Mach Learn (2015) 98:331–357
DOI 10.1007/s10994-013-5339-6

Probabilistic consensus clustering using evidence accumulation

André Lourenço · Samuel Rota Bulò · Nicola Rebagliati · Ana L.N. Fred · Mário A.T. Figueiredo · Marcello Pelillo

Received: 25 May 2012 / Accepted: 2 March 2013 / Published online: 3 April 2013

Abstract Clustering ensemble methods produce a consensus partition of a set of data points by combining the results of a collection of base clustering algorithms. In the *evidence accumulation clustering* (EAC) paradigm, the clustering ensemble is transformed into a pairwise co-association matrix, thus avoiding the label correspondence problem, which is intrinsic to other clustering ensemble schemes. In this paper, we propose a consensus clustering approach based on the EAC paradigm, which is not limited to crisp partitions and fully exploits the nature of the co-association matrix. Our solution determines probabilistic assign-

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

A. Lourenço
Instituto Superior de Engenharia de Lisboa, Lisboa, Portugal

A. Lourenço · A.L.N. Fred · M.A.T. Figueiredo
Instituto de Telecomunicações, Lisboa, Portugal

A. Lourenço
e-mail: arlourenco@lx.it.pt

A.L.N. Fred
e-mail: afred@lx.it.pt

M.A.T. Figueiredo
e-mail: mario.figueiredo@lx.it.pt

S. Rota Bulò (✉) · M. Pelillo
DAIS, via Torino, 155, Mestre, Venezia, Italy
e-mail: srotabul@dais.unive.it

M. Pelillo
e-mail: pelillo@dais.unive.it

N. Rebagliati
VTT Technical Research Center of Finland, P.O. Box 1000, VTT 02044, Finland
e-mail: nicola.rebagliati@gmail.com

A.L.N. Fred · M.A.T. Figueiredo
Instituto Superior Técnico, 1049-001 Lisboa, Portugal

ments of data points to clusters by minimizing a Bregman divergence between the observed co-association frequencies and the corresponding co-occurrence probabilities expressed as functions of the unknown assignments. We additionally propose an optimization algorithm to find a solution under any double-convex Bregman divergence. Experiments on both synthetic and real benchmark data show the effectiveness of the proposed approach.

Keywords Consensus clustering · Evidence Accumulation · Ensemble clustering · Bregman divergence

1 Introduction

Clustering ensemble methods look for consensus solutions from a set of base clustering algorithms, thus trying to combine into a single partition the information present in many different ones. Several authors have shown that these methods tend to reveal more robust and stable cluster structures than the individual clusterings in the ensemble (Fred 2001; Fred and Jain 2002; Strehl and Ghosh 2003). Leveraging an ensemble of clusterings is considerably more difficult than combining an ensemble of classifiers, due to the label correspondence problem: how to put in correspondence the cluster labels produced by different clustering algorithms? This problem is made more serious if clusterings with different numbers of clusters are allowed in the ensemble.

A possible solution to sidestep the cluster label correspondence problem has been proposed in the Evidence Accumulation Clustering (EAC) framework (Fred and Jain 2005). The core idea is based on the assumption that similar data points are very likely grouped together by some clustering algorithm and, conversely, data points that co-occur very often in the same cluster should be regarded as being very similar. Hence, it is reasonable to summarize a clustering ensemble in terms of a pair-wise similarity matrix, called *co-association matrix*, where each entry counts the number of clusterings in the ensemble in which a given pair of data points is placed in the same cluster. This new mapping can then be used as input for any similarity-based clustering algorithm. In Fred and Jain (2005), agglomerative hierarchical algorithms are used to extract the consensus partition (e.g. Single Link, Average Link, or Ward's Link). In Fred and Jain (2006), an extension is proposed, entitled Multi-Criteria Evidence Accumulation Clustering (Multi-EAC), filtering the cluster combination process using a cluster stability criterion. Instead of using the information of the different partitions, it is assumed that, since algorithms can have different levels of performance in different regions of the space, only certain clusters should be considered.

The way the co-association matrix is exploited in the literature is very naïve. Indeed, standard approaches based on EAC simply run a generic pairwise clustering algorithm with the co-association matrix as input. The underlying clustering criteria of ad hoc algorithms, however, do not take advantage of the statistical interpretation of the computed similarities, which is an intrinsic part of the EAC framework. Also, the direct application of a clustering algorithm to the co-association matrix typically induces a hard partition of the data. Although having crisp partitions as baseline for the accumulation of evidence of data organization is reasonable, this assumption is too restrictive in the phase of producing a consensus clustering. Indeed, the consensus partition is a solution that tries to accommodate the different clusterings in the ensemble and by allowing soft assignments of data points to clusters we can preserve some information about their intrinsic variability and capture the level of uncertainty of the overall label assignments, which would not be detected in the case of hard consensus partitions. The variability in the clustering solution of the ensemble might depend not only on the different algorithms and parametrizations adopted to build the ensemble, but also on the presence of clusters that naturally overlap in the data. This is the case for many important applications such as clustering micro-array gene expression data, text categorization, perceptual grouping, labelling of visual scenes and medical diagnosis. In these cases, having a consensus solution in terms of a soft partition allows to detect also overlapping clusters charactering the data. It is worth mentioning that the importance of dealing with overlapping clusters has been recognized long ago (Jardine and Sibson 1968) and, in the machine learning community, there has been a renewed interest around this problem (Banerjee et al. 2005a; Heller and Ghahramani 2007). As an alternative, the consensus extraction could be obtained by running fuzzy k-medoids (Mei and Chen 2010) on the co-association matrix as it were a standard similarity matrix, or fuzzy k-means (Bezdek 1981) by interpreting each row of the co-association matrix as a feature vector. However, such solutions would not take into account the underlying probabilistic meaning of the co-association matrix and lack any formal statistical support.

In this paper, we propose a consensus clustering approach which is based on the EAC paradigm. Our solution fully exploits the nature of the co-association matrix and does not lead to crisp partitions, as opposed to the standard approaches in the literature. Indeed, it consists of a model in which data points are probabilistically assigned a cluster. Moreover, each entry of the co-association matrix, which is derived from the ensemble, is regarded as a realization of a Binomial random variable, parametrized by the unknown cluster assignments, that counts the number of times two specific data points are expected to be clustered together. A consensus clustering is then obtained by means of a maximum likelihood estimation of the unknown probabilistic cluster assignments. We further show that this approach is equivalent to minimizing the Kullback-Leibler (KL) divergence between the observed co-occurrence frequencies derived from the co-association matrix and the co-occurrence probabilities parametrizing the Binomial random variables. By replacing the KL-divergence with any Bregman divergence, we come up with a more general formulation for consensus clustering. In particular we consider, as an additional example, the case where the squared Euclidean distance is used as divergence. We also propose an optimization algorithm to solve the minimization problem derived from our formulation, which works for any double-convex Bregman divergence, and a comprehensive set of experiments shows the effectiveness of our new consensus clustering approach.

The remainder of the paper is organized as follows. In Sect. 2 we provide definitions and notations that will be used across the manuscript. In Sects. 3 and 4, we describe the proposed formulation for consensus clustering and the corresponding optimization problem. In Sect. 5, we present an optimization algorithm the can be used to find a consensus solution. Section 6 briefly reviews related work, and Sect. 7 reports experimental results. Finally, Sect. 8 presents some concluding remarks. A preliminary version of this paper appeared in Rota Bulò et al. (2010).

2 Notation and definitions

Sets are denoted by upper-case calligraphic letters (*e.g.*, $\mathcal{O}$, $\mathcal{E}$, ...) except for $\mathbb{R}$ and $\mathbb{R}_+$ which represent as usual the sets of real numbers and non-negative real numbers, respectively. The *cardinality* of a finite set is written as $|\cdot|$. We denote *vectors* with lower-case boldface letters (*e.g.*, $\mathbf{x}$, $\mathbf{y}$, ...) and *matrices* with upper-case boldface letters (*e.g.*, $\mathbf{X}$, $\mathbf{Y}$, ...). The ith component of a vector $\mathbf{x}$ is denoted as x_i and the (i, j)th component of a matrix $\mathbf{Y}$ is written as y_{ij}. The *transposition* operator is given by the symbol $^\top$. The ℓ_p-norm

Table 1 Examples of double-convex Bregman divergences

Divergence	$\phi(\mathbf{x})$	$d_\phi(\mathbf{x},\mathbf{y})$	Domain
Squared ℓ_2	$\|\mathbf{x}\|^2$	$\|\mathbf{x}-\mathbf{y}\|^2$	$\mathbf{x},\mathbf{y}\in\mathbb{R}^K$
Mahalanobis	$\mathbf{x}^\top A\mathbf{x}$	$(\mathbf{x}-\mathbf{y})^\top A(\mathbf{x}-\mathbf{y})$	$\mathbf{x},\mathbf{y}\in\mathbb{R}^K$, $A\succcurlyeq 0$
Kullback-Leibler	$-H(\mathbf{x})$	$\sum_{j=1}^K x_j\log(\frac{x_j}{y_j})$	$\mathbf{x},\mathbf{y}\in\Delta_K$
Generalized I-div.	$-H(\mathbf{x})-\mathbf{e}^\top\mathbf{x}$	$\sum_{j=1}^K x_j\log(\frac{x_j}{y_j})-x_j+y_j$	$\mathbf{x},\mathbf{y}\in\mathbb{R}^K_+$

of a vector $\mathbf{x}$ is written as $\|\mathbf{x}\|_p$ and we implicitly assume a ℓ_2 (or Euclidean) norm, where p is omitted. We denote by $\mathbf{e}_n$ a n-dimensional column vector of all 1's and by $\mathbf{e}_n^{(j)}$ the jth column of the n-dimensional identity matrix. The *trace* of matrix $\mathbf{M}\in\mathbb{R}^{n\times n}$ is given by $\mathrm{Tr}(\mathbf{M})=\sum_{i=1}^n m_{ii}$. The domain of a function f is denoted by $\mathrm{dom}(f)$ and $\mathbb{1}_P$ is the indicator function giving 1 if P is true, 0 otherwise.

A *probability distribution* over a finite set $\{1,\dots,K\}$ is an element of the *standard simplex* Δ_K, which is defined as

$$\Delta_K=\big\{\boldsymbol{x}\in\mathbb{R}^K_+:\|\mathbf{x}\|_1=1\big\}.$$

The *support* $\sigma(\mathbf{x})$ of a probability distribution $\mathbf{x}\in\Delta_K$ is the set of indices corresponding to positive components of $\mathbf{x}$, *i.e.*,

$$\sigma(\mathbf{x})=\big\{i\in\{1,\dots,K\}:x_i>0\big\}.$$

The *entropy* of a probability distribution $\mathbf{x}\in\Delta_K$ is given by

$$H(\mathbf{x})=-\sum_{j=1}^K x_j\log(x_j)$$

and the *Kullback-Leibler divergence* between two distributions $\mathbf{x},\mathbf{y}\in\Delta_K$ is given by

$$D_{KL}(\mathbf{x}\|\mathbf{y})=\sum_{j=1}^K x_j\log\left(\frac{x_j}{y_j}\right),$$

where we assume $\log 0\equiv-\infty$ and $0\log 0\equiv 0$.

Given a continuously-differentiable, real-valued and strictly convex function $\phi:\Delta_K\to\mathbb{R}$, we denote by

$$B_\phi(\mathbf{x}\|\mathbf{y})=\phi(\mathbf{x})-\phi(\mathbf{y})-(\mathbf{x}-\mathbf{y})^\top\nabla\phi(\mathbf{y})$$

the *Bregman divergence* associated with ϕ for points $\mathbf{x},\mathbf{y}\in\Delta_K$, where ∇ is the *gradient operator*. By construction, the Bregman divergence is convex in its first argument. If convexity holds also for the second one, then we say that the Bregman divergence is *double-convex*.

