VAR and DIST produce precision-biased clusterings, whereas 2GAUSS gives more recall-favouring ones.

These results are encouraging, but we believe an analysis of the performance of EWOCS on REUTERS as the ensemble size parameter R increases is required to obtain insights into its behaviour. Such an analysis is provided in the following paragraphs.

Ensemble size sensitivity Figure 13a contains a plot of the performance of EW-SRBC method, in terms of AUC and F1, using a fixed value of $k_{max} = 50$ and successively increasing ensemble sizes R, on the REUTERS dataset.

Increases in the ensemble size lead to an improvement of the performance of EWOCS, up until a point where the results stabilize. This matches the behaviour observed on SYNTH (Sect. 6.4.3). Nevertheless, given the greater complexity of the task, a larger number of individual clusterings are required for the results to converge—in fact, one several orders of magnitude larger.

This improvement is more relevant for the unsupervised DIST and NGAUSS+VAR. Thus, we believe the observed performance boost comes more from an increase in the gap between the scores of foreground and background objects—one which allows unsupervised criteria to detect the threshold more accurately—than from significant changes in the relative values of the scores. If this second phenomenon were the case, the improvements would affect equally both unsupervised and supervised criteria.

Figure 13b contains a plot of the mean and standard deviation of F1 across 10 runs of EW-SRBC, using two of the proposed criteria.[15] One can observe how the standard deviation of the results is reduced considerably as the ensembles grow in size, almost disappearing by the time the number of clusters reaches $R = 100000$.

Overall, the results of this last series of experiments confirm those in Sect. 6.4.3: the parameter R does have a considerable influence on the results obtained by EWOCS-based approaches. In particular, a larger clustering ensemble increases the separation between the scores of background and foreground objects, thus improving the accuracy of the threshold detection stage. The evaluation on the larger datasets from APW-ACE will provide more insights about the runtime trade-offs associated to the setting of the R parameter, and we hence defer further discussion to that point.

7.3.2 Hold-out clustering on the APW-ACE *dataset collection*

Table 8 contains the full table of results for the compared approaches on each one of the datasets in the APW-ACE collection. For reasons of space, the divergence used by each method has been omitted: it is EUC for BBOCC and BBCPRESS, and G(0.1, 0.1) for EW-SRBC. Figure 14 contains the Bergmann-Hommel tests for the F1 score using the BEST criterion and the AUC metric, which summarize and assess the statistical significance of the results in the table.

As seen in both the table and the figure, minority clustering algorithms outperform the reference unsupervised relation detection approach GRAMS both in terms of AUC and F1 score. Only kMD obtains lower scores, as its behaviour degrades towards the ALLFG baseline: it assigns almost all objects to the foreground. We believe the poor performance exhibited by kMD in both this and the REUTERS collection—compared to the good results it obtained on SYNTH, where the Gaussian distributions in the data matched those in the model—casts doubts on the suitability of the method on datasets whose sources follow non-standard or unknown distributions.

Concerning the other three methods, the hypothesis test finds no significant differences between BBOCC, BBCPRESS and EW-SRBC, even if the last is the one to provide the best AUC and F1 scores overall.

Regarding the detection of the threshold, Table 9 contains the results for each one of the datasets and criteria of the EW-SRBC/50000 × 100 method. The results are similar to those in SYNTH and REUTERS, with the extra supervision used by SIZE allowing it to stay within 2–3 % of the F1 score of BEST, and the two unsupervised methods DIST and 2GAUSS

[15] The results are similar for the other four, and are omitted here for brevity.

