Pattern Recognition Letters 31 (2010) 2415–2424

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Clustering ensembles and space discretization – A new regard toward diversity and consensus

Jugurta Montalvão [a,*], Jânio Canuto [b]

[a] Universidade Federal de Sergipe (UFS), São Cristóvão, CEP. 49100-000, Brazil
[b] Universidade Estadual de Campinas (UNICAMP), Campinas-SP, CEP. 13083-852, Brazil

ARTICLE INFO

Article history:
Received 2 July 2009
Available online 4 August 2010
Communicated by A. Fred

Keywords:
Clustering ensembles
Weak partitions
ANMI criterion
Binary morphology

ABSTRACT

In recent years, the cluster ensembles have been successfully used to tackle well known drawbacks of individual clustering algorithms. Beyond the expected improvement provided by the averaging effect of many clustering algorithms (clustering committee) aiming at the same goal, some interesting experimental results also show that even committees of completely *random* partitions may lead to a useful consensus. Another powerful finding in cluster ensemble research is that the *blind* criterion Averaged Normalized Mutual Information seems to replace actual misclassification ratio, whenever labels are given to actual clusters. In this work, we study what is behind these interesting results and the *blind* criterion, and we use what we learn from this study to propose a new point of view for analysis and design of clustering committees. The usefulness of this new perspective is illustrated through experimental results.

1. Introduction

Clustering is a self-organized process that plays an important role in a wide range of fields, ranging from typical applications, as Pattern Recognition, Signal Compression (Duda et al., 2001), and Knowledge Discovery in Databases (Mitra et al., 2002), to less common ones such as communication channel estimation and/or equalization (Montalvão et al., 2002). Roughly speaking, clustering is aimed at partitioning a set of objects into groups that share some kind of (predefined) similarity. Clearly, it is not a well-posed problem.

In recent years, the success of ensemble methods for supervised learning has motivated the development of ensemble methods for unsupervised learning (Fred and Jain, 2005). At first glace, clustering ensembles just generate multiple partitions of a data set by taking multiple looks at it. Thus, by combining the resulting partitions, one can obtain a good data partitioning even when the clusters are not compact and well separated. Indeed, according to Strehl and Ghosh (2002), an ensemble is a multi-learner system in which each learner tries to solve the very same task, and its main goal is to improve overall accuracy and robustness in doing this task. This point of view links the clustering ensemble to the Connectionist paradigm (Feldman and Ballard, 1982) and its redundant structures.

However, far beyond the expected improvement provided by the averaging effect of many redundant clustering algorithms (clustering committee), some interesting experimental results also show that even committees of completely *random* partitions (Fern and Brodley, 2003; Topchy et al., 2005) may lead to useful consensus. Another surprising finding is that, if we assume that data from each cluster can be labeled (one label per cluster), and this label is seen as a hidden parameter to be found by clustering analysis, the *blind*[1] criterion Averaged Normalized Mutual Information (ANMI) seems to replace the supervised label misclassification ratio (Strehl and Ghosh, 2002).

In this work, we study what is behind these interesting results and we reformulate the *blind* ANMI criterion into a Lagrangian function. What we learn from this study is used to propose a new point of view for analysis and design of clustering committees.

According to this proposed new point of view, clustering ensembles play the role of an irregular space quantizer, thus mapping patterns in a new nonlinearly transformed space where clusters are likely to be more compact and well separated. In this new space, consensus finder plays the role of a conventional clustering algorithm, although it uses distances between vectors of labels, instead of continuous distances. Through this new perspective, we also try to provide new understandings of how clustering ensembles works. For instance, we show through simple reasoning and some experiments how the paradigm from which committee members come from affects clustering results.

* Corresponding author. Tel.: +55 79 32 23 17 36; fax: +55 79 21 05 66 84.
E-mail addresses: jmontalvao@ufs.br, jugurta.montalvao@infonet.com.br (J. Montalvão), j079927@dac.unicamp.br (J. Canuto).

[1] Blind in the sense that actual labels are unknown.

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.07.018

Moreover, we can also see clearly how diversity amongst committee members is necessary to provide useful space discretization grids. The usefulness of this new perspective is illustrated through experimental results.

Unfortunately, clustering ensembles is a wide field of research and some relevant perspectives are not addressed in this work. That is the case, for instance, of works where pattern spaces are directly transformed, such as in (Al-Razgan and Domeniconi, 2006), where to cope with the high-dimensionality of data, a soft feature selection procedure is proposed, the Locally Adaptive Clustering (LAC). Indeed, the LAC, along with the two proposed consensus algorithms (i.e. the Weighted Similarity Partitioning Algorithm and the Weighted Bipartite Partitioning Algorithm) are related to space transformations, which, in some ways, would link them to the point of view proposed here. Although, for conciseness sake, in this first presentation of a new perspective, we only discuss ensembles of clustering algorithms working on original or affine transformed pattern space.

However, we highlight that the concept of pattern space quantization presented here naturally encompasses the three main approaches mentioned in (Gullo et al., 2009), namely instance-based clustering ensembles, cluster-based clustering ensembles, and hybrid clustering ensembles. Indeed, most instance-based methods operate on the co-occurrence or co-association matrix, which are equivalent to Hamming distances between patterns projected in label space, whereas cluster-based clustering ensembles directly address projected pattern in label space.

In Section 2, we formalize some theoretical tools for clustering ensemble analysis, whereas in Section 3, we analyse the ANMI criterion and we compare it to its non-normalized version, whose optimization is also discussed. In Section 4, the clustering ensemble consensus is seen as a data space deformation, through a flexible space quantization grid, whose effects on clustering tasks are studied in Section 5, and whose consequences concerning ensemble diversity is discussed in Section 6. Finally, in Section 7, we summarize the proposed new perspective for clustering ensembles, along with its useful consequences.

2. Clustering ensemble formalization

We assume that N objects or patterns, $N \in \mathcal{N}^*$, are available. Therefore, they form a nonempty set, denoted hereafter as $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$. Moreover, each clustering attempt maps each pattern from $\mathcal{X}$ into a label. Accordingly, a label put in the ith object, by the jth clustering attempt, is denoted as $y_{i,j}$.

For convenience, we gather all labels, and therefore the entire clustering ensemble, into a matrix $\boldsymbol{Y} = \{y_{i,j}\}$, referred to as *the labeling committee*, where $i (i = 1,2,\ldots,N)$ represents the number of objects/patterns to be clustered, and $j (j = 1,2,\ldots,C)$ represents the number of available clustering attempts.

2.1. The entropy associated to a vector of labels

In Section 3, we are going to analyze consensus clustering criteria based on entropy. Thus, in order to properly define the entropy of a clustering outcome, we first associate probabilities to vectors of labels, as follows: let $\boldsymbol{y}^{(c)}$ be the cth column of $\boldsymbol{Y}$, whereas $K^{(c)}$ is the number of distinct labels in $\boldsymbol{y}^{(c)}$. If we randomly pick an element from $\mathcal{X}, x_n$, the probability of each element being picked is $1/N$, and its corresponding label, $\boldsymbol{y}^{(c)}(n)$, can be regarded as an instance of a random source of symbols, $Y^{(c)}$, with $K^{(c)}$ possible outcomes. Furthermore, the probability of label $l_k^{(c)}$ being chosen is $p_k^{(c)} = n_k^{(c)}/N$, where $n_k^{(c)}$ stands for the number of labels $l_k^{(c)}$ in $\boldsymbol{y}^{(c)}$. Thus, we can define the entropy associated to $\boldsymbol{y}^{(c)}$ as:

$$H(Y^{(c)}) = -\sum_{k=1}^{K^{(c)}} p_k^{(c)} log\left(p_k^{(c)}\right)$$

Alternatively, by stretching our notation (for convenience), we use both $H(\boldsymbol{p}^{(c)})$ or $H(\boldsymbol{y}^{(c)})$ indistinctly, instead of $H(Y^{(c)})$, where $\boldsymbol{p}^{(c)} = \left[p_1^{(c)}, p_2^{(c)}, \ldots, p_{K^{(c)}}^{(c)}\right]$ is a vector of probabilities.