The Kullback-Leibler divergence is a special case of double-convex Bregman divergence, which is obtained by considering $\phi(\mathbf{x})=-H(\mathbf{x})$. In Table 1 we report other examples of double-convex Bregman divergences.

3 A probabilistic model for consensus clustering

Consensus clustering is an unsupervised learning approach that summarizes an ensemble of partitions obtained from a set of base clustering algorithms into a single consensus partition. In this section, we introduce a novel model for consensus clustering, which collapses the information gathered from the clustering ensemble into a single partition, in which data points are assigned to clusters in a probabilistic sense.

Let $\mathcal{O} = \{1, \ldots, n\}$ be the indices of a set of data points to be clustered and let $\mathcal{E} = \{p_u\}_{u=1}^N$ be a clustering ensemble, *i.e.*, a set of N clusterings obtained by different algorithms with possibly different parametrizations and/or initializations and/or sub-sampled versions of the data set. Each clustering $p_u \in \mathcal{E}$ is a function $p_u : \mathcal{O}_u \to \{1, \ldots, K_u\}$ assigning a cluster out of K_u available ones to data points in $\mathcal{O}_u \subseteq \mathcal{O}$, where $\mathcal{O}_u$ and K_u can be different across the clusterings indexed by u. We put forward data sub-sampling as a most general framework for the following reasons: it favours the diversity of the clustering ensemble and it models situations of distributed clustering where local clusters have only partial access to the data.

Since each clustering in the ensemble may stem from a sub-sampled version of the original dataset, some pairs of data points may not appear in all clusterings. Let $\Omega_{ij} \subseteq \{1, \ldots, N\}$ denote the set of indices of clusterings in the ensemble where both data points i and j appear, *i.e.*, $(u \in \Omega_{ij}) \Leftrightarrow (\{i, j\} \subseteq \mathcal{O}_u)$, and let $N_{ij} = |\Omega_{ij}|$ denote its cardinality. Clearly, $\Omega_{ij} = \Omega_{ji}$ and consequently $N_{ij} = N_{ji}$ for all pairs (i, j) of data points. The ensemble of clusterings is summarized in the *co-association matrix* $\mathbf{C} = [c_{ij}] \in \{0, \ldots, N\}^{n \times n}$, where

$$c_{ij} = \sum_{u \in \Omega_{ij}} \mathbb{1}_{p_u(i)=p_u(j)}$$

is the number of times i and j are co-clustered in the ensemble $\mathcal{E}$; of course, $0 \leq c_{ij} \leq N_{ij} \leq N$ and $c_{ij} = c_{ji}$.

In standard EAC literature the co-association matrix holds the *fraction* of times two data points are co-clustered, while in our definition it holds the *number* of times this event occurs. The reason of this choice stems from the fact that we allow subsampling in the ensemble construction. Consequently, the number of times two data points appear in a clustering of the ensemble is not constant over all possible pairs. This renders the observation of the fraction of times co-clustering occurs statistically more significant for some pairs of data points and less for other ones. By considering $\mathbf{C}$ in absolute terms and by keeping track of the quantities N_{ij}'s we can capture this information. As an example, consider that the ensemble consists on 100 partitions, and due to subsampling, let a pair of samples (i, j), co-appear in partitions $N_{ij} = 80$, and be co-clustered 70 times. Then, $N = 100$, $N_{ij} = 80$ and $c_{ij} = 70$.

Our model assumes that each data point has an unknown probability of being assigned to each cluster. We denote by $\mathbf{y}_i = (y_{1i}, \ldots, y_{Ki})^\top \in \Delta_K$ the probability distribution over the set of K clusters $\{1, \ldots, K\}$ which characterizes data point $i \in \mathcal{O}$, *i.e.*, $y_{ki} = \mathbb{P}[i \in \mathcal{C}_k]$, where $\mathcal{C}_k$ denotes the subset of $\mathcal{O}$ that constitutes the k-th cluster. The model parameter K should not be understood as the desired number of clusters but rather as a maximum number of clusters. Without prior knowledge, K might coincide with the number of data points, *i.e.*, $K = n$. Finally, we store all the $\mathbf{y}_i$'s in a $K \times n$ matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \Delta_K^n$. The probability that data points i and j are co-clustered is thus

$$\sum_{k=1}^K \mathbb{P}[i \in \mathcal{C}_k, j \in \mathcal{C}_k] = \sum_{k=1}^K \mathbb{P}[i \in \mathcal{C}_k]\mathbb{P}[j \in \mathcal{C}_k] = \sum_{k=1}^K y_{ki} y_{kj} = \mathbf{y}_i^\top \mathbf{y}_j.$$

Table 2 Summary of notation

Symbol	Description
N_{ij}	Number of times data points i and j are on the same partition
c_{ij}	Number of times data points i and j are co-clustered
$\mathbf{C}$	$\mathbf{C} = [c_{ij}]$, co-association matrix
$\mathbf{y}_i$	Probability distribution of data point i over the set of K clusters
$\mathbf{Y}$	$\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \Delta_K^n$
C_{ij}	$C_{ij} \sim \text{Binomial}(N_{ij}, \mathbf{y}_i^\top \mathbf{y}_j)$

Let C_{ij}, $i < j$, be a binomial random variable representing the number of times that data points i and j are co-clustered; from the assumptions above we have that $C_{ij} \sim \text{Binomial}(N_{ij}, \mathbf{y}_i^\top \mathbf{y}_j)$, that is

$$\mathbb{P}[C_{ij} = c|\mathbf{y}_i, \mathbf{y}_j] = \binom{N_{ij}}{c} \left(\mathbf{y}_i^\top \mathbf{y}_j\right)^c \left(1 - \mathbf{y}_i^\top \mathbf{y}_j\right)^{N_{ij}-c}.$$

Each element c_{ij}, $i < j$, of the *co-associaton matrix* $\mathbf{C}$, is interpreted as a sample of the random variable C_{ij} and due to the symmetry of C, entries c_{ij} and c_{ji} are considered as the same sample. The model considers the different C_{ij}'s independent. This simplification is essential, in practice, because by decoupling the pairwise, or higher order, correlations present in the consensus the likelihood becomes more tractable. Consequently, the probability of observing $\mathbf{C}$, given the cluster probabilities $\mathbf{Y}$, is given by

$$\mathbb{P}[\mathbf{C} \mid \mathbf{Y}] = \prod_{\{i,j\} \in \mathcal{P}} \binom{N_{ij}}{c_{ij}} \left(\mathbf{y}_i^\top \mathbf{y}_j\right)^{c_{ij}} \left(1 - \mathbf{y}_i^\top \mathbf{y}_j\right)^{N_{ij}-c_{ij}}$$

where $\mathcal{P} = \{\{i, j\} \subseteq \mathcal{O} : i \neq j\}$ is the set of all distinct pairs of data points. Since we consider the observations c_{ij} and c_{ji} as being identical due to the symmetry of C, the product is taken over the set of distinct *unordered* pairs of data points.

We can now estimate the unknown cluster assignments by maximizing the log-likelihood $\log \mathbb{P}[\mathbf{C}|\mathbf{Y}]$ with respect to $\mathbf{Y}$, which is given by

$$\log \mathbb{P}[\mathbf{C}|\mathbf{Y}] = \sum_{\{i,j\} \in \mathcal{P}} \log \binom{N_{ij}}{c_{ij}} + c_{ij} \log\left(\mathbf{y}_i^\top \mathbf{y}_j\right) + (N_{ij} - c_{ij}) \log\left(1 - \mathbf{y}_i^\top \mathbf{y}_j\right).$$

This yields the following maximization problem, where terms not depending on $\mathbf{Y}$ have been dropped:

$$\mathbf{Y}^* \in \operatorname*{arg\,max}_{\mathbf{Y} \in \Delta_K^n} \left\{ \sum_{\{i,j\} \in \mathcal{P}} c_{ij} \log\left(\mathbf{y}_i^\top \mathbf{y}_j\right) + (N_{ij} - c_{ij}) \log\left(1 - \mathbf{y}_i^\top \mathbf{y}_j\right) \right\}. \tag{1}$$

Matrix $\mathbf{Y}^*$, the solution of problem (1), provides probabilistic cluster assignments for the data points, which constitute the solution to the consensus clustering problem according to our model.

In Table 2 we summarize the notation introduced in this section.

4 A class of alternative formulations

The formulation introduced in the previous section for consensus clustering can be seen as a special instance of a more general setting, which will be described in this section.

Let $\psi : \mathbb{R} \to \mathbb{R}^2$ be a function mapping a scalar to a 2-dimensional vector defined as $\psi(x) = (x, 1-x)^\top$ and let $d_\phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be given as follows:

$$d_\phi(x_1, x_2) = B_\phi\big(\psi(x_1) \| \psi(x_2)\big).$$

Consider now the following class of formulations for consensus clustering, which is parametrized by a continuously-differentiable and strictly convex function $\phi : \Delta_2 \to \mathbb{R}$:

$$\mathbf{Y}^* \in \arg\min_{\mathbf{Y} \in \Delta_K^n} f(\mathbf{Y}), \tag{2}$$

where

$$f(\mathbf{Y}) = \sum_{\{i,j\} \in \mathcal{P}} N_{ij}\, d_\phi\left(\frac{c_{ij}}{N_{ij}}, \mathbf{y}_i^\top \mathbf{y}_j\right). \tag{3}$$

Intuitively, the solution $\mathbf{Y}^*$ to (2) is a probabilistic cluster assignment yielding a minimum Bregman divergence between the observed co-occurrence statistics of each pair of data points and the estimated ones. Moreover, each term of $f(\mathbf{Y})$ is weighted by N_{ij} in order to account of the statistical significance of the observations.

The formulation in (2) encompasses the one introduced in the previous section as a special case. Indeed, by considering the parametrization $\phi(\mathbf{x}) = -H(\mathbf{x})$, we have that $B_\phi \equiv D_{KL}$, *i.e.*, the Bregman divergence coincides with the KL-divergence, and by simple algebra the equivalence between (2) and (1) can be derived. For a formal proof we refer to Proposition 1 in Appendix.

Different algorithms for consensus clustering can be derived by adopting different Bregman divergences in (2), *i.e.*, by changing the way errors between observed frequencies and estimated probabilities of co-occurrence are penalized. This is close in spirit to the work (Banerjee et al. 2005b), where a similar approach has been adopted in the context of partitional data clustering. In addition to the formulation corresponding to the KL-divergence, in this paper, we study also the case where a squared ℓ_2 penalization is considered in (3), *i.e.*, when $\phi(\mathbf{x}) = \|\mathbf{x}\|^2$ and d_ϕ becomes the squared Euclidean distance. This yields the following optimization problem:

$$\mathbf{Y}^* \in \arg\min_{\mathbf{Y} \in \Delta_K^n} \left\{ \sum_{\{i,j\} \in \mathcal{P}} N_{ij} \left(\frac{c_{ij}}{N_{ij}} - \mathbf{y}_i^\top \mathbf{y}_j\right)^2 \right\}. \tag{4}$$

In the next section we will cover the algorithmic aspects of the computation of probabilistic assignments, which represent our solution to the consensus clustering problem.

5 Optimization algorithm

In this section, we describe an efficient optimization procedure which allows to find a local solution to (2), which works for any double-convex Bregman divergence. This procedure falls in the class of primal line-search methods because it iteratively finds a feasible descent direction, *i.e.*, satisfying the constraints and guaranteeing a local decrease of the objective.

This section is organized into four parts. The first part is devoted to the problem of finding a feasible, descent direction, while the second part addresses the problem of searching a better solution along that direction. In the third part, we summarize the optimization algorithm and provide some additional techniques to reduce its computational complexity. Finally, in the last part we show how our algorithm can be adapted to efficiently cluster large-scale datasets.