Table 8 Results for APW-ACE data

		FAC-GPE				FAC-LOC				FAC-PER			
		AUC	BEST			AUC	BEST			AUC	BEST		
			Prc	Rec	F1		Prc	Rec	F1		Prc	Rec	F1
RANDOM	–	**0.500**	18.1	18.1	**18.1**	**0.500**	23.5	23.5	**23.5**	**0.500**	16.8	16.8	**16.8**
ALLFG	–	**0.500**	18.1	100.0	**30.6**	**0.500**	23.5	100.0	**38.0**	**0.500**	16.8	100.0	**28.7**
GRAMS	–	**0.754**	67.6	56.8	**61.7**	**0.624**	64.7	29.7	**40.7**	**0.592**	51.0	22.9	**31.6**
BBOCC	–	**0.900**	61.1	74.6	**67.2**	**0.818**	61.6	62.8	**62.2**	**0.746**	35.8	62.6	**45.5**
BBCPRESS	10	**0.899**	58.2	79.3	**67.0**	**0.809**	65.3	61.6	**63.4**	**0.751**	40.9	55.3	**47.0**
KMD	100 × 800–50	**0.500**	18.1	100.0	**30.6**	**0.500**	23.5	100.0	**38.0**	**0.500**	16.8	100.0	**28.7**
EW-SRBC	50000 × 100	**0.907**	59.7	78.1	**67.6**	**0.814**	62.5	64.1	**63.3**	**0.759**	38.3	59.5	**46.6**

		GPE-LOC				GPE-ORG				GPE-PER			
		AUC	BEST			AUC	BEST			AUC	BEST		
			Prc	Rec	F1		Prc	Rec	F1		Prc	Rec	F1
RANDOM	–	**0.500**	15.6	15.6	**15.6**	**0.500**	11.2	11.2	**11.2**	**0.500**	12.3	12.3	**12.3**
ALLFG	–	**0.500**	15.6	100.0	**26.9**	**0.500**	11.2	100.0	**20.2**	**0.500**	12.3	100.0	**21.9**
GRAMS	–	**0.767**	67.8	58.6	**62.8**	**0.847**	68.3	73.8	**70.9**	**0.777**	55.1	62.6	**58.6**
BBOCC	–	**0.892**	57.1	74.8	**64.8**	**0.923**	54.6	73.9	**62.8**	**0.866**	58.7	59.5	**59.1**
BBCPRESS	10	**0.894**	72.0	62.4	**66.8**	**0.926**	66.4	63.5	**64.9**	**0.878**	62.9	59.3	**61.0**
KMD	100 × 800–50	**0.500**	15.6	100.0	**26.9**	**0.500**	11.2	100.0	**20.2**	**0.500**	12.3	100.0	**21.9**
EW-SRBC	50000 × 100	**0.897**	59.7	73.1	**65.7**	**0.922**	59.1	70.5	**64.3**	**0.877**	61.0	59.7	**60.3**

		GPE-VEH				LOC-PER				ORG-PER			
		AUC	BEST			AUC	BEST			AUC	BEST		
			Prc	Rec	F1		Prc	Rec	F1		Prc	Rec	F1
RANDOM	–	**0.500**	12.9	12.9	**12.9**	**0.500**	11.1	11.1	**11.1**	**0.500**	11.8	11.8	**11.8**
ALLFG	–	**0.500**	12.9	100.0	**22.8**	**0.500**	11.1	100.0	**19.9**	**0.500**	11.8	100.0	**21.1**
GRAMS	–	**0.738**	71.1	50.6	**59.1**	**0.611**	47.9	25.7	**33.4**	**0.813**	52.1	71.4	**60.2**
BBOCC	–	**0.886**	65.5	59.4	**62.3**	**0.767**	32.3	60.4	**42.1**	**0.894**	51.7	62.3	**56.5**
BBCPRESS	10	**0.884**	55.4	66.9	**60.6**	**0.798**	36.2	55.2	**43.7**	**0.901**	55.0	64.3	**59.3**
KMD	100 × 800–50	**0.500**	12.9	100.0	**22.8**	**0.500**	11.1	100.0	**19.9**	**0.500**	11.8	100.0	**21.1**
EW-SRBC	50000 × 100	**0.888**	59.7	63.6	**61.6**	**0.798**	35.9	57.3	**44.1**	**0.906**	53.4	69.5	**60.4**