When we consider two columns of $\boldsymbol{Y}$, $\boldsymbol{y}^{(c)}$ and $\boldsymbol{y}^{(d)}$, there is then a pair of labels associated to each object x_n, $(l^{(c)}, l^{(d)})$. This pair of labels play the role of a single label, say $l^{(c,d)}$, whose probability of being picked, if objects are randomly drawn, is $p_k^{(c,d)} = n_k^{(c,d)}/N$, where $n_k^{(c,d)}$ is the number of simultaneous occurrences of labels $l^{(c)}$ and $l^{(d)}$ through the N objects. Thus, we can also define the joint entropy associated to two columns of $\boldsymbol{Y}$ as:

$$H(\boldsymbol{p}^{(c,d)}) = -\sum_{k=1}^{K^{(c,d)}} p_k^{(c,d)} log\left(p_k^{(c,d)}\right)$$

where $K^{(c,d)}$ is the number of distinct pairs of labels found when columns $\boldsymbol{y}^{(c)}$ and $\boldsymbol{y}^{(d)}$ are put together. It is worth noting that $H(\boldsymbol{p}^{(c,d)})$ is the *joint entropy* of two random sources of symbols, each one associated to each vector of labels.

Therefore, the pairwise mutual information associated to columns $\boldsymbol{y}^{(c)}$ and $\boldsymbol{y}^{(d)}$ is:

$$I(\boldsymbol{p}^{(c)}, \boldsymbol{p}^{(d)}) = H(\boldsymbol{p}^{(c)}) + H(\boldsymbol{p}^{(d)}) - H(\boldsymbol{p}^{(c,d)})$$

A useful property of entropy is its recursivity (or decomposability) (MacKay, 2003), which we briefly present here as a lemma, since it will play an important role throughout this work.

Lemma 2.1. *Let y be a vector of labels given to N objects, where K ($K \leqslant N$) is the number of distinct labels, $l_1, l_2, \ldots, l_K$, and $n_1, n_2, \ldots, n_K$ stand for the number of objects under each label. We further denote $p_1, p_2, \ldots, p_K$ as being the probability of randomly picking one label. If we subpartition the set of n_a objects, labeled with l_a, into Q subsets, with $s_{a,q}n_a$ objects in each subset (where $\sum_{q=1}^{Q} s_{a,q} = 1$), then the new vector of labels, $\check{\boldsymbol{y}}$, has an entropy increased by $p_a H(s_a)$, or, equivalently, $H(\check{\boldsymbol{p}}) = H(\boldsymbol{p}) + p_a H(\boldsymbol{s}_a)$, where $H(\boldsymbol{s}_a) = -\sum_{q=1}^{Q} s_{a,q} log(s_{a,q})$ is the entropy of the probability vector $s_a = [s_{a,1}, \ldots, s_{a,Q}]$.*

Proof. The entropy of the subpartitioned vector of labels, $\check{\boldsymbol{y}}$, can be split into two sums: one for the subpartition with Q new labels, and another for the remaining $K - 1$ labels, as follows:

$$H(\check{\boldsymbol{p}}) = -\left[\sum_{\substack{k=1 \\ k \neq p}}^{K} \frac{n_k}{N} log\left(\frac{n_k}{N}\right) + \sum_{q=1}^{Q} \frac{s_{a,q}n_p}{N} log\left(\frac{s_{a,q}n_p}{N}\right)\right]$$

Since N and n_p do not depend on q, the subpartition related term can be split into two ones:

$$H(\check{\boldsymbol{p}}) = -\left[\sum_{\substack{k=1 \\ k \neq p}}^{K} \frac{n_k}{N} log\left(\frac{n_k}{N}\right) + \frac{n_p}{N} \sum_{q=1}^{Q} s_{a,q}\left(log(s_{a,q}) + log\left(\frac{n_p}{N}\right)\right)\right]$$

and

$$H(\check{\boldsymbol{p}}) = -\left[\sum_{\substack{k=1 \\ k \neq p}}^{K} \frac{n_k}{N} log\left(\frac{n_k}{N}\right) + \frac{n_p}{N} log\left(\frac{n_p}{N}\right) + \frac{n_p}{N} \sum_{q=1}^{Q} s_{a,q} log(s_{a,q})\right]$$

Finally, the entropy of p and that of the subpartition can be separated, and the former is multiplied by the probability associated to the label whose subset was subpartitioned:

$$H(\check{\boldsymbol{p}}) = H(\boldsymbol{p}) - \frac{n_p}{N}\sum_{q=1}^{Q} s_{a,q}\log(s_{a,q}) = H(\boldsymbol{p}) + p_a H(\boldsymbol{s}_a) \qquad \square$$

Remark 1. Any nontrivial subpartition (i.e. $s_{a,q} \neq 1,\ \forall q$) increases the entropy associated to the original vector of labels by an amount proportional to the cardinality of the subpartitioned set, n_p. Moreover, this increment is maximized for equally sized subpartitions.

Remark 2. In the opposite direction, we readily realize that if two or more subsets of objects, under two different labels are merged under a single label, the resulting vector of labels, $\hat{\boldsymbol{y}}$, has its entropy reduced by an amount which is proportional to the summed probability of the merged subsets, and this reduction is maximized when merged labels have equal probabilities.

Fig. 1 illustrates both subpartition and fusion effects.

Example. Let $y = [A\ A\ A\ A\ B\ B\ A\ A\ C\ C\ C\ B]^t$, whose entropy is $H(\boldsymbol{p}) = \frac{6}{12}\log(2) + \frac{3}{12}\log(4) + \frac{3}{12}\log(4) = 1.5$ *bits*. If we split the set of labels '*A*' into two equally sized new sets, we have:

$$\check{\boldsymbol{y}}_1 = [A1\ A1\ A1\ A2\ B\ B\ A\ 2\ A\ 2\ C\ C\ C\ B]^t$$

and the new entropy is

$$H(\check{\boldsymbol{p}}_1) = \frac{3}{12}\log(4) + \frac{3}{12}\log(4) + \frac{3}{12}\log(4) + \frac{3}{12}\log(4) = 2\ bits$$

By contrast, if we split that group into unbalanced subsets, say

$$\check{\boldsymbol{y}}_2 = [A1\ A2\ A2\ A2\ B\ B\ A2\ A2\ C\ C\ C\ B]^t$$

the corresponding entropy is smaller:

$$H(\check{\boldsymbol{p}}_2) = \frac{1}{12}\log(12) + \frac{5}{12}\log(12/5) + \frac{3}{12}\log(4) + \frac{3}{12}\log(4) \approx 1.82\ bits$$

And we note that, as expected,

$$H(\check{\boldsymbol{p}}_1) = H(\boldsymbol{p}) + \underbrace{(6/12)[(1/2)\log(2) + (1/2)\log(2)]}_{\text{entropy increment}}$$

and

$$H(\check{\boldsymbol{p}}_2) = H(\boldsymbol{p}) + \underbrace{(6/12)[(1/6)\log(6) + (5/6)\log(6/5)]}_{\text{entropy increment}}$$

On the other hand, if we merge labels '*B*' and '*C*', the new vector of labels, $\hat{\boldsymbol{y}} = [A\ A\ A\ A\ BC\ BC\ A\ A\ BC\ BC\ BC\ BC]^t$, has a reduced entropy given by $H(\hat{\boldsymbol{p}}) = \frac{6}{12}\log(12/6) + \frac{6}{12}\log(12/6) = 1\ bit$, or, according to Remark 2:

$$H(\hat{\boldsymbol{p}}) = H(\boldsymbol{p}) - \underbrace{\frac{(3+3)}{12}[\log(6/3) + \log(6/3)]}_{\text{entropy decrement}}$$

Definition 1. Two label vectors, say $y^{(c)}$ and $y^{(d)}$, are said *equivalent* if they correspond to the same partitions of $\mathcal{X}$. An important consequence is that $H(p^{(c)}) = H(p^{(d)}) = H(p^{(c,d)})$. Hereafter, we denote equivalence between $y^{(c)}$ and $y^{(d)}$ as $y^{(c)} \equiv y^{(d)}$.

Definition 2. If $H(p^{(c)}) = H(p^{(c,d)})$ but $H(p^{(c)}) > H(p^{(d)})$, then the partition provided by $y^{(c)}$ contains that of $y^{(d)}$. This is compactly represented as $y^{(c)} = \text{subpart}\,(y^{(d)})$.

Example. Vectors $y^{(1)} = [A\ A\ A\ A\ B\ B\ A\ A]^t$ and $y^{(2)} = [\beta\ \beta\ \beta\ \beta\ \alpha\ \alpha\ \beta\ \beta]^t$ are equivalent (i.e. $y^{(1)} \equiv y^{(2)}$), whereas $y^{(3)} = \text{subpart}\,(y^{(1)})$ and $y^{(3)} = \text{subpart}\,(y^{(2)})$, for $y^{(3)} = [1\ 1\ 1\ 1\ 2\ 2\ 3\ 3]^t$.