5.1 Computation of a search direction

Given a non-optimal feasible solution $\mathbf{Y} \in \Delta_K^n$ of (2), we can look for a better solution along a direction $\mathbf{D} \in \mathbb{R}^{K \times n}$ by finding a value of ϵ such that $f(\mathbf{Z}_\epsilon) < f(\mathbf{Y})$, where $\mathbf{Z}_\epsilon = \mathbf{Y} + \epsilon\mathbf{D}$. The search direction $\mathbf{D}$ is said to be *feasible* and *descending* at $\mathbf{Y}$ if the two following conditions hold for all sufficiently small positive values of ϵ: $\mathbf{Z}_\epsilon \in \Delta_K^n$ and $f(\mathbf{Z}_\epsilon) < f(\mathbf{Y})$.

Our algorithm considers search directions at $\mathbf{Y}$ that are everywhere zero except for two entries lying on the same column. Specifically, it selects directions belonging to the following set:

$$\mathcal{D}(\mathbf{Y}) = \left\{ \left(\mathbf{e}_K^u - \mathbf{e}_K^v\right)\left(\mathbf{e}_n^j\right)^\top : j \in \mathcal{O}, u \in \{1, \dots, K\}, v \in \sigma(\mathbf{y}_j), u \neq v \right\}.$$

Here, the condition imposing $v \in \sigma(\mathbf{y}_j)$ guarantees that every direction in $\mathcal{D}(\mathbf{Y})$ is feasible at $\mathbf{Y}$ (see Proposition 2 in Appendix). Among this set, by taking a greedy decision, we select the direction leading to the steepest descent, *i.e.*, we look for a solution to the following optimization problem:

$$\mathbf{D}^* \in \operatorname*{arg\,min}_{\mathbf{D} \in \mathcal{D}(\mathbf{Y})} \left\{ \lim_{\epsilon \to 0} \frac{d}{d\epsilon} f(\mathbf{Y} + \epsilon\mathbf{D}) \right\}. \tag{5}$$

By exploiting the definition of $\mathcal{D}(\mathbf{Y})$ the solution to (5) can be written as $\mathbf{D}^* = (\mathbf{e}_K^U - \mathbf{e}_K^V)(\mathbf{e}_n^J)^\top$, where the indices U, V and J are determined as follows. Let U_j, V_j be given by

$$U_j \in \operatorname*{arg\,min}_{k \in \{1 \dots K\}} \left[g_j(\mathbf{Y})\right]_k \quad \text{and} \quad V_j \in \operatorname*{arg\,max}_{k \in \sigma(\mathbf{y}_j)} \left[g_j(\mathbf{Y})\right]_k, \tag{6}$$

for all $j \in \mathcal{O}$, where $g_j(\mathbf{Y})$ is the partial derivative of f with respect to $\mathbf{y}_j$, which is given by

$$g_j(\mathbf{Y}) = \frac{\partial}{\partial \mathbf{y}_j} f(\mathbf{Y}) = \sum_{i \in \mathcal{P}_j} N_{ji} \mathbf{y}_i \frac{\partial d_\phi}{\partial x_2}\left(\frac{c_{ji}}{N_{ji}}, \mathbf{y}_j^\top \mathbf{y}_i\right). \tag{7}$$

Here $\mathcal{P}_j = \{i \in \mathcal{O} : \{i, j\} \in \mathcal{P}\}$. Then, by Proposition 3 in Appendix, J can be computed as

$$J \in \operatorname*{arg\,min}_{j \in \mathcal{O}} \left\{ \left[g_j(\mathbf{Y})\right]_{U_j} - \left[g_j(\mathbf{Y})\right]_{V_j} \right\}, \tag{8}$$

while $U = U_J$ and $V = V_J$.

The search direction $\mathbf{D}^*$ at $\mathbf{Y}$ obtained from (5) is clearly feasible since it belongs to $\mathcal{D}(\mathbf{Y})$ but it is also always descending, unless $\mathbf{Y}$ satisfies the Karush-Kuhn-Tucker (KKT) conditions, *i.e.*, the first-order necessary conditions for local optimality, for the minimization problem in (2). This result is formally proven in Proposition 4 in Appendix.

5.2 Computation of an optimal step size

Once a feasible descending direction $\mathbf{D}^* = (\mathbf{e}_K^U - \mathbf{e}_K^V)(\mathbf{e}_n^J)^\top$ is computed from (5), we have to find an optimal step size ϵ^* that allows us to achieve a decrease in the objective value. The optimal step is given as a solution to the following one dimensional optimization problem,

$$\epsilon^* \in \operatorname*{arg\,min}_{0 \le \epsilon \le y_{VJ}} f(\mathbf{Z}_\epsilon), \tag{9}$$

where $\mathbf{Z}_\epsilon = \mathbf{Y} + \epsilon \mathbf{D}^*$ and the feasible interval for ϵ follows from the constraint that $\mathbf{Z}_\epsilon \in \Delta_K^n$. This problem is convex thanks to the assumption of double-convexity imposed on the Bregman divergence (see Proposition 5 in Appendix).

Let $\rho(\epsilon')$ denote the first order derivative of f with respect to ϵ evaluated at ϵ', *i.e.*,

$$\rho(\epsilon') = \lim_{\epsilon \to \epsilon'} \frac{d}{d\epsilon} f(\mathbf{Z}_\epsilon) = \left[g_J(\mathbf{Z}_{\epsilon'})\right]_U - \left[g_J(\mathbf{Z}_{\epsilon'})\right]_V.$$

By the convexity of (9) and Kachurovskii's theorem (Kachurovskii 1960) we have that ρ is non-decreasing in the interval $0 \le \epsilon \le y_{VJ}$. Moreover, $\rho(0) < 0$ since $\mathbf{D}^*$ is a descending direction as stated by Proposition 4. Otherwise, we would have that $\mathbf{Y}$ satisfies the KKT conditions for local optimality.

In order to compute the optimal step size ϵ^* in (9) we distinguish two cases:

- if $\rho(y_{VJ}) \le 0$ then $\epsilon^* = y_{VJ}$ for $f(\mathbf{Z}_\epsilon)$ would be non-increasing in the feasible set of (9);
- if $\rho(y_{VJ}) > 0$ then ϵ^* is a zero of ρ that can be found in general using a dichotomic search which preserves the discording signs of ρ at the endpoints of the search interval.

In the specific, if the second case holds the optimal step size ϵ^* can be found by iteratively updating the search interval as follows:

$$\begin{aligned} \left(\ell^{(0)}, r^{(0)}\right) &= (0, y_{VJ}) \\ \left(\ell^{(t+1)}, r^{(t+1)}\right) &= \begin{cases} (\ell^{(t)}, m^{(t)}) & \text{if } \rho(m^{(t)}) > 0, \\ (m^{(t)}, r^{(t)}) & \text{if } \rho(m^{(t)}) < 0 \\ (m^{(t)}, m^{(t)}) & \text{if } \rho(m^{(t)}) = 0, \end{cases} \end{aligned} \tag{10}$$

for all $t > 0$, where $m^{(t)}$ denotes the center of segment $[\ell^{(t)}, r^{(t)}]$, *i.e.*, $m^{(t)} = (\ell^{(t)} + r^{(t)})/2$. Since an approximation of ϵ^* is sufficient, the dichotomic search is carried out until the interval size is below a given threshold. If δ is this threshold, the number of iterations required is $\log_2(y_{VJ}/\delta)$ at worst.

In some cases (9) has a closed form solution. This of course depends on the nature of the Bregman divergence adopted. For instance, if we consider the squared ℓ_2 distance as a divergence (*i.e.*, $\phi(\mathbf{x}) = \|\mathbf{x}\|^2$), then $f(\mathbf{Z}_\epsilon)$ becomes a quadratic polynomial in the variable ϵ which can be trivially minimized in closed-form.

5.3 Algorithm

The proposed consensus clustering method is summarized in Algorithm 1. The input arguments consist of the ensemble of clusterings $\mathcal{E}$, the parameter ϕ for the Bregman divergence, and an initial guess $\mathbf{Y}^{(0)}$ for the cluster assignments (cluster assignments are uniformly distributed in the absence of prior knowledge).

At an abstract level, the algorithm iteratively finds a feasible, descending direction $\mathbf{D}^*$ at the current solution $\mathbf{Y}^{(t)}$, computes the optimal step ϵ^* and performs an update of the solution as $\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \epsilon^* \mathbf{D}^*$. This procedure is iterated until a stopping criterion is met.

In order to obtain a time complexity per-iteration that is linear in the number of variables, we exploit the extreme sparseness of the search direction $\mathbf{D}^*$ for the update of matrix $\mathbf{Y}^{(t)\top}\mathbf{Y}^{(t)}$ (denoted by $\mathbf{A}^{(t)}$ in the pseudocode) and for the update of the gradient vectors $g_i^{(t)}$. Each iteration, indeed, depends on these two fundamental quantities. In the specific, the computation of $\mathbf{A}^{(t+1)}$ can be obtained in $O(n)$ by simply changing the Jth row and the Jth column of $\mathbf{A}^{(t)}$ (it follows from the update formula at line 10). By exploiting $\mathbf{A}^{(t+1)}$, the gradient vectors can be computed in $O(Kn)$. In fact, we obtain $g_i^{(t+1)}$ for all $i \in \mathcal{O} \setminus \{J\}$ by performing a constant time operation on each entry of $g_i^{(t)}$ (lines 12–14) and we compute $g_J^{(t+1)}$ (line 15) in $O(Kn)$ as well. Having $\mathbf{A}^{(t)}$ and the gradient vectors computed allows us to find the search direction $\mathbf{D}^*$ at line 8 in $O(nK)$, since it suffices to access each element of the gradient vectors only once to determine J, U and V. Moreover, the computation of the optimal step size at line 9 can be carried out in $O(n \log_2(1/\delta))$, if a dichotomic search is employed, and in constant time in cases where a closed-form solution exists (*e.g.*, if $\phi(\mathbf{x}) = \|\mathbf{x}\|^2$). Finally, the update of the solution at line 11 can be carried out in constant time by the sparsity of $\mathbf{D}^*$. The time complexity of each iteration is thus given by $O(n \max(K, \log_2(1/\delta)))$.

The most costly part of the algorithm is the initialization (2–5) which has $O(n^2 K)$ time complexity. Hence, the overall complexity of the algorithm is $O(n^2 K + mn \times \max(K, \log_2(1/\delta)))$ where m is the number of required iterations, which is difficult to know in advance. As a rule of thumb, we need $m \in \Omega(nK)$ iterations to converge, because every entry of $\mathbf{Y}$ should be modified at least once. In that case the complexity is decided by the iterations only.

Finally, the stopping criterion ideally should test whether $\mathbf{D}^*$ is a descending direction. Indeed, if this does not hold then we know that $\mathbf{Y}^{(t)}$ is satisfying the KKT conditions (it follows from Proposition 4 in Appendix) and we can stop. In practice, we simply check if the quantity $g_J(\mathbf{Y}^{(t)})_V - g_J(\mathbf{Y}^{(t)})_U$ is below a given threshold τ and we stop if this happens. Indeed, if that quantity is precisely zero, then $\mathbf{Y}^{(t)}$ satisfies the KKT conditions. Additionally, we put an upper bound to the number of iterations.

5.4 A note on scalability

In applications where the number of data points to cluster is very large, the computation of the whole co-association matrix becomes impossible. In this cases one resorts to sparsifying the co-association matrix by keeping a number of entries that scales linearly with the number of data points.