		ORG-VEH				PER-VEH			
		AUC	BEST			AUC	BEST		
			Prc	Rec	F1		Prc	Rec	F1
RANDOM	–	**0.500**	13.8	13.8	**13.8**	**0.500**	10.7	10.7	**10.7**
ALLFG	–	**0.500**	13.8	100.0	**24.3**	**0.500**	10.7	100.0	**19.3**
GRAMS	–	**0.749**	91.1	50.5	**65.0**	**0.612**	59.1	24.5	**34.7**
BBOCC	–	**0.889**	81.5	65.3	**72.5**	**0.807**	34.4	59.2	**43.5**
BBCPRESS	10	**0.880**	75.3	63.4	**68.8**	**0.800**	41.2	46.2	**43.5**
KMD	100 × 800–50	**0.500**	13.8	100.0	**24.3**	**0.500**	10.7	100.0	**19.3**
EW-SRBC	50000 × 100	**0.886**	79.8	66.3	**72.4**	**0.808**	33.0	63.0	**43.3**

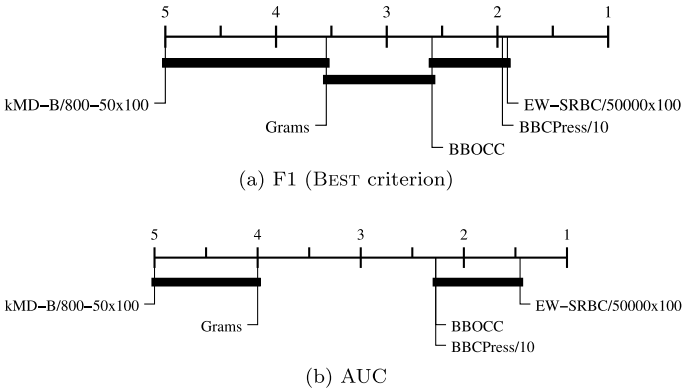


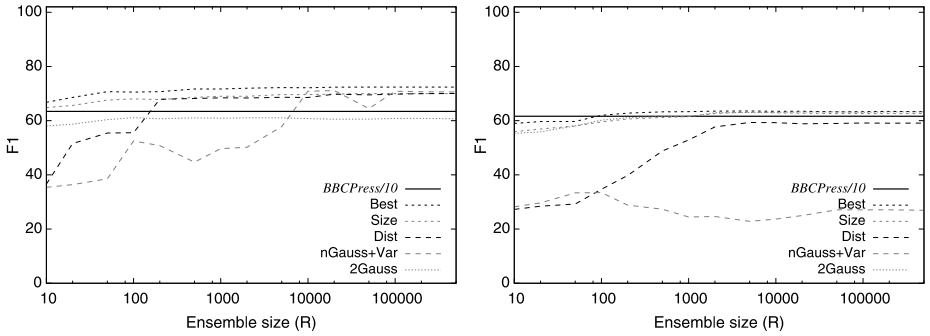
Fig. 14 Bergmann-Hommel tests for the compared approaches (APW-ACE data)

Table 9 Results for APW-ACE data (EW-SRBC/50000 × 100 using G(0.1, 0.1))

	BEST			SIZE			DIST			2GAUSS			NGAUSS + VAR		
	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
FAC-GPE	59.7	78.1	**67.6**	65.8	65.9	**65.9**	70.6	59.6	**64.6**	60.2	76.0	**67.2**	86.9	17.4	**29.0**
FAC-LOC	62.5	64.1	**63.3**	62.4	62.8	**62.6**	64.1	54.6	**59.0**	62.5	64.1	**63.3**	57.7	17.7	**27.1**
FAC-PER	38.3	59.5	**46.6**	43.0	43.1	**43.0**	46.3	33.4	**38.8**	40.2	51.9	**45.3**	60.8	16.3	**25.7**
GPE-LOC	59.7	73.1	**65.7**	63.9	64.0	**63.9**	61.0	69.3	**64.9**	52.9	80.9	**64.0**	81.1	31.7	**45.6**
GPE-ORG	59.1	70.5	**64.3**	61.2	61.8	**61.5**	52.2	76.2	**61.9**	42.3	88.4	**57.2**	66.4	34.9	**45.8**
GPE-PER	61.0	59.7	**60.3**	60.2	60.2	**60.2**	56.1	63.3	**59.5**	41.4	77.7	**54.0**	78.8	37.1	**50.4**
GPE-VEH	59.7	63.6	**61.6**	59.2	59.6	**59.4**	47.5	71.5	**57.0**	39.5	85.0	**54.0**	69.4	43.7	**53.5**
LOC-PER	35.9	57.3	**44.1**	40.8	40.9	**40.8**	38.0	44.8	**41.1**	33.4	60.9	**43.1**	47.0	18.7	**26.7**
ORG-PER	53.4	69.5	**60.4**	58.3	58.3	**58.3**	59.4	56.7	**58.0**	41.5	82.5	**55.3**	73.9	28.0	**40.6**
ORG-VEH	79.8	66.3	**72.4**	69.6	70.3	**70.0**	69.3	69.7	**69.5**	48.8	79.8	**60.6**	77.0	58.2	**64.5**
PER-VEH	33.0	63.0	**43.3**	39.7	39.9	**39.8**	37.9	41.7	**39.7**	32.8	62.6	**43.1**	53.1	29.3	**37.8**