Definition 3. Concatenation between vectors of labels is defined as follows: given two vectors of N labels, $y^{(c)}$ and $y^{(d)}$, such as, for the nth data/pattern in $\mathcal{X}$, $y^{(a)}(n) = A$ and $y^{(b)}(n) = B$ (where A and B are labels), then the concatenation operation is defined as $c = y^{(c)} \oplus y^{(d)}$, such as $c(n) = AB$ is the new concatenated label of the nth object. Fig. 2 illustrates that label vectors concatenation is equivalent to the merging of data set partitions.

Furthermore, it is a straightforward matter to verify that, if $\boldsymbol{y}^{(c)} = \text{subpart}\,(\boldsymbol{y}^{(d)})$, then:

$$H(\boldsymbol{y} \oplus \text{subpart}(\boldsymbol{y})) = H(\text{subpart}(\boldsymbol{y}))$$

where subpart $(\boldsymbol{y})$ stands for any subpartitioning of $\boldsymbol{y}$. Finally, since the labeling committee Y maps data patterns, from pattern spaces, into vector of labels, its is useful to assume that:

Definition 4. Vectors of labels lie in a metric space (i.e. a set with a metric), hereafter referred to as *label space*, whose metric is the Hamming distance (Duda et al., 2001) between two vectors of labels, which equals the number of positions for which the corresponding labels are different.

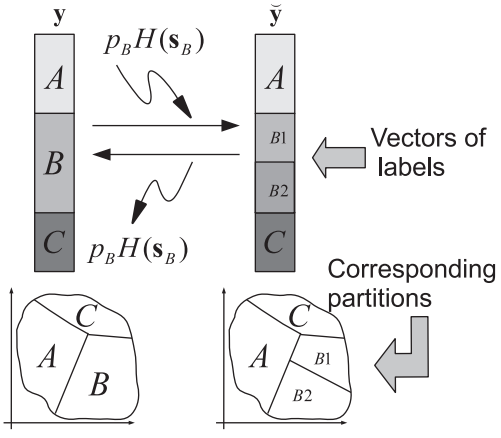

Fig. 1. Entropy increment/decrement due to cell subpartitioning/merging.

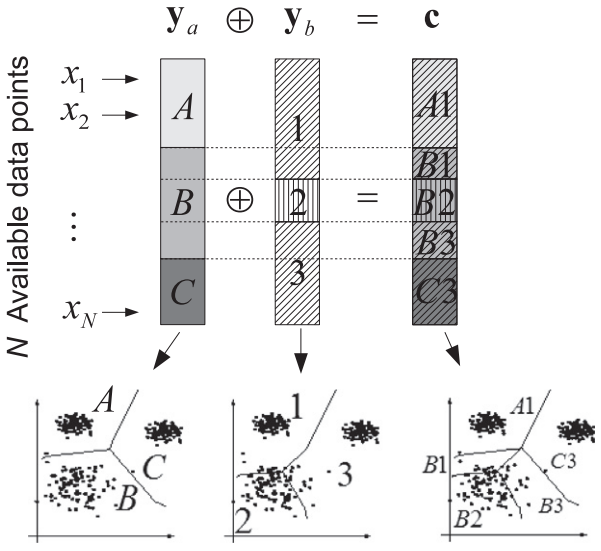

Fig. 2. Concatenation of label vectors as a partition combination.

3. Consensus criteria analysis

Each column of the labeling committee, $\boldsymbol{Y}$, defines a partition of $\mathcal{X}$. Thus, consensus can be regarded as finding a single partition which attains the maximum agreement amongst all partitions/columns of $\boldsymbol{Y}$. Clearly, this consensus is a vector of labels, itself, denoted as $\boldsymbol{c}$.

In literature, one consensus criteria seems to be more prominent now, the *Averaged Normalized Mutual Information* (ANMI) (Strehl and Ghosh, 2002), which is sometimes directly used in consensus optimization, such as in the Voting Active Clusters (VAC) approach, proposed by Tumer and Agogino (2008), and in the greedy optimization approach, the most straightforward algorithm proposed in (Strehl and Ghosh, 2002). In this Section, we analyse this criterion to better understand how it works. First, however, for a proper approach into the matter, we start by defining a similar but simpler criterion: the Averaged Mutual Information (AMI), which differs from the ANMI only by neglecting normalization.

3.1. Averaged mutual information

This very simple criterion is given by:

$$J^{\mathrm{AMI}}(\boldsymbol{c};\boldsymbol{Y}) = (1/C)\sum_{k=1}^{C} I(\boldsymbol{c},\boldsymbol{y}^{(k)}) \tag{1}$$

where $I(\boldsymbol{c},\boldsymbol{y}^{(k)}) = H(\boldsymbol{c}) + H(\boldsymbol{y}^{(k)}) - H(\boldsymbol{c} \oplus \boldsymbol{y}^{(k)})$.

Thanks to its simplicity, we are able to state the following theorem

Theorem 3.1. *The maximum AMI is given by* $J^{AMI}_{\max} = (1/C)\sum_{k=1}^{C} H(\boldsymbol{y}^{(k)})$*, and it is provided by a consensus,* $\boldsymbol{c}_0^{AMI}$*, resulting from the concatenation of all columns of Y, i.e.* $\boldsymbol{c}_0^{AMI} = \boldsymbol{y}^{(1)} \oplus \boldsymbol{y}^{(2)} \oplus \ldots \oplus \boldsymbol{y}^{(C)}$*, or any subpartitioning of* $\boldsymbol{c}_0^{AMI}$*.*

Proof. The mutual information between the consensus and the kth column of Y is given by:

$$I(\boldsymbol{c}_0^{\mathrm{AMI}},\boldsymbol{y}^{(k)}) = H(\boldsymbol{c}_0^{\mathrm{AMI}}) + H(\boldsymbol{y}^{(k)}) - H(\boldsymbol{c}_0^{\mathrm{AMI}} \oplus \boldsymbol{y}^{(k)}) \tag{2}$$

Since $\boldsymbol{c}_0^{\mathrm{AMI}} = \boldsymbol{y}^{(1)} \oplus \boldsymbol{y}^{(2)} \oplus \cdots \oplus \boldsymbol{y}^{(C)}$, the concatenation of $\boldsymbol{c}_0^{\mathrm{AMI}}$ and any column of Y produces a vector of labels equivalent to $\boldsymbol{c}_0^{\mathrm{AMI}}$ (see Definitions 1 and 3, in Section 2.1), and $H(\boldsymbol{c}_0^{\mathrm{AMI}} \oplus \boldsymbol{y}^{(k)}) = H(\boldsymbol{c}_0^{\mathrm{AMI}})$. Therefore, Eq. (2) can be simplified to

$$I(\boldsymbol{c}_0^{\mathrm{AMI}},\boldsymbol{y}^{(k)}) = H(\boldsymbol{y}^{(k)})$$

Consequently, the AMI equals $(1/C)\sum_{k=1}^{C} H(\boldsymbol{y}^{(k)})$.

On the other hand, for any subpartitioning of $\boldsymbol{c}_0^{\mathrm{AMI}}$, say $\boldsymbol{c}_s = \mathrm{subpart}(\boldsymbol{c}_0^{\mathrm{AMI}})$, since *it contains* $\boldsymbol{c}_0^{\mathrm{AMI}}$, (see Definition 2, in Section 2.1) which, in turn, contains $y^{(k)}$, for any $1 \leqslant k \leqslant C$, then we also have $H(c_s \oplus y^{(k)}) = H(c_s)$, and the AMI is again simplified, through Eqs. (2) and (1), to $J^{\mathrm{AMI}}(\boldsymbol{c}_s,\boldsymbol{Y}) = \sum_{k=1}^{C} H(\boldsymbol{y}^{(k)})$.