Our algorithm can be easily adapted to deal with sparse co-association matrices. Assume that $\mathcal{P}$ contains only a sparse set of observable data point pairs. Let ℓ be the expected average number of entries of $\mathcal{P}_i$, *i.e.*, $\ell = \sum_{i \in \mathcal{O}} |\mathcal{P}_i|/n$ and assume that the input quantities c_{ij}'s and N_{ij}'s are given only for the pairs $\{i, j\} \in \mathcal{P}$. Since we need to know the value of $\mathbf{y}_i^\top \mathbf{y}_j$ again only for pairs of data points in $\mathcal{P}$, the computation of $\mathbf{A}^{(0)}$ is not fully required and only the entries indexed by $\mathcal{P}$ should be computed. This reduces to $O(K\ell n)$ the complexity of line 2 of Algorithm 1, where $\ell \ll n$. The same complexity characterizes the initialization of the gradient at lines 3–5. The subsequent updates of matrix $\mathbf{A}^{(t)}$ at line 10 and of the gradient at lines 12–15 require only $O(\ell)$ and $O(K\,\ell)$ operations, respectively. By adopting a priority queue (*e.g.*, heap based), the computation of the optimal direction in terms of U, V and J at line 8 requires only an overall complexity of $O(K \log_2(n))$ per iteration. This

Algorithm 1 Probabilistic Consensus Clustering (PCC)

```
Require: $\mathcal{E}$: ensemble of clusterings
Require: $\phi : \Delta_2 \to \mathbb{R}$ parameter of the Bregman divergence
Require: $\mathbf{Y}^{(0)} \in \Delta_K^n$: starting point
 1: Compute $\mathbf{C}$ and $\{N_{ij}\}$ from $\mathcal{E}$ as described in Sect. 3
 2: Initialize $\mathbf{A}^{(0)} \leftarrow \mathbf{Y}^{(0)\top}\mathbf{Y}^{(0)}$
 3: for all $i \in \mathcal{O}$ do
 4:   $g_i^{(0)} \leftarrow \sum_{j\in\mathcal{P}_i} N_{ij}\mathbf{y}_j^{(0)} \frac{\partial d_\phi}{\partial x_2}(\frac{c_{ij}}{N_{ij}}, a_{ij}^{(0)})$
 5: end for
 6: $t \leftarrow 0$
 7: repeat
 8:   Compute $\mathbf{D}^*$ at $\mathbf{Y}^{(t)}$ as described in Sect. 5.1
 9:   Compute $\epsilon^*$ as described in Sect. 5.2
10:   Update $\mathbf{A}^{(t+1)} = \mathbf{Y}^{(t+1)\top}\mathbf{Y}^{(t+1)} = \mathbf{A}^{(t)} + \epsilon^*(\mathbf{D}^{*\top}\mathbf{Y}^{(t)} + \mathbf{Y}^{(t)\top}\mathbf{D}^* + \epsilon^*\mathbf{D}^{*\top}\mathbf{D}^*)$.
11:   Update $\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \epsilon^*\mathbf{D}^*$
12:   for all $i \in \mathcal{O} : J \in \mathcal{P}_i$ do
13:     $g_i^{(t+1)} = g_i^{(t)} + N_{iJ}[\mathbf{y}_J^{(t+1)} \frac{\partial d_\phi}{\partial x_2}(\frac{c_{iJ}}{N_{iJ}}, a_{iJ}^{(t+1)}) - \mathbf{y}_J^{(t)} \frac{\partial d_\phi}{\partial x_2}(\frac{c_{iJ}}{N_{iJ}}, a_{iJ}^{(t)})]$
14:   end for
15:   $g_J^{(t+1)} \leftarrow \sum_{i\in\mathcal{P}_J} N_{Ji}\mathbf{y}_i^{(t+1)} \frac{\partial d_\phi}{\partial x_2}(\frac{c_{Ji}}{N_{Ji}}, a_{Ji}^{(t+1)})$
16:   $t \leftarrow t+1$
17: until stopping criterion met
18: return $\mathbf{Y}^{(t)}$
```

can be achieved by initially storing in the priority queue the best values of U and V for all $i \in \mathcal{O}$ and by updating the priorities based on the sparse changes in the gradient values. The optimal step at line 9 can be computed in $O(\ell \log_2(1/\delta))$, where δ is the tolerance for the dichotomic search. Finally, the update of $\mathbf{Y}$ remains with a constant complexity. The overall per-iteration complexity becomes $O(\max(\ell \log_2(1/\delta), K \log_2(n)))$. As for the number of iterations the considerations made in Sect. 5.3 still hold.

6 Related work

Several consensus methods have been proposed in the literature (Fred 2001; Strehl and Ghosh 2003; Fred and Jain 2005; Topchy et al. 2004; Dimitriadou et al. 2002; Ayad and Kamel 2008; Fern and Brodley 2004). Some of these methods are based on the similarity between data points, which is induced by the clustering ensemble, others are based on estimates of similarity between partitions and others cast the problem as a categorical clustering problem. All these methods tend to reveal a more robust and stable clustering solution than the individual clusterings used as input for the problem. A very recent survey can be found in Ghosh et al. (2011).

Strehl and Ghosh (2003) formulated the clustering ensemble problem as an optimization problem based on the maximal average mutual information between the optimal combined clustering and the clustering ensemble, presenting three algorithms to solve it, exploring graph theoretical concepts. The first one, entitled Cluster-based Similarity Partitioning Algorithm (CSPA), uses a graph partitioning algorithm, METIS (Karypis and Kumar 1998), for extracting a consensus partition from the co-association matrix. The second and third algorithms, Hyper Graph Partitioning Algorithm (HGPA) and Meta CLustering Algorithm

(MCLA), respectively, are based on hyper-graphs, where vertices correspond to data points, and the hyper-edges, which allow the connection of several vertices, correspond to clusters of the Clustering ensemble. HGPA obtains the consensus solution using an hyper-graph partitioning algorithm, HMETIS (Karypis et al. 1997); MCLA, uses another heuristic which allows clustering clusters.

Fern and Brodley (2004) reduce the problem to graph partitioning. The proposed model, entitled Hybrid Bipartite Graph Formulation (HBGF), uses as vertices both instances and clusters of the ensemble, retaining all of the information provided by the clustering ensemble, and allowing to consider the similarity among instances and among clusters. The partitioning of this bipartite graph is produced using the multi-way spectral graph partitioning algorithm proposed by Ng et al. (2001), which optimizes the normalized cut criterion (Shi and Malik 2000), or, as alternative, the graph partitioning algorithm METIS (Karypis and Kumar 1998).

These approaches were later extended by Punera and Ghosh (2007, 2008), to allow soft base clusterings on the clustering ensemble, showing that the addition of information on the ensemble is useful; the proposed models were the soft version of CSPA, of MCLA, and HBGF. Additionally they proposed to use information theoretic K-means (Dhillon et al. 2003), an algorithm very similar to K-means, differing only in the distance measure, using KL-divergence, for clustering in the feature space obtained from concatenating all the posteriors from the ensemble.

Topchy *et al.* (2003, 2004, 2005) proposed two different formulations, both derived from similarities between the partitions in the ensemble, rather than similarities between data points, differently from the case of co-association based approaches. The first one is a multinomial mixture model (MM) over the labels of the clustering ensemble, thus each partition is considered as a feature with categorical attributes. The second one is based on the notion of median partition and is entitled Quadratic Mutual Information Algorithm (QMI). The median partition is defined as the partition that best summarizes the partitions of the ensemble.

Wang *et al.* (2009, 2011) extended this idea, introducing a Bayesian version of the multinomial mixture model, the Bayesian cluster ensembles (BCE). Although the posterior distribution cannot be calculated in closed form, it is approximated using variational inference and Gibbs sampling, in a very similar procedure as in *latent Dirichlet allocation* (LDA) models (Griffiths and Steyvers 2004; Steyvers and Griffiths 2007), but applied to a different input feature space, the feature space of the labels of the ensembles. In Wang et al. (2010), a nonparametric version of BCE was proposed.

Ayad and Kamel (2008), followed Dimitriadou et al. (2002), proposed the idea of cumulative voting as a solution for the problem of aligning the cluster labels. Each clustering of the ensemble is transformed into a probabilistic representation with respect to a common reference clustering. Three voting schemes are presented: Un-normalized fixed-Reference Cumulative Voting (URCV), fixed-Reference Cumulative Voting (RCV), and Adaptive Cumulative Voting (ACV).

Lourenço *et al.* (2011), modelled the problem of consensus extraction taking as input space pairwise information, and using a generative aspect model for dyadic data. The extraction of a consensus solutions is found by solving a maximum likelihood estimation problem, using the Expectation-Maximization (EM) algorithm.

Our framework is also related to Non-negative Matrix Factorization (Paatero and Tapper 1994; Lee and Seung 2000), which is the problem of approximatively factorizing a given matrix $\mathbf{M}$, with two entrywise non-negative matrices $\mathbf{F}$ and $\mathbf{G}$, so that $\mathbf{M} \approx \mathbf{F}\,\mathbf{G}$. Indeed our formulation can be regarded as a kind of matrix factorization of the co-association matrix in terms of matrix $\mathbf{Y}^\top\mathbf{Y}$ under the constraint that $\mathbf{Y}$ is column stochastic. This particular setting has been considered, for the ℓ_2 norm, in Arora et al. (2011) and in Nepusz et al. (2008).

7 Experimental results

In this section we evaluate our formulation using synthetic datasets and real-world datasets from the UCI Irvine and UCI KDD Machine Learning Repository. We performed four different series of experiments: (i) we study the convergence properties of our algorithm on synthetically generated co-association matrices, (ii) we compare the consensus clustering obtained on different datasets with the known, crisp, ground truth partitions using standard evaluation criteria and we compare against other consensus clustering approaches, (iii) we perform an experiment on a large-scale dataset with incomplete partitions in the ensemble, (iv) we perform a qualitative analysis of a real-world dataset by deriving additional information from the probabilistic output of our algorithm.

We evaluate the performance of our Probabilistic Consensus Clustering (PCC) algorithm with KL-divergence (PCC-KL) and with squared ℓ_2 divergence (PCC-ℓ_2). From the quantitative perspective, we compare the performance of PCC-ℓ_2 and PCC-KL against other state-of-the-art consensus algorithms: the classical EAC algorithm using as extraction criteria the hierarchical agglomerative single-link (EAC-SL) and average-link (EAC-AL) algorithms; Cluster-based Similarity Partitioning Algorithm (CSPA) (Strehl and Ghosh 2003); Hybrid Bipartite Graph Formulation (HBGF) (Fern and Brodley 2004); Mixture Model (MM) (Topchy et al. 2004, 2005); Quadratic Mutual Information Algorithm (QMI) (Topchy et al. 2003, 2005).

In order to evaluate the quality of a consensus clustering result against a hard, ground truth partition we convert our probabilistic assignments in hard assignments according to a maximum likelihood criterion. We compare then two hard clusterings $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ and $\mathcal{Q} = \{\mathcal{Q}_1, \ldots, \mathcal{Q}_k\}$ using the $\mathcal{H}$ criterion based on cluster matching (Meila 2003) and the Adjusted-Rand index (Jain and Dubes 1988), which is based on counting pairs. Note that we assume without loss of generality that $\mathcal{P}$ and $\mathcal{Q}$ have the same number of elements, since we can add empty clusters where needed. The $\mathcal{H}$ criterion (Meila 2003) gives the accuracy of the partitions and is obtained by finding the optimal one-to-one matching between the clusters in $\mathcal{P}$ with the ground truth labels in $\mathcal{Q}$:

$$\mathcal{H}(\mathcal{P}, \mathcal{Q}) = \frac{1}{n} \max_{\mathbf{v}} \sum_{j=1}^{k} |\mathcal{P}_j \cap \mathcal{Q}_{v_j}|, \tag{11}$$

where the vector $\mathbf{v}$ of the maximization runs over all possible permutations of the vector $(1, \ldots, k)$.