providing similar result slightly below those of the supervised ones. Only the behaviour of NGAUSS+VAR is significantly different, its figures being much lower than those of its counterparts. We will return to this issue shortly, and try to provide a likely explanation for it.

With respect to the relation between the performance of the diverse threshold detection criteria and the size, the trend observed in REUTERS appears again in APW-ACE. For the DIST and 2GAUSS criteria, increasing the ensemble size improves their scores and reduces the gap between them and the supervised BEST and SIZE. However, for criterion NGAUSS+VAR again, the pattern is not so clear: whereas for a few pairs (GPE-PER, GPE-VEH, ORG-VEH, PER-VEH) the detected threshold improves as more clusterings are added to the ensemble, in the rest of the datasets the performance metrics stagnate in the lower part of the scale. To illustrate this phenomenon, Fig. 15 contains two plots of the F1 score



(a) ORG-VEH (b) FAC-LOC

Fig. 15 Effect of R on EW-SRBC/$R \times 100$ performance (using G(0.1,0.1) on ACE-APW data)

as a function of the ensemble size R for the ORG-VEH and FAC-LOC datasets.[16] The value achieved by BBCPRESS/10 using the BEST threshold is also included for reference.

This inconsistency can be due to the use of Eq. (17): if, without being as large as the background one, the variances of diverse foreground sources differ significantly from one another, it is possible for the point of maximum inter-variance difference to fall among the foreground objects, thus providing a threshold with higher precision and lower recall. This could be the case in datasets generated from relation detection problems, because, for a given entity type pair, some relations can be expressed using a reduced set of linguistic patterns (and thus give place to particularly dense regions), whereas for other there can be a wider variety. The criterion thus may not be robust to foreground sources of heterogeneous density—and further exploration is required in order to improve it.

Convergence and runtime As mentioned at the end of Sect. 7.3.1, augmenting the ensemble size R increases the separation between the scores of background and foreground objects, thus improving the accuracy of the threshold detection stage. We believe the larger datasets in APW-ACE offer a good testbed to study the ratio of convergence of these scores.

Differently from other iterative algorithms, the fact that weak clustering algorithms are being used means that convergence of the scores is not smooth, but presents an alternation of larger and smaller steps. To study this process, we have considered the average score change produced by the R-th clustering:

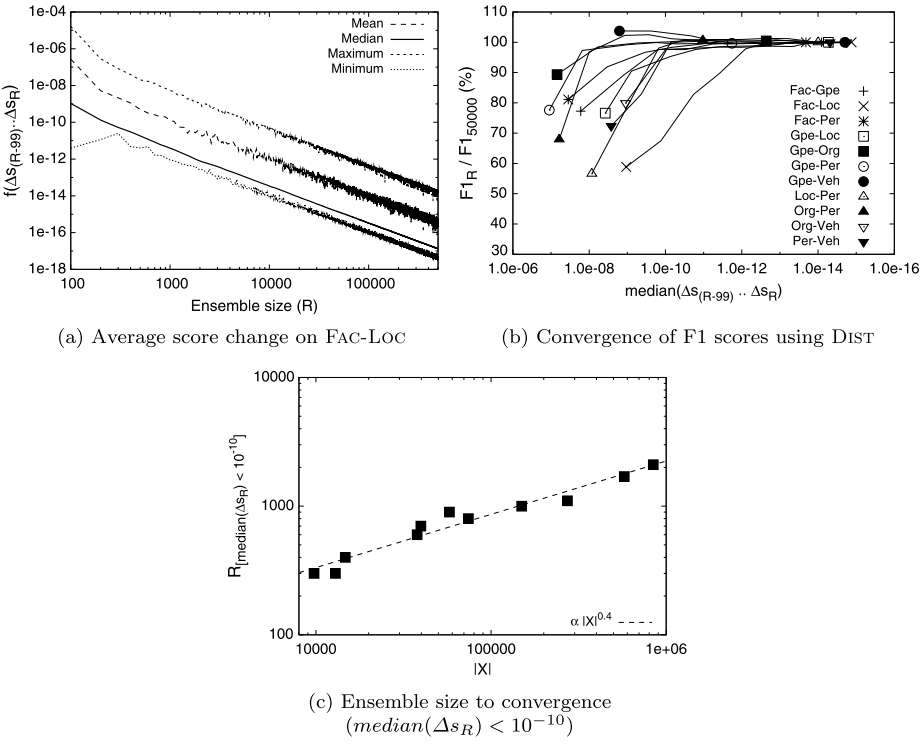
$$\Delta s_R = \frac{\sum_{x_i \in \mathcal{X}} (s^\star_{Ri} - s^\star_{(R-1)i})^2}{|\mathcal{X}|}$$

Figure 16a contains a sample plot of the values of Δs_R, aggregated in disjoint windows of 100 repeats, using DIST on the FAC-LOC dataset.[17] It can be seen how the maximum, mean and minimum values show an overall descending trend, yet present continuous oscillations. On the contrary, the medians exhibit a smooth decreasing behaviour, and seem thus to be useful as indicators of the convergence rate of the scores.

To confirm this intuition, Fig. 16b shows the values of F1 achieved with criterion DIST after R weak clusterings (being of the methods which benefit the most from an increased

[16] The rest of the plots have been omitted here for brevity.

[17] Again, the rest of the plots have been omitted here for brevity.



(a) Average score change on FAC-LOC

(b) Convergence of F1 scores using DIST

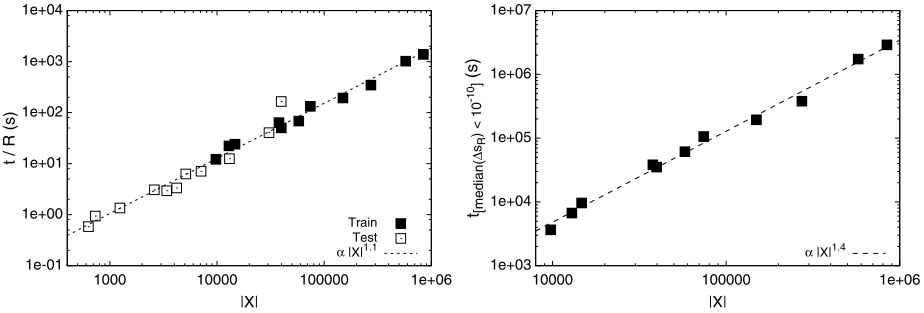
(c) Ensemble size to convergence ($median(\Delta s_R) < 10^{-10}$)

Fig. 16 Score convergence on EW-SRBC/$R \times 100$ (using G(0.1,0.1) on APW-ACE data)

ensemble size) relative to the ones obtained with the presented $R = 50000$, as a function of the median of the score changes Δs_R in 100 clustering windows. The plot shows how, for almost all collections, the values of F1 have stabilized by the point the median Δs_R falls below 10^{-10}—the only exception being FAC-LOC, which already starts with $median(\Delta s_1 \ldots \Delta s_{100}) \approx 10^{-9}$, and does not converge until the value reaches 10^{-12} with an ensemble of $R = 2000$ clusterings. This behaviour suggests a replacement of parameter R by a threshold on $median(\Delta s_R)$, and one which gives place to a natural parallelization of the algorithm: we can obtain a batch of weak clusterings of the dataset in parallel, and then use the median of the average score changes produced by them to determine whether the scores have converged. In this direction, Fig. 16c, plots the ensemble size required to achieve this $median(\Delta s_R) < 10^{-10}$ level, as a function of dataset size. The speed of convergence of the scores seems to be proportional to $|\mathcal{X}|^{0.4}$.