Finally, for a consensus, c, nor equivalent to $\boldsymbol{c}_0^{\mathrm{AMI}}$, neither to any subpartitioning of it, it is clear that at least one concatenation $y^{(k)} \oplus c$ must produce a vector equivalent to a subpartitioning of c (otherwise c would be equivalent to $\boldsymbol{c}_0^{\mathrm{AMI}}$). Therefore, according to Lemma 2.1, for at least one k, $H(y^{(k)} \oplus c) > H(c)$, and, from Eq. (2), we obtain $I(y^{(k)},c) < H(y^{(k)})$. Consequently,

$$J^{\mathrm{AMI}}(\boldsymbol{c}) = \frac{1}{C}\sum_{k=1}^{C} I(\boldsymbol{c},\boldsymbol{y}^{(k)}) < \frac{1}{C}\sum_{k=1}^{C} H(\boldsymbol{y}^{(k)})$$

and, given that $\frac{1}{C}\sum_{k=1}^{C} H(\boldsymbol{y}^{(k)})$ is the value of J^{AMI}, for $\boldsymbol{c}_0^{\mathrm{AMI}}$ or any subpartitioning of it, we conclude that the AMI criterion is maximized to $\frac{1}{C}\sum_{k=1}^{C} H(\boldsymbol{y}^{(k)})$ for any vector equivalent to $\boldsymbol{c}_0^{\mathrm{AMI}}$, or equivalent to any subpartitioning of it. □

Probably, the first consequence of this theorem is that the number of distinct labels in $\boldsymbol{c}_0^{\mathrm{AMI}}$ is not flexible, and should not be used as a reasonable estimate to the number of clusters. For instance, in Fig. 2, where we clearly see three clusters, the concatenation of only two vector of labels, which is equivalent to the merging of two partitions of dataset $\mathcal{X}$, as illustrated in the same Figure, produces a consensus with 5 cells. Evidently, even if such a consensus is optimum in terms of AMI, it is almost useless for cluster analysis purposes, because it may divide actual clusters into separated cells.

Indeed, in cluster analysis, either we know beforehand the number of clusters, or we look for an estimate of it, usually through criteria that somehow compare intra-cluster compactness and between-cluster sparseness. However, if we limit our clustering approach to that where we only have access to vectors of labels (and no access to original data), then we must compensate for it through two strategies, namely:

(I) By indirectly measuring distances between patterns through their labels.
(II) By penalizing consensus with an elevated number of labels, thus inducing the concatenation of cells in the partition corresponding to $\boldsymbol{c}_0^{\mathrm{AMI}}$.

3.2. A parsimoniousness role for normalization of mutual information

In this subsection, we explain how the normalization of mutual information can be regarded as a strategy of type II. Then we discuss how strategies of type I may explain the good performances of clustering ensembles.

The Averaged Normalized Mutual Information (ANMI), as defined in (Strehl and Ghosh, 2002), is given by:

$$J^{\mathrm{ANMI}}(\boldsymbol{c};\boldsymbol{Y}) = \frac{1}{C}\sum_{k=1}^{C} \frac{I(\boldsymbol{c},\boldsymbol{y}^{(k)})}{\sqrt{H(\boldsymbol{c})H(\boldsymbol{y}^{(k)})}}$$

or

$$J^{\mathrm{ANMI}}(\boldsymbol{c};\boldsymbol{Y}) = \frac{1}{C}\sum_{k=1}^{C} \frac{H(\boldsymbol{c}) + H(\boldsymbol{y}^{(k)}) - H(\boldsymbol{c} \oplus \boldsymbol{y}^{(k)})}{\sqrt{H(\boldsymbol{c})H(\boldsymbol{y}^{(k)})}} \tag{3}$$

Since $\boldsymbol{c} \oplus \boldsymbol{y}^{(k)}$ either is equivalent, or contains $\boldsymbol{c}$ (see Definitions 1 and 2), then, according to Theorem 3.1, for any consensus $\boldsymbol{c}$, $H(\boldsymbol{c} \oplus \boldsymbol{y}^{(k)}) \geqslant H(\boldsymbol{c})$, where equality holds only when $\boldsymbol{c} = \boldsymbol{c}_0^{\mathrm{AMI}}$ or $\boldsymbol{c}$ is a subpartitioning of $\boldsymbol{c}_0^{\mathrm{AMI}}$. It is useful to define a positive entropy increment $\Delta^{(k)}$, such as:

$$H(\boldsymbol{c} \oplus \boldsymbol{y}^{(k)}) = H(\boldsymbol{c}) + \Delta^{(k)}, \quad \Delta^{(k)} \geqslant 0$$

where $\Delta^{(k)} = 0$, $\forall k$ only for $\boldsymbol{c} = \boldsymbol{c}_0^{\mathrm{AMI}}$. Thus, we can rewrite Eq. (3) as:

$$J^{\mathrm{ANMI}}(\boldsymbol{c};\boldsymbol{Y}) = \frac{1}{C}\sum_{k=1}^{C} \frac{H(\boldsymbol{y}^{(k)}) - \Delta^{(k)}}{\sqrt{H(\boldsymbol{c})H(\boldsymbol{y}^{(k)})}} \tag{4}$$

By applying logarithm on both sides of Eq. (4), we can re-arrange $\log(J^{\mathrm{ANMI}}(\boldsymbol{c};\boldsymbol{Y}))$ as the sum of two terms:

$$\log(J^{\mathrm{ANMI}}(\boldsymbol{c};\boldsymbol{Y})) = f(\boldsymbol{c};\boldsymbol{Y}) + \lambda g(\boldsymbol{c}) \tag{5}$$

where $\lambda = 1$, $f(\boldsymbol{c};\boldsymbol{Y}) = \log\left(\sum_{k=1}^{C} \frac{H(\boldsymbol{y}^{(k)}) - \Delta^{(k)}}{\sqrt{H(\boldsymbol{y}^{(k)})}}\right)$, and $g(\boldsymbol{c}) = -\log\left(C\sqrt{H(\boldsymbol{c})}\right)$.

This formulation puts into evidence an underlying Lagrangian function (Duda et al., 2001). That is to say that, while the first term $f(\boldsymbol{c};\boldsymbol{Y})$ is maximized for $\Delta^{(k)} = 0$, $\forall k$, the second term penalizes consensus with too much labels, because it is not parameterized by $\boldsymbol{Y}$, and depends on $H(\boldsymbol{c})$. Therefore, the first term is equivalent to the non-normalized criterion AMI and leads to the very same consensus $\boldsymbol{c}_0^{\mathrm{AMI}}$, if optimized alone. By contrast, the second term goes in the opposite direction, and clearly

induces a consensus, $\boldsymbol{c}_0^{\mathrm{ANMI}}$, corresponding to a partition with less cells than that corresponding to $\boldsymbol{c}_0^{\mathrm{AMI}}$.

A question may raise from this proposed point of view: is the *Lagrange multiplier*, $\lambda = 1$, the best choice for any data set? Unfortunately, this seems to be a too complex question to be properly addressed here, and we will postpone it to future works. For a while, we just claim that:

(a) Consensus $\boldsymbol{c}_0^{\mathrm{ANMI}}$ usually produces better clusterings than $\boldsymbol{c}_0^{\mathrm{AMI}}$ because normalization plays a parsimonious role which reduces the number or partition cells.
(b) Consensus $\boldsymbol{c}_0^{\mathrm{ANMI}}$ is not necessarily optimized when the number of partition cells coincides with the number of actual data clusters. A straightforward example is that one where the number of clusters, by chance, equals the number of partition cells produced by $\boldsymbol{c}_0^{\mathrm{AMI}}$ (therefore greater than that by $\boldsymbol{c}_0^{\mathrm{ANMI}}$).
(c) A lesson we learn from the usefulness of the J^{ANMI}, shown through practical examples in literature, in spite of its non-optimized Lagrangian formulation, is that *better consensus are probably reached by concatenation of cells in partitions given by* $\boldsymbol{c}_0^{\mathrm{AMI}}$.
(d) The ANMI criterion is not enough to guide concatenation of partition cells. Indeed, for this purpose, it is necessary to measure distances between data points or, alternatively, between partition cells.
(e) A primary motivation for developing cluster ensembles (Strehl and Ghosh, 2002) is related to data privacy, since it provides new data analysis/segmentations without going back to the original features. In order to keep this motivation on, in this work, we must avoid the use of distances between data points in their original spaces.

Claims (c) and (d) pave the way for most known algorithms. For instance, in (Strehl and Ghosh, 2002), three algorithms are proposed, besides the greedy optimization of J^{ANMI}, namely: the Cluster-based Similarity Partitioning Algorithm (CSPA), the Hyper-Graph Partitioning Algorithm (HGPA), and the Meta-CLustering Algorithm (MCLA). They all indirectly address distances between data points by counting the number of times subsets of points appear in the same cluster. In the Graph Theoretic background used by the authors, coincidence pointers are referred to as hyper-graphs, whereas data points are associated to vertices. Similar indirect distance measures between data points, though in a simpler (not graph based) formulation reappear, for instance, in (Fred and Jain, 2005; Yu et al., 2008; Hong et al., 2009).