When we have a soft, ground truth partition given in terms of a probabilistic assignment $\mathbf{Z} \in \Delta_k^n$, we evaluate the divergence between a soft consensus partition $\mathbf{Y} \in \Delta_k^n$ and $\mathbf{Z}$ in terms of the Jensen-Shannon (JS) divergence. In more details, let $D_{JS}(\cdot\|\cdot)$ denote the JS-divergence between two distributions given as points of Δ_k. Then the divergence between $\mathbf{Z}$ and $\mathbf{Y}$ is given by

$$\mathcal{J}(\mathbf{Y}, \mathbf{Z}) = \frac{1}{n} \min_{\mathbf{P}} \sum_{i=1}^{n} D_{JS}(\mathbf{z}_i \| \mathbf{P}\mathbf{y}_i),$$

where the matrix $\mathbf{P}$ in the minimization runs over all possible $k \times k$ permutation matrices. Similarly to the case of hard partitions, we assume without loss of generality that $\mathbf{Z}$ and $\mathbf{Y}$

have the same number of rows, since we can eventually add zero rows to fill the gap. Note that $0 \le \mathcal{J}(\mathbf{Y}, \mathbf{Z}) \le 1$ holds for any $\mathbf{Y}, \mathbf{Z} \in \Delta_K^n$.

For the qualitative experiments, we analyse the probabilistic assignments $\mathbf{Y}$ in order to exploit the information about the cluster uncertainty. For this analysis, we remove probability values, lower than a predefined threshold δ, and then we normalize each column to sum again to one. We measure the normalized similarity between two clusters i, j as the expected value of common elements over the expected cardinality of the ith clusters:

$$m_{ij} = \frac{\mathbb{E}_{\mathbf{Y}}[|\mathcal{P}_i \cap \mathcal{Q}_j|]}{\mathbb{E}_{\mathbf{Y}}[|\mathcal{P}_i|]}.$$

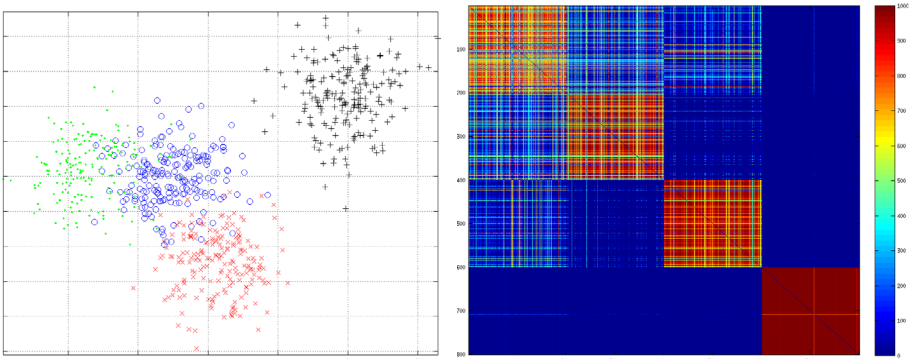
Given a set of data points $\{\mathbf{x}_i\}_{i=1}^n$, we define the centroid of class k according to $\mathbf{Y}$ as

$$\boldsymbol{\mu}_k = \frac{\sum_j y_{kj} \vec{x}_j}{\mathbb{E}_{\mathbf{Y}}[|\mathcal{P}_k|]}.$$

Given the weighted matrix $\mathbf{M} = [m_{ij}]$, which is usually sparse, and the centroids, we can visualize the obtained clusters and their relationships simply by drawing them in the plane with a weighted graph. The structures found in these graphs, like paths or cliques, highly depend on the type of data and the geometry of the consensus set, leading to a different and interesting way of interpreting the consensus results.

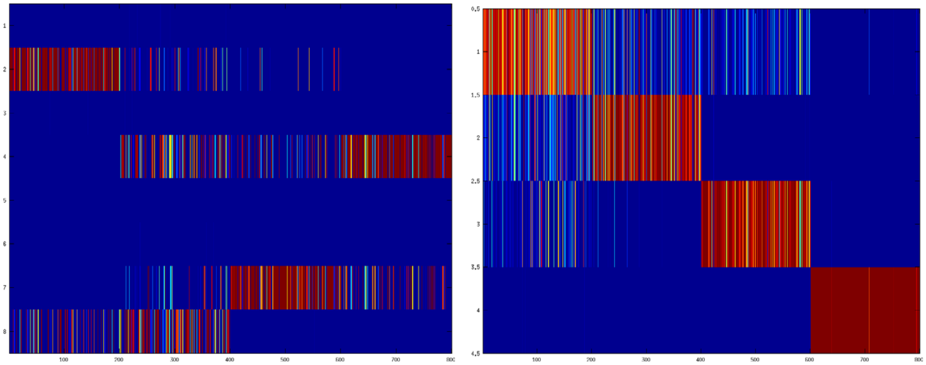
7.1 Simulated data

We first study the proposed formulation using a synthetic experiment with soft partitions as ground truth. A soft partition $\mathbf{Y}^*$ is determined by generating 4 isotropic, bivariate, planar Gaussian distributions, each consisting of 200 data points, with mean vectors randomly selected in the four orthants, and by computing for each point the normalized probability of having been generated by one of the 4 Gaussians. Given a soft partition $\mathbf{Y}^*$ we artificially generated an ensemble by randomly sampling $N = 1000$ hard partitions with cluster assignments determined by $\mathbf{Y}^*$ and we constructed the corresponding co-association matrices. Figure 1(a) illustrates one example of such a dataset, where there is some overlap between



(a) Dataset (b) Co-association matrix **C**

Fig. 1 Experiment with a 4-component bivariate Gaussian mixture: (**a**) an example of a dataset and (**b**) the corresponding synthetically generated co-association matrix



(a) $\mathbf{Y}$ (b) $\mathbf{Y}^*$

Fig. 2 4-component bivariate Gaussian mixture: $\mathbf{Y}$, estimated cluster assignments and $\mathbf{Y}^*$, the ground truth cluster assignments

the components, and Fig. 1(b) shows the corresponding co-association matrix. We generated 10 different datasets according to the aforementioned procedure.

For each dataset we run our PCC-KL and PCC-ℓ_2 algorithms with the purpose of recovering the ground truth soft partition $\mathbf{Y}^*$. Although the optimal number of clusters is $K = 4$, we run our algorithms with a larger value of $K = 8$. This is not a problem as our formulation can automatically tune itself to select a smaller number of clusters. Indeed, we can see from Fig. 2(a) the estimated cluster assignments $\mathbf{Y}$ corresponding to the dataset in Fig. 1, where only 4 components have significant probabilities thus confirming our previous claim. We evaluated the divergence between the ground truth soft partition and the recovered one by our algorithms on each of the 10 datasets using the $\mathcal{J}$-criterion introduced at the beginning of Sect. 7. Both our algorithms obtained an average divergence of 0.0012 and a standard deviation of ±0.00005, which indicate a good recovery of the ground truth probabilistic cluster assignments.

7.2 UCI and synthetic data

We followed the usual strategy of producing clustering ensembles and combining them using the co-association matrix. Two different types of ensembles were created: (1) using k-means with random initialization and random number of clusters (Lourenço et al. 2010), splitting natural clusters intro micro-blocks; (2) combining multiple algorithms (agglomerative hierarchical algorithms: single, average, ward, centroid link; k-means Jain and Dubes 1988; spectral clustering Ng et al. 2001) with different number of clusters, inducing block-wise co-association matrices.

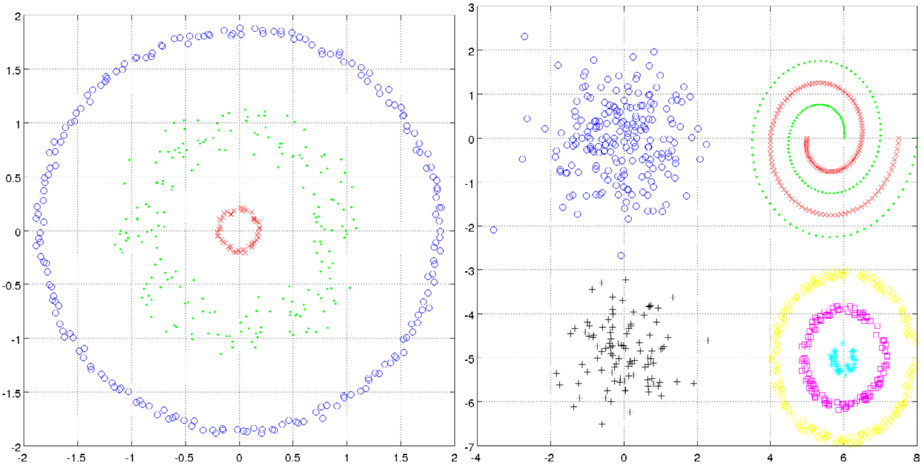
Table 3 summarizes the main characteristics of the UCI and synthetic datasets used in the evaluation, and the parameters used for generating ensemble (2). Figure 3 illustrates the synthetic datasets used in the evaluation: (a) rings; (b) image-1.

We summarized the performance of both algorithms after several runs, accounting for possible different solutions due to initialization, in terms of $\mathcal{H}$ and Adjusted Rand criteria, in Tables 4, 5 and 6, 7, respectively. We present for both validation indices, the average performance (avg), the standard deviation (std), maximum value (max), and minimum value (min), highlighting in bold the best results for each data-set.

The performance of PCC-KL and PCC-ℓ_2 depends on the type of ensemble. On Ensemble (1), PCC-KL and PCC-ℓ_2, have generally lower performance when compared with EAC

Table 3 Benchmark datasets

Data-Sets	K	n	Ensemble
			K_i
(s–1) rings	3	450	2–8
(s–2) image-1	8	1000	8–15, 20, 30
(r–1) iris	3	150	3–10, 15, 20
(r–2) wine	3	178	4–10, 15, 20
(r–3) house-votes	2	232	4–10, 15, 20
(r–4) ionsphere	2	351	4–10, 15, 20
(r–5) std-yeast-cell	5	384	5–10, 15, 20
(r–6) breast-cancer	2	683	3–10, 15, 20
(r–7) optdigits	10	1000	10, 12, 15, 20, 35, 50



(a) (s-1) (b) (s-2)

Fig. 3 Sketch of the Synthetic Data Sets

and CSPA (both on Adjusted-Rand Index and *CI*), that seem very suitable for this kind of ensembles. Nevertheless, on the UCI datasets, both obtain promising results: PCC-ℓ_2 is the best in 1 (over 9) dataset, and PCC-KL the best in 1 (over 9) and is very close to the best consensus in several situations. On ensemble (2), PCC-KL obtains the best results almost on all data-sets, 7 (over 9).

Its also very important to notice that the standard deviation of the proposed methods is very low, being in almost every datasets very close to zero.

Figure 4 shows examples of obtained co-association matrices, where the matrices have been reordered according to VAT algorithm (Bezdek and Hathaway 2002), to highlight the clustering structure. Its color scheme ranges from black (c_{ij} =0) to white ($c_{ij} = N_{ij}$), corresponding to the magnitude of similarity. As is possible to see, the co-association of ensemble (1), has not a so evident blockwise structure, since it was produced splitting of natural clusters into smaller clusters inducing micro-blocks in the co-association matrix; on Ensembles (2), the co-association matrices has a much more blockwise form, as it was generated

Table 4 Results from experiments conducted with an ensemble of type 1 evaluated with criterion $\mathcal{H}$. See Sect. 7.2 for details

Alg		s–1	s–2	r–1	r–2	r–3	r–4	r–5	r–6	r–7
PCC-KL	avg	0.53	0.57	0.86	0.96	0.89	0.54	0.92	0.63	**0.87**
	std	0.02	0.03	0.10	0.00	0.01	0.00	0.11	0.00	0.04
	max	0.55	0.61	0.91	0.96	0.90	0.54	0.97	0.63	0.90
	min	0.51	0.55	0.69	0.96	0.88	0.54	0.73	0.63	0.80
PCC-ℓ_2	avg	0.57	0.46	0.71	**0.97**	0.88	0.54	0.74	0.60	**0.87**
	std	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.03
	max	0.57	0.48	0.71	0.97	0.88	0.54	0.74	0.62	0.89
	min	0.57	0.43	0.71	0.97	0.88	0.54	0.74	0.57	0.82
QMI	avg	0.52	0.39	0.55	0.67	0.77	0.43	0.73	0.64	0.36
	std	0.13	0.03	0.10	0.18	0.16	0.11	0.24	0.07	0.10
	max	0.75	0.43	0.72	0.96	0.93	0.57	0.97	0.75	0.49
	min	0.44	0.36	0.46	0.53	0.53	0.30	0.41	0.57	0.24
MM	avg	0.54	0.34	0.61	0.65	0.70	0.38	0.78	0.62	0.57
	std	0.02	0.03	0.08	0.13	0.11	0.07	0.18	0.12	0.05
	max	0.57	0.39	0.73	0.85	0.85	0.48	0.95	0.75	0.63
	min	0.51	0.30	0.52	0.52	0.54	0.32	0.54	0.48	0.49
HBGF	avg	0.50	0.37	0.66	0.71	0.58	0.44	0.68	0.65	0.49
	std	0.11	0.07	0.12	0.18	0.06	0.07	0.13	0.07	0.05
	max	0.64	0.47	0.84	0.96	0.66	0.50	0.86	0.71	0.56
	min	0.35	0.29	0.55	0.49	0.53	0.34	0.54	0.53	0.43
EAC	SL	**1.00**	**0.67**	0.75	0.67	0.67	0.53	0.66	0.67	0.62
	AL	**1.00**	0.59	0.89	0.93	0.87	**0.69**	**0.97**	0.54	0.80
	WL	0.73	0.47	0.89	0.96	0.85	0.54	0.73	0.61	0.90
CSPA		0.78	0.49	**0.97**	0.92	**0.93**	0.52	0.85	**0.68**	**0.87**

with a combination of several algorithms with numbers of clusters ranging from small to large. The results show that blockwise matrices are very adequate for the proposed model, even in cases where there is much overlap.