To inspect how the use of a convergence criterion affects the runtime of the algorithm, Fig. 17a shows the runtime per clustering (separated in training and testing) of EW-SRBC for each one of the collections in APW-ACE. As we can see, the runtime cost can be fit proportionally to $|\mathcal{X}|^{1.1}$, only slightly above the theoretical linear complexity $O(|\mathcal{X}|)$ we had considered in Sect. 3.6. The fact that the larger datasets we have used also have a higher number of features is likely to be the cause for this quasi-linear behaviour.

Finally, Fig. 17b plots the total runtimes of EW-SRBC up to the point where the median of Δs_R falls below 10^{-10}, against the dataset size. The points are distributed quite closely to a $t \propto |\mathcal{X}|^{1.5}$ curve, as could be expected from the previous two fits. We believe this is certainly another positive result: we have seen how other approaches in the state of the art (e.g.,



(a) Runtime per clustering

(b) Total runtime to convergence $(median(\Delta s_R) < 10^{-10})$

Fig. 17 Runtime for EW-SRBC/$R \times 100$ (using G(0.1,0.1) on APW-ACE data)

DGRADE, AUTOHDS) have computational complexities of $O(|\mathcal{X}|^3)$, which render them unusable for large-scale datasets. Moreover, the fact that EWOCS is easily parallelizable also makes it an attractive option in terms of runtime.

8 Conclusions

In this article, we have considered the problem of minority clustering, contrasting it with regular all-in clustering. We have identified a key limitation of existing minority clustering algorithms—namely, we have seen how the approaches proposed so far for minority clustering are supervised, in the sense that they require the number and distribution of the foreground clusters, as well as the background distribution, as input.

The fact that, in supervised learning and all-in clustering tasks, combination methods have been successfully applied to obtain distribution-free learners, even from the output of weak individual algorithms, has led us to make a three-fold proposal.

First, we have presented a novel ensemble minority clustering algorithm, EWOCS, suitable for weak clustering combination. The properties of EWOCS have been theoretically proved under a set of weak constraints. Second, we have presented two weak clustering algorithms: one, RBC, based on Bregman divergences; and another, RSPLIT, an extension of a previously presented random splitting one. Third, we have proposed an unsupervised procedure to determine the scaling parameters for a Gaussian kernel, used within a minority clustering algorithm.

We have implemented a number of approaches built from the proposed components, and evaluated them on a collection of synthetic datasets, for a comparison to other minority clustering methods in the state of the art. The results of the evaluation show how approaches based on EWOCS, and especially the one built using SRBC as weak clustering algorithm and G(AUTO) as object divergence function, are competitive with respect to—and even outperform—other minority clustering approaches in the state of the art, in terms of F1 and AUC measures of the obtained clusterings.

The completely unsupervised minority clustering approach, built from EWOCS, SRBC, G(AUTO) and an unsupervised threshold detection criterion (one of DIST, NGAUSS+VAR or 2GAUSS) already outperforms other supervised minority clustering approaches. With only the minor supervision introduced by replacing the threshold detection by NGAUSS+BEST,

the resulting approach outperforms all other considered systems, including the much more supervised kMD.

The results on synthetic data have been corroborated with an evaluation on real-world data. A first dataset—more specifically, a subset of the classical text classification Reuters corpus—has allowed us to study the influence of the clustering ensemble size on the results achieved by EWOCS. Specifically, we have found larger ensembles to boost the accuracy of the unsupervised threshold detection criteria. The completely unsupervised minority clustering approach built from EWOCS, SRBC, EUC and 2GAUSS obtains a performance within hundredths of the upper bound of EWOCS.

Additionally, the approach has been applied to a collection of datasets coming from unsupervised relation detection problems of an even larger scale. In that task, the use of EWOCS after a reduction of the problem to minority clustering allows the detection of pairs of related entities with more accuracy than using an approach specifically tailored to relation detection. Moreover, the fact that EWOCS builds a clustering model allows the detection of related entities in new documents not available at clustering time.