In (Topchy et al., 2005; Nguyen and Carauna, 2007) we find a second kind of approach, which defines and uses distances between cells in the final data partition given by $\boldsymbol{c}_0^{\mathrm{AMI}}$. Typically, the Hamming distance is used, even though, in (Topchy et al., 2005), likelihood is used instead. It is worth noting that they use the partition corresponding to $\boldsymbol{c}_0^{\mathrm{AMI}}$ as a starting point for cell recombination, though they do not define the AMI criterion.

4. Clustering ensemble as a (Re)quantization approach

In order to link partition cells and clusters, we highlight that close patterns in the pattern space are likely to be mapped into the same cluster by most clustering committee members. Similarly, closer patterns are likely mapped into closer cells in most data partition consensuses. Thus, the Hamming distance between partition cells, in the label space, indirectly measures distances between patterns inside these cells.

Consequently, regardless which kind of clustering committee is used, clustering ensemble approaches can be regarded as a two-layered strategy, where the consensus finder in the second layer is, itself, a clustering algorithm running in the label space. Fig. 3 illustrates this point of view.

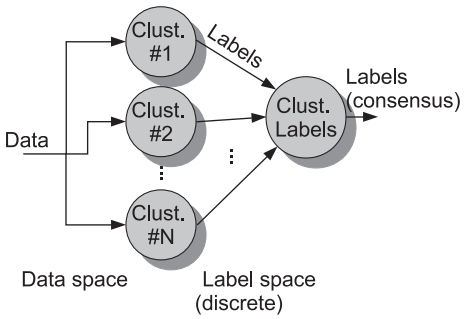

Fig. 3. Clustering Consensus as a network of clustering algorithms – first, in data space, and secondly, in label (discrete) space.

Virtually any conventional clustering approach can be adapted to play the role of a consensus searcher, which is, essentially, a clustering algorithm in the label space. For instance, in (Nguyen and Carauna, 2007), a straightforward adaptation of the *K*-means algorithm was used for finding consensus, under the denomination *Iterative Voting Consensus (IVC)*. Interestingly, in the experimental results presented in (Nguyen and Carauna, 2007), this very simple approach outperformed those proposed in (Strehl and Ghosh, 2002), which are sensibly more elaborated. Many other examples come from more straightforward approaches to finding consensus, where pairwise distances between vectors of labels are arranged into distance matrices, and agglomerative clustering algorithms provide dendrograms through which clusters are analysed. Not surprisingly, the typical drawbacks of agglomerative algorithms are re-found in the label space based adaptations (e.g. the computational and storage complexity of the CSPA approach (Strehl and Ghosh, 2002)).

From this two-layer clustering perspective, one important question arises: why is it better than conventional clustering in data space? Beyond the obvious averaging effect obtained from the re-doing of non-robust cluster analysis many times, there is evidence of other less obvious effects, since the averaging effect alone does not explain, for instance, how combinations of weak clusterings solve difficult problems (Fern and Brodley, 2003; Topchy et al., 2005).

An interesting answer may came from a work published in 1993, by Postaire et al. (1993) (also detailed in Theodoridis and Koutroumbas, 2003), the "Binary Morphology Clustering Algorithm" (BMCA). In the BMCA, data space is first quantized through a regular grid, thus producing a data set partition. Then, morphological operations discard partition cells with too low data densities, whereas high-density cells are fused. In Fig. 4, the Half-rings data set used in (Postaire et al., 1993) is depicted, along with a regular grid, to illustrate the approach. As we can observe, discarding low-density cells is similar to a pruning of possible outliers, whereas the fusion of high-density ones may create clusters with virtually any shape (non-spherical clusters, for instance).

Alternatively, we can regard the regular grid in Fig. 4 as a specific case of weak partition combination, where each line is a committee member, similarly to the random hyperplans used in (Topchy et al., 2005). Thus, grid cells may be labeled and regarded as points in a label space, where distances between cells are measured by the minimum number of boundaries between them (Hamming distance). Now, for instance, if we discard sparse cells which, together, contains 30% of N ($N = 1000$ points, in this case), and if we proceed any cluster analyzes in this new metric space, we can easily infer the existence of two clusters.

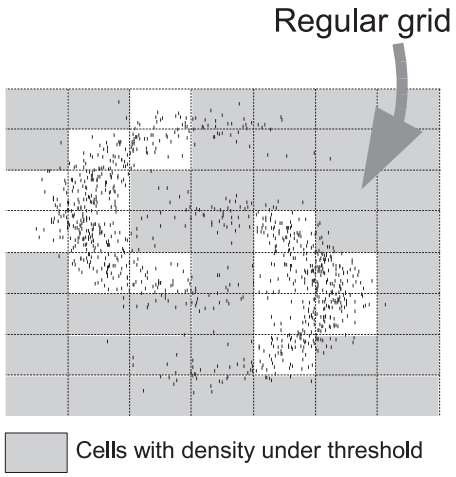

Fig. 4. Regular grid – all cells have the same area.

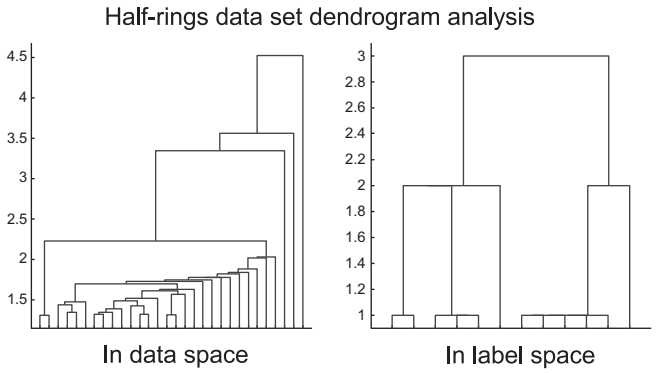

Fig. 5. Single-link dendrograms from the Half-rings data set in data space (left) and in label space (right).

Fig. 5 shows both single-link dendrogram from the Half-rings data set, in the original data space, and in the label space after pruning of sparse cells. As expected, in the label space, the two non-spherical clusters are clearly noticed, whereas, in the original data space, they are not.

To further illustrate the analogy between BMCA and clustering ensembles, we replace the regular grid in Fig. 4 with an irregular one, corresponding to an ensemble of 80 random lines (2-D version of random hyperplans studied in Topchy et al., 2005). Fig. 6 shows one instance of this *random clustering ensemble*, where each (poor) clustering hypothesis randomly splits data in only 2 clusters.

Therefore, each pattern from original 2-D space is mapped to a vector of 80 labels. If we further assume that the labels used in each clustering are '1' and '2', then this association corresponds to a mapping from $\mathbb{R}^2$ to $\{1,2\}^{80}$. In this new 80-D label space, we may use any conventional clustering approach (for instance, by replacing Euclidean distance with Hamming distance in the *K*-Means algorithm) as a consensus finder. Nonetheless, we arbitrarily choose 3 consensus methods proposed in literature specifically for this task,[2] and then we just count the number of pattern misassignments (misclassifications).

Clearly, patterns mapped into the same set of labels lie inside the same cell of the quantization grid. Therefore, we may discard low-density cells, just as we did in the regular-grid based

[2] We dropped the HGPA method because it produced useless results in this experiment: almost 50% of misclassifications.

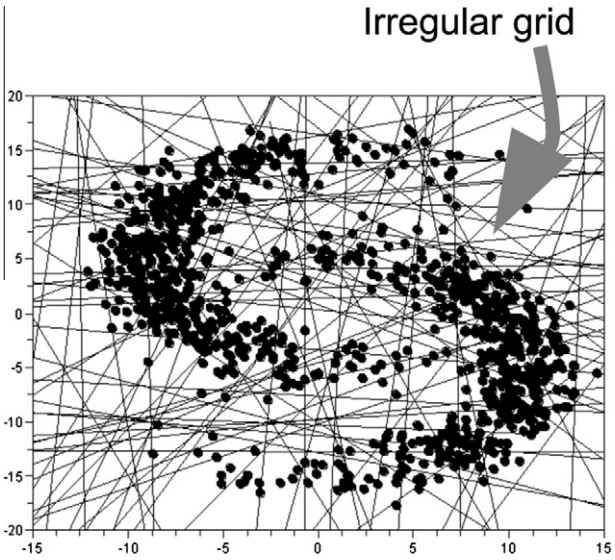

Fig. 6. Irregular grid – cells have randomly distributed areas.

illustration. Thus, we discard cells which, together, contains 30% of *N*, and we proceed cluster analyzes again.