7.3 A large-scale experiment

In order to show that our algorithm can be used also on large-scale datasets we propose here an experiment on a KKD Cup 1999 dataset[1]. From the available datasets we analysed a subset of “kddcup.data10percent”, consisting of 120.000 data points characterized by 41 attributes distributed in 3 classes. The preprocessing consisted in standardizing numerical features, and discretizing categorical features, arriving to a 39-dimensional feature space. We produced an ensemble consisting of 100 K-means partitions obtained on random subsets of the dataset (sampling rate 50 %) with random initializations and random number

[1] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Table 5 Results from experiments conducted with an ensemble of type 2 evaluated with criterion $\mathcal{H}$. See Sect. 7.2 for details

Alg		s–1	s–2	r–1	r–2	r–3	r–4	r–5	r–6	r–7
PCC-KL	avg	**0.71**	**0.71**	**0.97**	**0.97**	0.91	**0.69**	**0.97**	**0.73**	0.61
	std	0.03	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.02
	max	0.75	0.73	0.97	0.97	0.91	0.69	0.97	0.73	0.64
	min	0.69	0.68	0.97	0.97	0.91	0.69	0.97	0.73	0.58
PCC-ℓ_2	avg	0.68	0.63	0.95	**0.97**	0.91	0.68	0.96	**0.73**	0.60
	std	0.00	0.04	0.02	0.00	0.00	0.00	0.03	0.00	0.01
	max	0.68	0.67	0.97	0.97	0.91	0.68	0.97	0.73	0.62
	min	0.68	0.57	0.93	0.97	0.91	0.68	0.91	0.72	0.59
QMI	avg	0.49	0.30	0.44	0.59	0.75	0.52	0.84	0.69	0.24
	std	0.11	0.11	0.11	0.11	0.20	0.12	0.17	0.05	0.15
	max	0.62	0.48	0.62	0.65	0.91	0.67	0.97	0.73	0.50
	min	0.38	0.23	0.33	0.40	0.53	0.35	0.65	0.64	0.15
MM	avg	0.53	0.38	0.61	0.57	0.63	0.48	0.85	0.67	0.51
	std	0.06	0.06	0.16	0.07	0.12	0.05	0.09	0.03	0.04
	max	0.64	0.46	0.87	0.64	0.80	0.55	0.96	0.70	0.54
	min	0.50	0.30	0.46	0.48	0.53	0.43	0.76	0.63	0.44
HBGF	avg	0.43	0.27	0.48	0.55	0.62	0.38	0.74	0.62	0.50
	std	0.04	0.02	0.07	0.11	0.03	0.05	0.15	0.10	0.06
	max	0.46	0.30	0.56	0.70	0.68	0.46	0.88	0.77	0.57
	min	0.37	0.25	0.39	0.39	0.58	0.34	0.54	0.54	0.41
EAC	SL	0.63	0.71	0.69	0.39	0.52	0.36	0.65	0.64	0.32
	AL	0.38	0.71	**0.97**	0.39	0.91	0.36	0.66	0.65	0.42
	WL	0.39	0.59	0.93	0.93	0.88	0.70	0.96	0.65	0.76
CSPA		0.68	0.56	0.96	0.93	**0.92**	0.53	0.85	0.66	**0.78**

of clusters ($2 \leq K \leq 10$). Since the ensemble is composed by incomplete partitions, the consensus clustering phase becomes more challenging.

In order to cope with the large amount of data points, which renders the construction of the co-association matrix impossible both from a space and computational time perspective, we run a sparsified version of our algorithm as described in Sect. 5.4. In the specific, we created $\mathcal{P}$ by sampling a share of 0.25 ‰ data points pairs from the available (around 8 billions) ones. Our algorithms (PCC-ℓ_2 and PCC-KL) were run with a maximum number of nK iterations. Our non-parallelized C implementations of PCC-ℓ_2 and PCC-KL took on average 13.8 s and 16.7 s, respectively, to deliver a solution on a dual-core 64-bits Pentium 2.8 GHz with 4 Gb RAM (only one core was effectively used). We were able to compare our algorithm only against CSPA, which nevertheless obtained competitive results in the previous set of experiments. All other approaches could not be run due to the large size of the dataset, or because of their inability of handling incomplete partitions in the ensemble.

Table 6 Results from experiments conducted with an ensemble of type 1 evaluated with the Adjusted-Rand index. See Sect. 7.2 for details

Alg		s–1	s–2	r–1	r–2	r–3	r–4	r–5	r–6	r–7
PCC-KL	avg	0.59	**0.86**	0.87	**0.95**	0.81	0.79	0.88	0.53	**0.96**
	std	0.01	0.01	0.05	0.00	0.02	0.00	0.15	0.00	0.01
	max	0.60	0.88	0.89	0.95	0.82	0.79	0.94	0.53	0.96
	min	0.58	0.85	0.78	0.95	0.79	0.78	0.61	0.53	0.95
PCC-ℓ_2	avg	0.60	0.83	0.78	**0.96**	0.78	0.78	0.61	0.52	**0.96**
	std	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.01
	max	0.60	0.84	0.78	0.96	0.78	0.78	0.61	0.53	0.96
	min	0.60	0.80	0.78	0.96	0.78	0.78	0.61	0.51	0.95
QMI	avg	0.52	0.61	0.65	0.75	0.69	0.61	0.70	0.54	0.68
	std	0.12	0.08	0.10	0.11	0.15	0.22	0.22	0.05	0.13
	max	0.66	0.68	0.78	0.95	0.86	0.75	0.94	0.63	0.84
	min	0.41	0.48	0.56	0.68	0.50	0.23	0.51	0.51	0.55
MM	avg	0.57	0.74	0.65	0.70	0.60	0.65	0.71	0.55	0.89
	std	0.05	0.01	0.07	0.10	0.09	0.08	0.18	0.06	0.01
	max	0.60	0.75	0.71	0.83	0.75	0.75	0.91	0.62	0.90
	min	0.49	0.72	0.55	0.54	0.50	0.53	0.50	0.50	0.88
HBGF	avg	0.57	0.75	0.71	0.76	0.52	0.71	0.60	0.55	0.87
	std	0.04	0.04	0.08	0.11	0.02	0.01	0.11	0.03	0.03
	max	0.63	0.78	0.83	0.95	0.55	0.72	0.75	0.59	0.90
	min	0.52	0.69	0.61	0.67	0.50	0.70	0.50	0.50	0.83
EAC	SL	**1.00**	**0.86**	0.79	0.73	0.55	0.70	0.55	0.55	0.89
	AL	**1.00**	**0.86**	0.88	0.90	0.77	**0.81**	**0.94**	0.50	0.95
	WL	0.66	0.84	0.88	0.95	0.75	0.79	0.61	0.52	**0.96**
CSPA		0.78	0.83	**0.97**	0.90	**0.86**	0.78	0.75	**0.56**	**0.96**

We report in Fig. 5 the results obtained in terms of accuracy ($\mathcal{H}$-criterion) by the algorithms at varying values of the parameter K. At the optimal number of clusters, $K = 3$, all approaches achieve their best score, but our approach outperforms CSPA both when the KL-divergence and the squared ℓ_2-divergence are used, the former being slightly better than the latter. Moreover, it turns out that our approach can automatically tune the optimal number of clusters thus being more robust to overestimations of the parameter K. Indeed, we remark that the parameter K for our approach is intended as a maximum number of clusters rather than the desired number of clusters that the algorithm must deliver. The partitions in the ensemble achieved on average an accuracy of 79 %. Clearly, due to the presence of incomplete partitions, this score has been computed by considering the data points that have effectively been used in each partition.

Our consensus solution provides a considerable improvement of this score, confirming the effectiveness of our algorithm also in the presence of incomplete partitions in the ensemble.

Table 7 Results from experiments conducted with an ensemble of type 2 evaluated with the Adjusted-Rand index. See Sect. 7.2 for details

Alg		s–1	s–2	r–1	r–2	r–3	r–4	r–5	r–6	r–7
PCC-KL	avg	**0.72**	**0.89**	**0.97**	**0.96**	0.83	**0.81**	**0.94**	**0.60**	0.85
	std	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
	max	0.77	0.89	0.97	0.96	0.83	0.81	0.94	0.60	0.87
	min	0.70	0.88	0.97	0.96	0.83	0.81	0.94	0.60	0.83
PCC-ℓ_2	avg	0.69	0.88	0.94	**0.96**	0.83	**0.81**	0.92	**0.60**	0.83
	std	0.00	0.01	0.03	0.00	0.00	0.00	0.05	0.00	0.01
	max	0.69	0.88	0.97	0.96	0.83	0.81	0.94	0.60	0.84
	min	0.69	0.87	0.92	0.96	0.83	0.81	0.84	0.60	0.82
QMI	avg	0.56	0.30	0.42	0.63	0.69	0.62	0.78	0.58	0.41
	std	0.06	0.22	0.09	0.16	0.17	0.23	0.21	0.03	0.30
	max	0.61	0.68	0.56	0.74	0.83	0.80	0.93	0.60	0.84
	min	0.49	0.17	0.33	0.34	0.50	0.23	0.54	0.54	0.19
MM	avg	0.58	0.76	0.65	0.63	0.55	0.71	0.75	0.56	0.86
	std	0.05	0.03	0.11	0.09	0.08	0.03	0.12	0.02	0.02
	max	0.63	0.81	0.84	0.72	0.68	0.75	0.92	0.58	0.88
	min	0.50	0.73	0.58	0.54	0.50	0.67	0.63	0.53	0.83
HBGF	avg	0.55	0.72	0.60	0.64	0.53	0.67	0.65	0.54	0.87
	std	0.02	0.01	0.05	0.05	0.02	0.03	0.13	0.06	0.01
	max	0.57	0.74	0.66	0.71	0.56	0.71	0.79	0.64	0.88
	min	0.52	0.70	0.52	0.58	0.51	0.63	0.50	0.50	0.85
EAC	SL	0.61	**0.89**	0.78	0.36	0.50	0.25	0.55	0.54	0.46
	AL	0.53	**0.89**	0.96	0.36	0.84	0.28	0.55	0.54	0.60
	WL	0.54	0.88	0.92	0.92	0.79	0.81	0.93	0.54	0.93
CSPA		**0.72**	0.84	0.95	0.91	**0.86**	0.78	0.74	0.55	**0.94**

In Fig. 6 we report the results in terms of accuracy, when varying the percentage of observed entries in the co-association matrix. The trend on the performance is constant with percentages larger than 0.08 ‰. By further reducing the number of observed entries, we experience a performance drop as one would expect.