At the light of the results, we believe that the EWOCS algorithm is an effective method for ensemble minority clustering, and that it allows the building of competitive and unsupervised approaches to the task. It is appealing because of its simplicity, flexibility and theoretical well-foundedness, and can hence be taken into account for clustering on a diversity of domains, where unsupervised minority clustering tasks may be the rule and not the exception.

Acknowledgements This work has been funded by the KNOW2 (TIN2009-14715-C04-04) and SKATER (TIN2012-38584-C06-01) projects.

References

ACE (2008). The ACE 2008 (ACE08) evaluation plan. http://www.itl.nist.gov/iad/mig/tests/ace/2008/doc/.

Ando, S. (2007). Clustering needles in a haystack: an information theoretic analysis of minority and outlier detection. In *7th IEEE international conference on data mining (ICDM)* (pp. 13–22).

Ando, S., & Suzuki, E. (2006). An information theoretic approach to detection of minority subsets in database. In *6th IEEE international conference on data mining (ICDM)* (pp. 11–20).

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2005). Clustering with Bregman divergences. *Journal of Machine Learning Research*, *6*, 1705–1749.

Banfield, J. D., & Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, *49*(3), 803–821.

Bergmann, B., & Hommel, G. (1988). Improvements of general multiple test procedures for redundant systems of hypotheses. In P. Bauer, G. Hommel, & E. Sonnemann (Eds.), *Multiple Hypothesenprüfung—multiple hypotheses testing* (pp. 100–115). Berlin: Springer.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum Press.

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(7), 719–725.

Cancedda, N., Gaussier, E., Goutte, C., & Renders, J. M. (2003). Word sequence kernels. *Journal of Machine Learning Research*, *3*, 1059–1082.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: a survey. *ACM Computing Surveys*, *41*, 15:1–58.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(8), 790–799.

Collins, M., & Duffy, N. (2002). New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In *40th annual meeting of the association for computational linguistics (ACL)* (pp. 263–270).

Crammer, K., & Chechik, G. (2004). A needle in a haystack: local one-class optimization. In *21st international conference on machine learning (ICML)* (pp. 26–33).

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, *2*, 265–292.

Crammer, K., Talukdar, P. P., & Pereira, F. C. (2008). A rate-distortion one-class model and its applications to clustering. In *25th international conference on machine learning (ICML)* (pp. 184–191).

Davé, R. N. (1991). Characterization and detection of noise in clustering. *Pattern Recognition Letters*, *12*(11), 657–664.

Davé, R. N., & Krishnapuram, R. (1997). Robust clustering methods: a unified view. *IEEE Transactions on Fuzzy Systems*, *5*(2).

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *23rd international conference on machine learning (ICML)* (pp. 233–240).

Dean, J., & Ghemawat, S. (2004). Mapreduce: simplified data processing on large clusters. In *6th symposium on operating system design and implementation*.

Deer, P. J., & Eklund, P. (2003). A study of parameter values for a Mahalanobis distance fuzzy classifier. *Fuzzy Sets and Systems*, *137*, 191–213.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Royal Statistical Society, Series B*, *39*(1).

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Dubes, R., & Jain, A. K. (1980). Clustering methodologies in exploratory data analysis. In M. C. Yovits (Ed.), *Advances in computers* (Vol. 19, pp. 113–228). Amsterdam: Elsevier.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *2nd ACM SIGKDD international conference on knowledge discovery and data mining (KDD)* (pp. 226–231).

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Fraley, C., & Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *Computer Journal*, *41*(8), 578–588.

Freund, Y., & Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, *37*(3), 277–296.

Fukunaga, K., & Hostetler, L. D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, *21*(1), 32–40.

García, S., & Herrera, F. (2008). An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, *9*, 2677–2694.

Ghosh, J., & Gupta, G. (2011). Bregman bubble clustering: a robust framework for mining dense clusters. In D. Holmes & L. C. Jain (Eds.), *Data mining: foundations and intelligent paradigms*. Berlin: Springer.