The average percentages of pattern misclassifications, with and without cell pruning, are presented in Table 1, averaged over 10 independent initializations of the random clustering ensemble and consensus methods.

The proportion of misclassifications, through all tested consensus methods is clearly reduced after pruning of cells. In other words, the same advantages of the BMCA concerning pruning are noticed here. Indeed, the replacement of regular grids with random grids may be regarded as a generalization of the space quantization proposed in (Postaire et al., 1993), with the advantage of an easier adaptation to high-dimensional pattern spaces.

This link between the Binary Morphology applied to Clustering and clustering ensemble is not explicitly exploited in clustering ensemble literature – to the extent of authors knowledge –, though it can be a useful development tool. For instance, any consensus method directly or indirectly analyzes vectors of concatenated labels (see Definition 3 in Section 2.1). Therefore, whenever a consensus method discards one specific concatenation of labels because it rarely appears during analysis, this action is analogous to a cell pruning, in the sense of BMCA, as far as each cell corresponds to a specific concatenation of labels.

One should agree that discarding rare concatenations of labels is a rather straightforward idea, by itself. Although it may be difficult to identify this step inside each specific consensus method, we may infer that most of them includes it, even in some indirect manner, and we claim that it may partially explain the superiority of clustering consensus algorithms, as compared to conventional clustering approaches.

But there is an even more important aspect of space quantization with clustering ensembles that must be taken into account as well: the quantization proposed in (Postaire et al., 1993) is made trough a regular grid, whereas the "grid" obtained from the merging of individual clustering partitions tends to be irregular. Indeed, cells

Table 1
Comparison pattern misclassifications with and without cell pruning.

Method	No pruning (%)	Pruning (%)
IVC	11.7	6.4
CSPA	8.3	3.4
MPLA	10.0	6.8

area/volume/hypervolume strongly depends on how the clustering committee is formed. For instance, we expect a committee of random lines/plans/hyperplans to provide cells with balanced area/volume/hypervolume through the whole pattern space. By contrast, committees of conventional clustering algorithms are expected to form a grid of unbalanced cells, for they are expected to put boundaries more likely around/between-clusters then across clusters.

In order to provide an easily understandable illustration of this idea, we adapted the MLP-CLUST algorithm (Montalvão et al., 2007) to move the random lines shown in Fig. 6, gradually pushing them toward regions of the space less dense in terms of patterns. This adaptation of the MLP-CLUST is a very simple kind of valley-seeking algorithm (Theodoridis and Koutroumbas, 2003), in which each line is associated to an artificial neuron, which is independently adapted through a Hebbian-like (self-organized) learning rule. Fig. 7 illustrates how lines tend toward the S-shaped valley between the two clusters. Consequently, cells inside the valley, after adaptation, tend to be smaller than outside it.

It is worth remembering that, in label space, the Hamming distance between the projection of patterns from two different cells is given by the minimum number of cell boundaries one must cross to go from one cell to the other. In other words, the averaged separation between-clusters, in Hamming distance, is clearly increased after adaptation (Fig. 7, right), since more lines are found inside the S-shaped valley between the two clusters. Through an analogous reasoning, we also realize that the averaged Hamming distance between pattern inside each cluster is reduced.

Increasing between-cluster separation and decreasing within-cluster dispersion is easily recognizable as a way to facilitate the clustering task, regardless the clustering algorithm we use. Therefore, an interesting question to be studied is whether any clustering ensemble has this suitable effect on the label space.

In the next Section, we address this question through experiments.

5. On the effect of irregular space quantization

In this Section, we experimentally study how clustering ensembles take advantage of irregular data space quantization. Five data sets were used in our experiments, two of them are synthetic and three are publicly available at the UCI repository of machine learning databases (Blake, 1998), namely:

- 4-Gaussians: A data set corresponding to toy clustering problem, with 4 well separated radial (standard variation = 0.1) clusters in 2-dimensional space, centered at $(-1,1)$, $(-1,-1)$, $(1,-1)$ and $(1,1)$, respectively. This set contains 200 real-valued vectors (50 per cluster).
- Rings: Half-rings data set used in (Postaire et al., 1993), with 1000 real-valued vectors corresponding to 2-dimensional patterns, from 2 classes, as illustrated in Fig. 4.
- Wines: Italian Wines data set, with 178 real-valued vectors corresponding to 13-dimensional patterns, from 3 classes.
- Iris: Iris data set, with 150 real-valued vectors corresponding to 4-dimensional patterns, also from 3 classes.
- WDBC: Wisconsin Diagnostic Breast Cancer data set, with 569 real-valued vectors corresponding to 30-dimensional patterns, from 2 classes.

To measure how pattern space quantization affects clusters dispersion in label space, we further define some criterion functions for within-class and between-class dispersions analysis, namely:

- J_w^E: Averaged Euclidean distance between patterns from the same classes.
- J_b^E: Averaged Euclidean distance between patterns from different classes.
- J_w^H: Averaged Hamming distance between label vectors from the same classes.
- J_b^H: Averaged Hamming distance between label vectors from different classes.

We highlight that these criterion functions use the *a priori* knowledge of the classes from which patterns come from. Please note that, for real-world clustering problems, we usually do not know beforehand whether classes of patterns do correspond to well separated clusters.

As stated before, in label space, the distance between two partition cells corresponds to the minimum number of cell boundaries one should cross to go from one to another cell. Therefore, the volume of each cell is not explicitly taken into account in distance measures. In other words, it does not matter whether two cells are separated by a big cell or by a very thin one, they are the same Hamming distance apart.

If a conventional clustering algorithm, such as the *K*-Means, is used with data sets featuring N_C well separated clusters, and if we further know beforehand the actual number of clusters to be found, then we naturally expect this algorithm to more likely place boundaries between-clusters than across them. If we gather an ensemble of C clusterings under such a favorable setup, then the averaged between-cluster Hamming distance must be close to C. Ideally, we must find C boundaries between each pair of clusters, one boundary from each clustering algorithm, in case they all correctly find all clusters. Although this idealized case makes the clustering ensemble obsolete, and it almost corresponds to what we obtain through the 4-Gaussians data set, we hope that this very simple set will help us to better explain why the J_b^H/J_w^H ratio increases in label spaces.

Thus, if we use an ensemble of 100 *K*-means with the 4-Gaussians data set, a ratio of $J_b^H = 94$ is obtained (close to $C = 100$, as expected), whereas $J_w^H = 5$. Note that if all 100 committee members had correctly found four clusters, these values would be 100 and

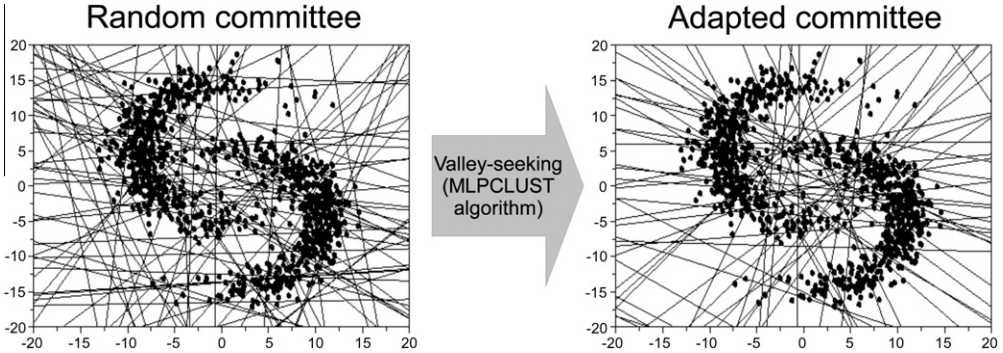

Fig. 7. Adaptation of 80 random partitions with a simple valley-seeking algorithm – The MLP-CLUST algorithm (Montalvão et al., 2007).

0, respectively. In other words, though most boundaries are lying between clusters, some few boundaries are lying across clusters.

5.1. Increasing J_b^H/J_w^H – Synthetic data

If we compare the original 2-D pattern space to the 100-D label space, through their respective between-cluster separation and within-cluster dispersion, we obtain:

- In original pattern space: $J_b^E/J_w^E = 6.7$ (Euclidean distances).
- In label space: $J_b^H/J_w^H = 18.8$ (Hamming distance).

Though distances are differently calculated in pattern and label spaces, this sounding difference between ratios clearly indicates that, in label space, the 4 clusters are even more compact and well separated then in pattern space.