7.4 Visualizing probabilistic relations

It was observed in Färber et al. (2010), Cui et al. (2010) that we can discover new structures in data using previous known information. A case study we consider here is that of the PenDigits (Frank and Asuncion 2012). The PenDigits dataset contains handwritten digits produced by different persons. Each digit is stored as a sequence of 8 (x, y) positions, collected at different time intervals during the execution of each single digit. A manual analysis of Cui et al. (2010) highlights that the same digit can be written in different ways, but this information is not contained in the ground truth which just collects each type of digit in the same class. These observations become apparent if we build a consensus matrix for each

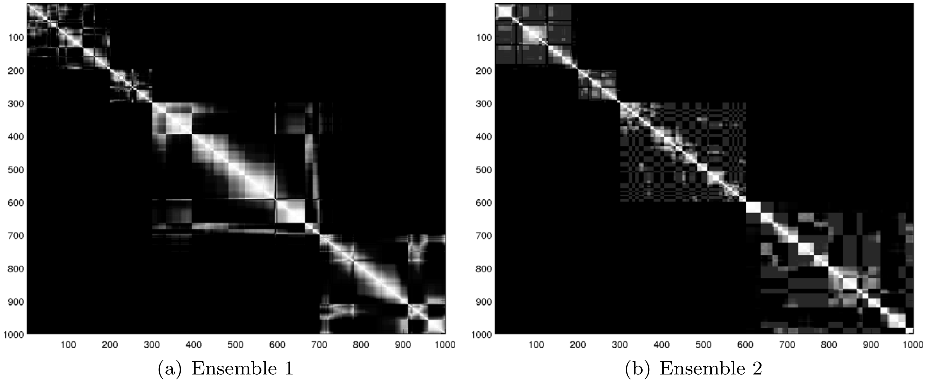


Fig. 4 Example of co-association matrices obtained with ensemble (1) and (2)—reordered using VAT (Bezdek and Hathaway 2002)

Fig. 5 Results obtained on a large-scale dataset. We report the accuracy obtained by the proposed algorithms (PCC-ℓ_2 and PCC-KL) and by a competing algorithm (CSPA) at varying values of the parameter K, the optimal one being $K = 3$

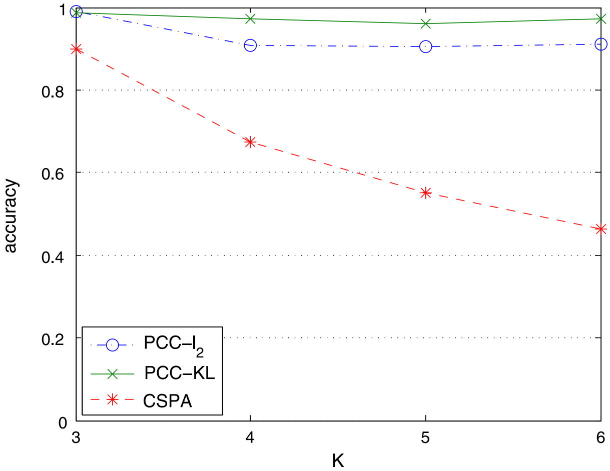
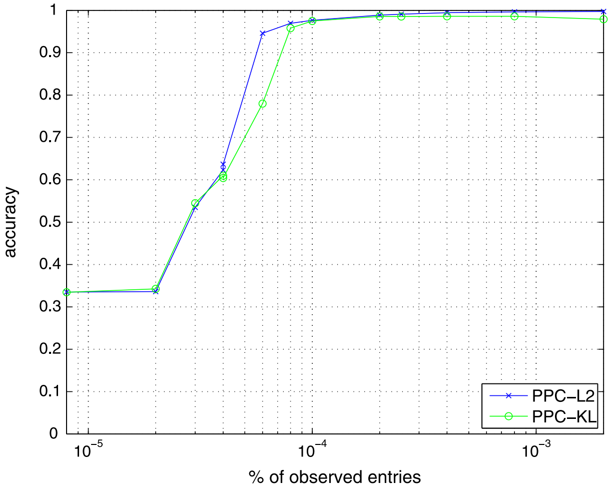
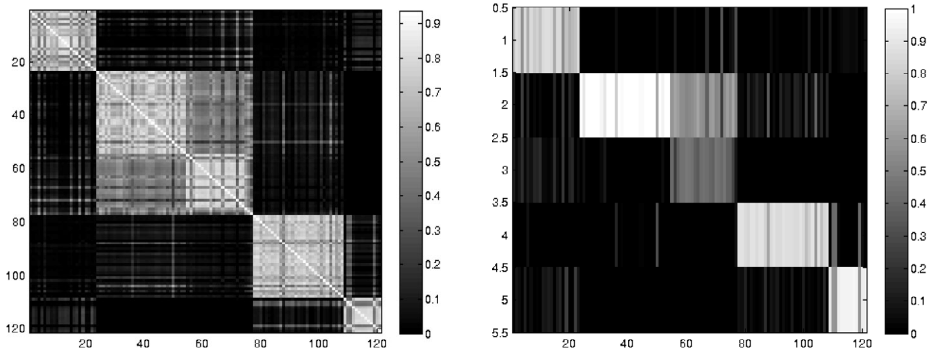


Fig. 6 Results obtained on a large-scale dataset. We report the accuracy obtained by the proposed algorithms (PCC-ℓ_2 and PCC-KL) varying the percentage of observed entries





(a) Co-association matrix for digit '2'

(b) **Y** matrix found with PCC-KL

Fig. 7 On the *left*, a co-association matrix is built on a subsample of the digit ‘2’ for the PenDigit dataset. Five blocks are present but they are not all clearly separated. On the *right*, resulting matrix **Y** obtained by running PCC-KL on the co-association matrix. The overlap of the second and third blocks is captured by the uncertainties in the second and third lines

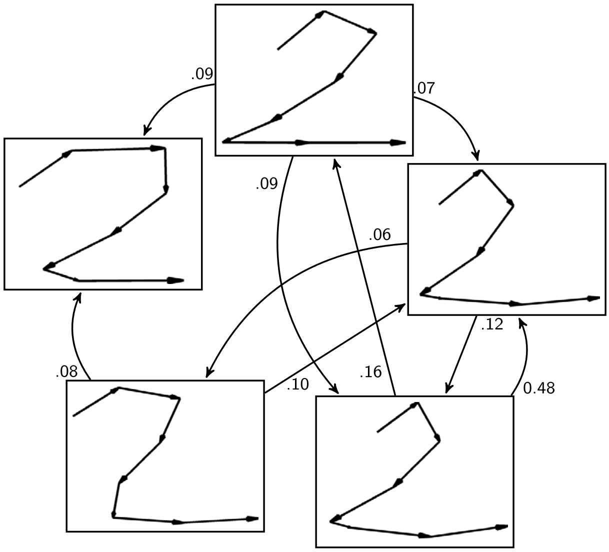


Fig. 8 Graph of the directed, weighted, relations among the centroids found in class ‘2’ of the PenDigit dataset. The first class is the *upper image*, other classes are ordered clockwise. Each centroid is made of 8 ordered points. The five centroids correspond to the five blocks of Fig. 7(a) and the five rows of Fig. 7(b). Each edge $i \sim j$ is weighted with m_{ij} (see introduction of Sect. 7). A strong dependence, 0.48, is present in the third class with the second and the first ones. This reflects the visual similarity of the respective centroids

digit taken in isolation and then visualize the obtained classes. Each digit has different ways to be written, but some of them are not completely different, so when building the consensus matrix we can see that these classes overlap. As an example we consider the digit ‘2’. In Fig. 7(a) we can see that the co-association matrix contains 5 blocks, two of which, the second and the third, are highly overlapped. The resulting matrix **Y** Fig. 7(b) reflects the overlap by assigning uncertainty to the two overlapping classes. The uncertainty is not symmetric, but the third class seems actually a subclass of the second. In Fig. 8 we can see the five centroids of the classes and their pairwise similarities. The centroids are ordered so that the upper image is the first class centroid and the others are in clockwise order. Each centroid visualizes eight points and the order in which they appear. The visualization of the centroids gives an explanation of the similarities/diversities that are numerically encoded on the edges, and in particular it is clear the dependence of class three with respect to class one and two.

8 Conclusions

In this paper, we introduced a new probabilistic consensus clustering formulation based on the EAC paradigm. Each entry of the co-association matrix, derived from the ensemble, is regarded as a Binomial random variable, parametrized by the unknown class assignments. We showed that the log-likelihood function corresponding to this model coincides with the KL divergence between the co-association relative frequencies and the co-occurrence probabilities parametrized by the Binomial random variables. This formulation can be seen as a special case of a more general setting, replacing the KL divergence with any Bregman divergence. We proposed an algorithm to find a consensus clustering solution according to our model, which works with any double-convex Bregman divergence. We also showed how the algorithm can be adapted to deal with large-scale datasets. Experiments on synthetic and real world datasets have demonstrated the effectiveness of our approach with ensembles composed by heterogeneous partitions obtained from multiple algorithms (agglomerative hierarchical algorithms, k-means, spectral clustering) with varying number of clusters. Additionally, we have shown that our algorithm is able to deal with large-scale datasets and can successfully be applied also in case of ensembles having incomplete partitions. On different datasets and ensembles, we outperformed the competing state-of-the-art algorithms and showed particularly outstanding results on the large-scale experiment. The qualitative analysis of the probabilistic consensus solutions provided some evidences that the proposed formulation can discover new structures in data. For the PenDigits dataset, we showed visual relationships between overlapping clusters representing the same digit, using the centroids of each cluster and similarities between clusters obtained from the probabilities of the consensus solution.

Acknowledgements This work was partially financed by an ERCIM "Alain Bensoussan" Fellowship Programme under the European Union Seventh Framework Programme (FP7/2007-2013), grant agreement n. 246016, by *Fundação para a Ciência e Tecnologia*, under grants PTDC/EIACCO/103230/2008, SFRH/PROTEC/ 49512/2009 and PEst-OE/EEI/LA0008/2011, and by the *Área Departamental de Engenharia Electronica e Telecomunicações e de Computadores* of *Instituto Superior de Engenharia de Lisboa*, whose support the authors gratefully acknowledge.

Appendix: Proof of results

Proposition 1 *Let* $\phi(\mathbf{x}) = -H(\mathbf{x})$. *Maximizers of* (1) *are minimizers of* (2) *and vice versa.*

Proof We have that for all $i, j \in \mathcal{O}$, $i \neq j$,

$$\begin{aligned}
&c_{ij}\log\left(\mathbf{y}_i^\top\mathbf{y}_j\right) + (N_{ij}-c_{ij})\log\left(1-\mathbf{y}_i^\top\mathbf{y}_j\right)\\
&= -N_{ij}\left[\frac{c_{ij}}{N_{ij}}\log\left(\frac{\frac{c_{ij}}{N_{ij}}}{\mathbf{y}_i^\top\mathbf{y}_j}\right) + \left(1-\frac{c_{ij}}{N_{ij}}\right)\log\left(\frac{1-\frac{c_{ij}}{N_{ij}}}{1-\mathbf{y}_i^\top\mathbf{y}_j}\right)\right]\\
&\quad + N_{ij}\left[\frac{c_{ij}}{N_{ij}}\log\left(\frac{c_{ij}}{N_{ij}}\right) + \left(1-\frac{c_{ij}}{N_{ij}}\right)\log\left(1-\frac{c_{ij}}{N_{ij}}\right)\right]\\
&= -N_{ij}D_{KL}\left(\psi\left(\frac{c_{ij}}{N_{ij}}\right)\middle\|\,\psi\left(\mathbf{y}_i^\top\mathbf{y}_j\right)\right) - N_{ij}H\left(\psi\left(\frac{c_{ij}}{N_{ij}}\right)\right)
\end{aligned}$$

$$= -N_{ij} B_\phi\left(\psi\left(\frac{c_{ij}}{N_{ij}}\right)\middle\|\psi\left(\mathbf{y}_i^\top \mathbf{y}_j\right)\right) - N_{ij} H\left(\psi\left(\frac{c_{ij}}{N_{ij}}\right)\right)$$

$$= -N_{ij} d_\phi\left(\frac{c_{ij}}{N_{ij}}, \mathbf{y}_i^\top \mathbf{y}_j\right) - N_{ij} H\left(\psi\left(\frac{c_{ij}}{N_{ij}}\right)\right)$$

where $\psi(x) = (x, 1-x)^\top$.