Ghosh, J., Strehl, A., & Merugu, S. (2002). A consensus framework for integrating distributed clusterings under limited knowledge sharing. In *NSF workshop on next generation data mining* (pp. 99–108).

Gionis, A., Mannila, H., & Tsaparas, P. (2005). Clustering aggregation. In *21st IEEE international conference on data engineering (ICDE)* (pp. 341–352).

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, *13*(3), 780–784.

Gonzàlez, E. (2012). *Unsupervised learning of relation detection patterns*. PhD thesis, Department de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

Gonzàlez, E., & Turmo, J. (2009). Unsupervised relation extraction by massive clustering. In *9th IEEE international conference on data mining (ICDM)* (pp. 782–787).

Graff, D. (2002). *The AQUAINT corpus of English news text* (Tech. Rep. LDC2002T31). Linguistic Data Consortium.

Guillemaud, R., & Brady, M. (1997). Estimating the bias field of MR images. *IEEE Transactions on Medical Imaging*, *16*(3), 238–251.

Gupta, G., & Ghosh, J. (2005). Robust one-class clustering using hybrid global and local search. In *22nd international conference on machine learning (ICML)* (pp. 273–280).

Gupta, G., & Ghosh, J. (2006). Bregman bubble clustering: a robust, scalable framework for locating multiple, dense regions in data. In *6th IEEE international conference on data mining (ICDM)* (pp. 232–243).

Gupta, G., Liu, A., & Ghosh, J. (2010). Automated hierarchical density shaving: a robust automated clustering and visualization framework for large biological data sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *7*(2), 223–237.

Hassan, H., Hassan, A., & Emam, O. (2006). Unsupervised information extraction approach using graph mutual reinforcement. In *Conference on empirical methods in natural language processing (EMNLP)*.

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*, 85–126.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, *31*(3), 264–323.

Kaufman, L., & Rousseeuw, P. J. (2005). *Finding groups in data: an introduction to cluster analysis*. New York: Wiley.

Klir, G. J., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic: theory and applications*. New York: Prentice Hall.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *5th Berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Moya, M. M., Koch, M. W., & Hostetler, L. D. (1993). One-class classifier networks for target recognition applications. In *World congress on neural networks* (pp. 797–801).

Okeke, F., & Karnieli, A. (2006). Linear mixture model approach for selecting fuzzy exponent value in fuzzy c-Means algorithm. *Ecological Informatics*, *1*, 117–124.

Peel, D., & McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, *10*, 339–348.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*(7), 1443–1471.

Schwämmle, V., & Jensen, O. N. (2010). A simple and fast method to determine the parameters for fuzzy c-Means cluster analysis. *Bioinformatics*, *26*(22), 2841–2848.

Schwartz, G. E. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge: Cambridge University Press.

Slonim, N., & Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In *23rd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 208–215).

Spärck-Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*, 11–21.

Strehl, A., & Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, *3*, 583–617.

Tan, P. N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining*. Reading: Addison-Wesley.

Tax, D. M., & Duin, R. P. (2004). Support vector data description. *Machine Learning*, *54*(1), 45–66.

Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. In *37th Allerton conference on communication, control, and computing*.

Topchy, A., Jain, A. K., & Punch, W. (2003). Combining multiple weak clusterings. In *3rd IEEE international conference on data mining (ICDM)* (pp. 331–338).

Topchy, A., Jain, A. K., & Punch, W. (2004). A mixture model for clustering ensembles. In *SIAM international conference on data mining (SDM)* (pp. 379–390).

Topchy, A., Jain, A. K., & Punch, W. (2005). Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(12), 1866–1881.

Wishart, D. (1969). Mode analysis: a generalization of nearest neighbour which reduces chaining effects. In *Colloquium on numerical taxonomy* (pp. 282–308).

Xu, R., & Wunsch, D. C. II (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*(3), 645–678.

Yu, J., Cheng, Q., & Huang, H. (2004). Analysis of the weighting exponent in the FCM. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *34*.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. In *ACM SIGMOD international conference on management of data* (pp. 103–114).

Zhao, Y., & Karypis, G. (2004). Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, *55*, 311–331.