K-means with Euclidean distance is probably the most popular clustering algorithm based on prototypes, and the 4-Gaussians data set fits pretty well the basic assumptions behind this algorithm: that clusters are well separated and radially dispersed. In order to test if this gain of J_b^H/J_w^H with respect to J_b^E/J_w^E is conditioned to the use of *K*-Means ensembles, we repeated the experiment with a committee of clustering algorithms from a very different paradigm: we replaced *K*-means with the Fukunaga's algorithm (Fukunaga, 1990).

The Fukunaga's algorithm, unlike the *K*-Means, doesn't use prototypes to partition data. Instead, it places cluster boundaries in pattern space regions with low data density – the "valleys" of underlying Probability Density Functions from which patterns are drawn. Moreover, every pattern may directly influence boundary contour, usually providing much more irregular boundaries than *K*-Means. Though the number of clusters, *K*, to be found must be given beforehand, as in *K*-Means, another important particularity of the Fukunaga's algorithm is that it may find a number of clusters smaller than *K*.

From the experiment with an ensemble of 100 Fukunaga's algorithms, we obtained, in label space: $J_b^H = 89$, less than 100, thus indicating that some committee members failed in finding the 4 clusters. And $J_w^H = 0$, indicating that no boundary is lying across cluster. In other words, J_b^H/J_w^H diverges to infinity, because every cluster felt into a specific cell. Since, in the label space, each cluster contracted to a single point, this is the most suitable effect to facilitate cluster analysis. This is even a better result as compared to that obtained with *K*-Means committee.

Another evidence of this helpful effect, with a data set not so easy to be clustered is that one shown in Fig. 7. There, a committee of random partition was adapted through another valley-seeking algorithm, the MLP-CLUST. In this case, with the random committee, we have $J_b^H/J_w^H = 2.4$, whereas this ratio is slightly increased to 2.8, after adaptation.

5.2. Increasing J_b^H/J_w^H – Real-world data

Table 2 shows J_b^E/J_w^E values for real-world data sets, in original pattern space, and the corresponding averaged J_b^H/J_w^H in label spaces, with 100 members per clustering committee.

Table 2
Comparison between J_w/J_b ratios in original and projected spaces.

Data	J_b^E/J_w^E	J_b^H/J_w^H (*K*-means)	J_b^H/J_w^H (Fukunaga's)
Wines	1.78	3.78	2.79
WDBC	1.04	1.78	1.46
Iris	2.37	2.29	2.35

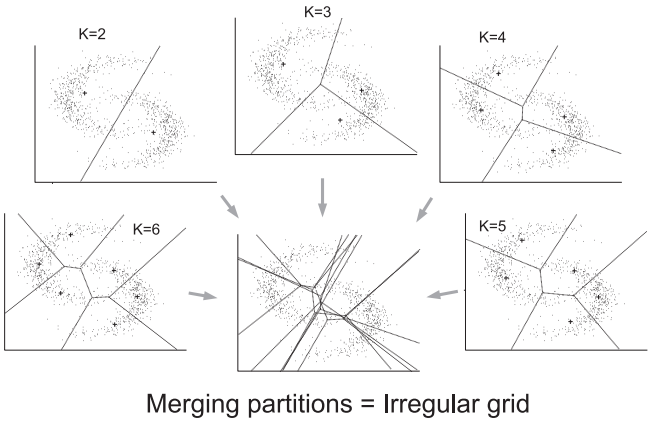

Fig. 8. From 2 to 6 individuals *K*-means partitions and their partition merging.

Except for the Iris data set, J_b^H/J_w^H is greater than J_b^E/J_w^E with both kinds of clustering ensembles. This result indicates that the 3 classes in Iris data set do not form well defined clusters, neither for *K*-Means, nor for Fukunaga's algorithm. In other words, both algorithms were unable to find stable cluster configurations, and we may conclude that the J_b^H/J_w^H gain with respect to J_b^E/J_w^E is data dependent.

5.3. Committees of valley-seeking algorithms

Both MLP-CLUST and Fukunaga's algorithm are examples of valley-seeking algorithms (Fukunaga, 1990; Theodoridis and Koutroumbas, 2003), which, along with boundary-detection algorithms (Atiya, 1990), tend to produce data space partitions with boundaries falling in low-density parts of the space, whereas algorithms such as the *K*-means, which aim at finding prototypes in dense regions, may place cells boundaries between those prototypes, even if it corresponds to high-density parts of the space.

Therefore, consensuses of a *K*-means based committees tend to produce grids which potentially split actual clusters in too many parts. Fig. 8 illustrates this effect by combining 5 partitions provided by 5 *K*-means runs, with 2–6 prototypes, respectively.

By contrast, as illustrated in Fig. 9, the combination of valley-seeking algorithms should, in average, put more boundaries inside the between-clusters regions (valleys), thus increasing the between-clusters distance in the label space, mainly when the imposed *K* is not too high.[3]

This hypothesized advantage of combination of valley-seeking algorithms is also suggested by the point of view from which clustering ensembles non-linearly quantizes pattern space. In order to gather evidences of it, new test results are presented in terms of cluster assignments in disagreement with the known class labels. This measure, frequently used in literature, assumes that classes do form separated clusters in data space. The number of bad cluster assignments is referred to as the *number of misclassifications*.

Each result presented in Tables 3–6 was averaged over 5 independently obtained values, corresponding to 5 independent committees of 50 clustering members each one. As for the IVC consensus algorithm, since it may provide rather unstable results, the number of misclassifications presented in all tables correspond to an averaging of 50 IVC results, for each committee. In all experiments, the number of clusters, *K*, assigned to each committee member is randomly drawn from $\{2,3,\ldots,10\}$.

[3] Please note that we no longer impose the same *K* (number of clusters) to committee members, as in experiments in the former Subsection.

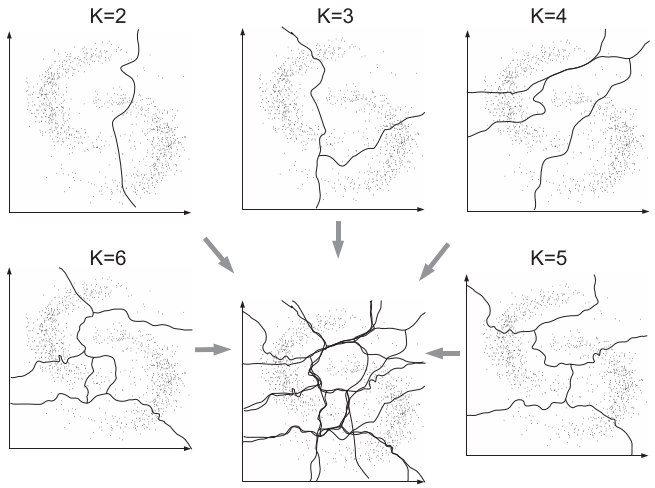

Fig. 9. From 2 to 6 individuals Fukunaga's partitions and their partition merging.

Table 3
Comparison between committee types – Rings database.

Method	*K*-means	Fukunaga
MCLA	38	33
CSPA	76	45
IVC	76	57

Note: With this data set, the HGPA provided almost 50% of misclassifications with both *K*-means and Fukunaga's committees.

Table 4
Comparison between committee types – Iris database.

Method	*K*-means	Fukunaga
MCLA	3.0	4.5
HGPA	7.2	4.0
CSPA	8.2	4.4
IVC	44.2	25.3

Table 5
Comparison between committee types – WDBC database.

Method	*K*-means	Fukunaga
MCLA	45.3	65.0
HGPA	201.5	78.1
CSPA	122.0	100.2
IVC	76.9	52.1

Table 6
Comparison between committee types – Wines database.

Method	*K*-means	Fukunaga
MCLA	10.6	11.6
HGPA	5.6	11.5
CSPA	12.3	13.1
IVC	25.5	22.3

From these tables, we see that, except for the Wines data set, the valley-seeking committee seems to induce improved consensus, except for the MCLA.

Interestingly, it is worth noting that projecting patterns in a label space, through clustering ensembles, in some cases, may even increase the number of misclassifications, in average! For instance, by independently applying the *K*-Means 500 times with the Wines data set, we obtain an averaged number of misclassifications of 9.2. By comparing this result to those displayed in Table 6, we may conclude that, for this data set, except for the HGPA with *K*-Means ensemble, the tested clustering ensemble strategies should be avoided if one's goal is to minimize misclassifications.

On the other hand, as formerly shown in Table 2, pattern space quantization through clustering ensemble do increase the J_b^H/J_w^H ratio, with respect to J_b^E/J_w^E. In fact, to better understand how these apparently contradictory results are related, we believe that a deep study on the relationship between misclassifications and cluster shapes is necessary, but it is beyond the scope of this work.

For a while, we keep claiming that valley-seeking committee may induce improved consensus. Though this claim is rather weakly supported by our experimental results, we highlight the analytical reasoning behind it, based on the nonlinear space quantization interpretation proposed in this paper. It is worth noting that the averaged superiority of valley-seeking committee was hypothesized before any experiment was done.

6. On the diversity of clustering ensembles

The placement of boundaries in pattern space also concerns another important concept in clustering ensemble: diversity. Formal definitions of ensemble diversity is frequently given through entropy related measures (e.g. based on Normalized Mutual Information), but other formal definitions may be given too, such as in (Gullo et al., 2009), where another assessment criterion used in Information Retrieval and Machine Learning, the F-Measure, was used to assign weights to clustering solutions in a given ensemble.

Beyond any formal definition, the concept of diversity of a clustering ensemble may be easily understood as a measure of disagreement amidst proposed partitions, or equivalently, how the various clustering solutions are dissimilar to each other. Therefore, from the space quantization point of view, higher diversity produces finer quantization grids. On the other hand, low diversity means many grid boundaries lying between the same subsets of patterns, forming void grid cells.

In Fig. 7, we can see that random partitions (left) produce high diversity whereas, after adaptation (right), boundaries (straight lines, in this case) converge to between-clusters spaces, thus reducing diversity.

Ensembles must have some level of diversity to properly work, but how much diversity is necessary? It is an important question, and we believe that the point of view proposed in this work may be useful again. For instance, let us consider the very simple ensembles presented in Fig. 7. It is clear that the high diversity random ensemble on the left is less useful, for clustering purposes, than the adapted ensemble on the right. Indeed, as discussed in Section 5, clustering in label space is easier than it is in original space, thanks to its increased J_b^H/J_w^H ratio, with respect to J_b^E/J_w^E.

On the other hand, if we just keep the random ensemble (higher diversity) unchanged, forming a very fine and regular quantization grid, and we measure distances between projected patters in label space, then Hamming distances in label space tend to be just discretized versions of original Euclidean distances, in pattern space, due to the fine and regular discretization grid. In other words, such an ensemble with very high diversity does not facilitate clustering task in label space.

To illustrate this reasoning with an experiment, we first run the *K*-Means algorithm 100 times (independent initializations) and we count the averaged number of misclassifications. Afterwards, we do the same, with the *K*-Means running in label spaces spanned by a ensembles of 80 random partitions (as in Fig. 6). The averaged results, in terms of misclassifications are:

- in pattern space: 127.5 misclassifications,
- in label space: 124.7 misclassifications.

As expected, results are similar. Thus, seeing clustering ensembles as a pattern space quantizer help us to understand that ideal diversity should not be too high, otherwise there is no gain in using clustering ensembles, with respect to conventional clustering approaches.

7. Conclusion

In this paper, a new perspective for clustering ensembles was presented. This new perspective is manyfold, and can be summarized as follows:

- The merging of all partitions from a clustering committee maximizes the (non-normalized) Averaged Mutual Information between a consensus partition and that committee.
- This consensus does not maximize the normalized criterion, the ANMI, which can be reformulated as a Lagrangian function, with a penalization term for consensus partitions with too many cells.
- Unfortunately, the ANMI maximization is not a guaranty of good clusterings. Its apparent success as a blind substitute for the misclassification ratio is probably limited to problems with a few actual number of clusters, since the ANMI penalizes consensus with an elevated number of partitions (estimated clusters).
- The combination of all partitions from a clustering committee seems to be a good starting point for finding consensus partitions.
- From this starting point, which maximizes the AMI, cells can be pruned and fused, thus allowing for a necessary decreasing of the AMI (after its maximization).
- To fuse partition cells, we must consider distances, either in data, or in label space.
- Consequently, Clustering Consensus can thus be regarded as a two-layered clustering task: first, in pattern space, and secondly, in label space (discretized space).
- If we consider distances in label space, Clustering Consensus becomes very close to the Binary Morphology Clustering Algorithm, proposed in 1993.
- From this similarity, we can learn that Clustering Consensus may take advantage of the pruning of sparse partition cells.
- Unlike the Binary Morphology Clustering Algorithm, Clustering Consensus may also take advantage of the nonlinearity of the grid quantization provided by the clustering committee. A well-tuned committee can, for instance, contract clusters and expand between-cluster spaces, through nonlinear mappings from pattern spaces into label spaces.

We hope that this proposed point of view can be used to improve Cluster Consensus design. For instance, we found some evidence that Committees of valley-seeking clustering algorithms may improve consensus performance, which corroborates our belief that nonlinear space distortions which contract clusters are provided by this kind of committee.

Moreover, through this new regard, we can also clearly see how diversity amongst committee members is necessary for boundaries in the space discretization grid not to collapse (thus avoiding partition cells with null area/volume/hypervolume). Similarly, we can also see that finding consensus is a matter of discarding sparse partition cells and fusing those which are closer in label space. It explains why even committees of completely *random* partitions may lead to useful consensus: because they tend to provide a uniform discretization grid. Moreover, we also infer that, in this case, only cell pruning can be used to improve consensus performance, since the uniform grid discards the possibility of taking advantage of a well-tuned nonlinear mapping between pattern-space and label-space.

In the sequel of this work, we are going to study how to design improved pattern spaces mappings into label spaces through the choices of committee members.

Acknowledgment

This work was granted by the *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq).

References

Al-Razgan, M., Domeniconi, C. 2006. Weighted clustering ensembles. In: Proc. SIAM Internat. Conf. Data Mining, pp. 258–269.
Atiya, A.F., 1990. An unsupervised learning technique for artificial neural networks. Neural Networks 3, 707–711.
Blake, Merz, C.J. 1998. UCI repository of machine learning databases.
Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification. John Wiley & Sons, New York.
Feldman, J.A., Ballard, D.H., 1982. Connectionist models and their properties. Cognitive Sci. 6, 205–254.
Fern, X.Z., Brodley, C.E. 2003. Random projection for high dimensional data clustering: A cluster ensemble approach. In: Proc. Twentieth Internat. Conf. Machine Learning, pp. 186–193.
Fred, A., Jain, A.K., 2005. Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Machine Intell. 27, 835–850.
Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition, second ed. Academic Press.
Gullo, F., Tagarelli, A., Greco, S. 2009. Diversity-based weighting schemes for clustering ensembles. In: Proc. SDM2009, pp. 437–448.
Hong, Y., Kwong, S., Wanga, H., Ren, Q., 2009. Resampling-based selective clustering ensembles. Pattern Recognition Lett. 30, 298–305.
MacKay, D.J.C., 2003. Information Theory, Inference, and Learning Algorithms. Cambridge University Press.
Mitra, S., Pal, S.K., Mitra, P., 2002. Data mining in soft computating framework: A survey. IEEE Trans. Neural Networks 13 (1), 3–14.
Montalvão, J.R., Dorizzi, B., Mota, J.C.M., 2002. Channel estimation by symmetrical clustering. IEEE Trans. Signal Process. 50 (6), 1459–1469.
Montalvão, J.R., Freire, E.O., Bezerra Jr., M.A., 2007. Clustering with multilayer perceptrons and self-organized (Hebbian) learning. J. Intell. Fuzzy Systems 18, 501–511.
Nguyen, N., Carauna, R. 2007. Consensus Clusterings. In: Proc. IEEE Internat. Conf. on Data Mining (ICDM'07).
Postaire, J.-G., Zhang, R.D., Lecocq-Botte, C., 1993. Cluster analysis by binary morphology. IEEE Trans. Pattern Anal. Machine Intell. 15 (2), 170–180.
Strehl, A., Ghosh, J., 2002. Cluster ensembles – A knowledge reuse framework for combining multiple partitions. J. Machine Learn. Res. 3, 583–617.
Theodoridis, S., Koutroumbas, K., 2003. Pattern Recognition, second ed. Elsevier Academic Press.
Topchy, A., Jain, A.K., Punch, W., 2005. Clustering ensembles: Models of consensus and weak partitions. IEEE Trans. Pattern Anal. Machine Intell. 27 (12), 1866–1881.
Tumer, K., Agogino, A.K., 2008. Ensemble clustering with voting active clusters. Pattern Recognition Lett. 29 (14).
Yu, Z., Deng, Z., Wong, H.-S., Wang, X. 2008. Knowledge based cluster ensemble. In: IEEE World Congress on Comput. Intell.