The result follows from

$$\begin{aligned}\operatorname*{arg\,max}_{\mathbf{Y}\in\Delta_K^n} f(\mathbf{Y}) &= \arg\max_{\mathbf{Y}\in\Delta_K^n}\left\{\textstyle\sum_{\{i,j\}\in\mathcal{P}} -N_{ij} d_\phi\left(\frac{c_{ij}}{N_{ij}}, \mathbf{y}_i^\top \mathbf{y}_j\right) - N_{ij} H\left(\psi\left(\frac{c_{ij}}{N_{ij}}\right)\right)\right\} \\ &= \arg\min_{\mathbf{Y}\in\Delta_K^n}\left\{\textstyle\sum_{\{i,j\}\in\mathcal{P}} N_{ij} d_\phi\left(\frac{c_{ij}}{N_{ij}}, \mathbf{y}_i^\top \mathbf{y}_j\right)\right\}.\end{aligned}$$ □

Proposition 2 *Any search direction* $\mathbf{D} \in \mathcal{D}(\mathbf{Y})$ *is feasible for* (2).

Proof Let $\mathbf{D} = (\mathbf{e}_K^u - \mathbf{e}_K^v)(\mathbf{e}_n^j)^\top \in \mathcal{D}(\mathbf{Y})$ and $\mathbf{Z}_\epsilon = \mathbf{Y} + \epsilon\mathbf{D}$. For any ϵ,

$$\mathbf{Z}_\epsilon^\top \mathbf{e}_K = (\mathbf{Y} + \epsilon\,\mathbf{D})^\top \mathbf{e}_K = \mathbf{Y}^\top \mathbf{e}_K + \epsilon\,\mathbf{D}^\top \mathbf{e}_K = \mathbf{e}_n + \epsilon\,\mathbf{e}_n^j\left(\mathbf{e}_K^u - \mathbf{e}_K^v\right)^\top \mathbf{e}_K = \mathbf{e}_n.$$

As ϵ increases, only the (v, j)th entry of $\mathbf{Z}_\epsilon$, which is given by $y_{vj} - \epsilon$, decreases. This entry is non-negative for all values of ϵ satisfying $\epsilon \le y_{vj}$. Hence, $\mathbf{Z}_\epsilon \in \Delta_K^n$ for all sufficiently small positive values of ϵ. □

Proposition 3 *A solution to* (5) *is*

$$\mathbf{D}^* = \left(\mathbf{e}_K^U - \mathbf{e}_K^V\right)\left(\mathbf{e}_n^J\right)^\top,$$

where J *is given as* (8), $U = U_J$ *and* $V = V_J$ (U_i *and* V_i *defined as* (6)).

Proof Let $\mathcal{I}(\mathbf{Y})$ be a set of triplets of indices given by

$$\mathcal{I}(\mathbf{Y}) = \left\{(j, u, v) : j \in \mathcal{O}, u \in \{1, \ldots, K\}, v \in \sigma(\mathbf{y}_j), u \neq v\right\}.$$

Optimization problem (5) can be rewritten as follows by exploiting the definition of $\mathcal{D}(\mathbf{Y})$:

$$(J, U, V) \in \operatorname*{arg\,min}_{(j,u,v)\in\mathcal{I}(\mathbf{Y})}\left\{\left(\mathbf{e}_K^u - \mathbf{e}_K^v\right)^\top g_j(\mathbf{Y})\right\}$$

and $\mathbf{D}^* = (\mathbf{e}_K^U - \mathbf{e}_K^V)(\mathbf{e}_n^J)^\top$. Here, J can be further characterized as the solution to

$$J \in \operatorname*{arg\,min}_{j\in\mathcal{O}}\left\{\left(\min_{u\in\{1,\ldots,K\}}\left[g_j(\mathbf{Y})\right]_u\right) - \left(\max_{v\in\sigma(\mathbf{y}_j)}\left[g_j(\mathbf{Y})\right]_v\right)\right\}.$$

The result follows by exploiting the definition of U_i and V_i in (6). □

Proposition 4 *If* $\mathbf{Y} \in \Delta_K^n$ *does not satisfy the KKT first-order necessary conditions for* (2) *then the search direction* $\mathbf{D}^*$ *at* $\mathbf{Y}$, *which is solution to* (5), *is descending.*

Proof To prove the result we have to show that $f(\mathbf{Y}+\epsilon\mathbf{D}^*) < f(\mathbf{Y})$ holds for all sufficiently small values of ϵ. This is equivalent to proving that

$$\lim_{\epsilon\to 0} f\big(\mathbf{Y}+\epsilon\mathbf{D}^*\big) = \big[g_J(\mathbf{Y})\big]_U - \big[g_J(\mathbf{Y})\big]_V < 0.$$

The KKT necessary conditions for local optimality for (2) are the following:

$$\begin{cases} g_i(\mathbf{Y}) - \lambda_i \mathbf{e}_n - \boldsymbol{\mu}_i = \mathbf{0}, & \forall i \in \mathcal{O} \\ \mathbf{Y}^\top \mathbf{e}_K - \mathbf{e}_n = \mathbf{0} \\ \mathrm{Tr}\big(\mathbf{M}^\top \mathbf{Y}\big) = 0, \end{cases} \tag{12}$$

where $\mathbf{M} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_n) \in \mathbb{R}_+^{K\times n}$ and $\boldsymbol{\lambda} \in \mathbb{R}^n$ are the Lagrangian multipliers. We can express the Lagrange multipliers $\boldsymbol{\lambda}$ in terms of $\mathbf{Y}$ from the relation

$$\mathbf{y}_i^\top \big[g_i(\mathbf{Y}) - \lambda_i \mathbf{e}_n - \boldsymbol{\mu}_i\big] = 0,$$

which yields $\lambda_i = \mathbf{y}_i^\top g_i(\mathbf{Y})$ for all $i \in \mathcal{O}$. This can then be used to obtain an alternative characterization of the KKT conditions, where the Lagrange multipliers do not appear:

$$\begin{cases} \big[r_i(\mathbf{Y})\big]_k = 0, & \forall i \in \mathcal{O}, \forall k \in \sigma(\mathbf{y}_i), \\ \big[r_i(\mathbf{Y})\big]_k \geq 0, & \forall i \in \mathcal{O}, \forall k \notin \sigma(\mathbf{y}_i), \\ \mathbf{Y}^\top \mathbf{e}_K - \mathbf{e}_n = \mathbf{0}, \end{cases} \tag{13}$$

where

$$r_i(\mathbf{Y}) = g_i(\mathbf{Y}) - \lambda_i \mathbf{e}_K = g_i(\mathbf{Y}) - \mathbf{y}_i^\top g_i(\mathbf{Y}) \mathbf{e}_K.$$

The two characterizations (13) and (12) are equivalent. This can be verified by exploiting the non negativity of both matrices $\mathbf{M}$ and $\mathbf{Y}$, and the complementary slackness conditions. Additionally we have that $[r_j(\mathbf{Y})]_{U_j} \leq 0 \leq [r_j(\mathbf{Y})]_{V_j}$ for all $j \in \mathcal{O}$. In fact,

$$\big[g_j(\mathbf{Y})\big]_{U_j} \leq \mathbf{y}_j^\top g_j(\mathbf{Y}) = \sum_{k\in\sigma(\mathbf{y}_j)} y_{kj} \big[g_j(\mathbf{Y})\big]_k \leq \sum_{k\in\sigma(\mathbf{y}_j)} y_{kj} \big[g_j(\mathbf{Y})\big]_{V_j} = \big[g_j(\mathbf{Y})\big]_{V_j}$$

and by subtracting $\mathbf{y}_j^\top g_j(\mathbf{Y})$ we obtain the desired relation

$$\big[r_j(\mathbf{Y})\big]_{U_j} \leq 0 \leq \big[r_j(\mathbf{Y})\big]_{V_j}. \tag{14}$$

Now, by assuming $\mathbf{Y}$ to be feasible but not satisfying the KKT conditions, we derive from (13) that there exists $j \in \mathcal{O}$ such that at least one of the two cases hold: $[r_j(\mathbf{Y})]_u < 0$ for some $u \in \{1, \dots, K\}$, or $[r_j(\mathbf{Y})]_v > 0$ for some $v \in \sigma(\mathbf{y}_j)$. This, by definition of U_j, V_j and by (14), implies that $[r_j(\mathbf{Y})]_{U_j} < 0 \leq [r_j(\mathbf{Y})]_{V_j}$ or $[r_j(\mathbf{Y})]_{U_j} \leq 0 < [r_j(\mathbf{Y})]_{V_j}$. Hence, by definition of J,

$$\big[g_J(\mathbf{Y})\big]_U - \big[g_J(\mathbf{Y})\big]_V \leq \big[g_j(\mathbf{Y})\big]_{U_j} - \big[g_j(\mathbf{Y})\big]_{V_j} = \big[r_j(\mathbf{Y})\big]_{U_j} - \big[r_j(\mathbf{Y})\big]_{V_j} < 0$$

from which the result follows. □

Proposition 5 *The optimization problem in* (9) *is convex*, *provided that the Bregman divergence is double-convex.*

Proof The search direction $\mathbf{D}^*$, solution to (5), is everywhere null excepting two entries of the Jth column. This and the fact that the sum in (3) is taken over all pairs (i, j) such that $i \neq j$ implies that the second argument of every $B_\phi(\cdot\|\cdot)$ function is linear in ϵ. The Bregman divergence $B_\phi(\cdot\|\cdot)$ adopted is by assumption double-convex and in particular convex in its second argument and trivially the same holds for the function d_ϕ. Since convexity is preserved by the composition of convex functions with linear ones and by the sum of convex functions (Boyd and Vandenberghe 2004) it follows that the minimization problem in (2) is convex as well. □

References

Arora, R., Gupta, M., Kapila, A., & Fazel, M. (2011). Clustering by left-stochastic matrix factorization. In L. Getoor & T. Scheffer (Eds.), *ICML* (pp. 761–768). Omnipress.

Ayad, H., & Kamel, M. S. (2008). Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(1), 160–173.

Banerjee, A., Krumpelman, C., Basu, S., Mooney, R. J., & Ghosh, J. (2005a). Model-based overlapping clustering. In *Int. conf. on knowledge discovery and data mining* (pp. 532–537).

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

Bezdek, J. (1981). *Pattern recognition with fuzzy objective function algorithms*. Norwell: Kluwer Academic.

Bezdek, J., & Hathaway, R. (2002). VAT: a tool for visual assessment of (cluster) tendency. In *Proceedings of the 2002 international joint conference on neural networks 2002*, IJCNN'02 (Vol. 3, pp. 2225–2230).

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization* (1st ed.). Cambridge: Cambridge University Press.

Cui, Y., Fern, X. Z., & Dy, J. G. (2010). Learning multiple nonredundant clusterings. In *Transactions on Knowledge Discovery from Data (TKDD)* (Vol. 4, pp. 1–32).

Dhillon, I. S., Mallela, S., & Kumar, R. (2003). A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, *3*, 1265–1287.

Dimitriadou, E., Weingessel, A., & Hornik, K. (2002). A combination scheme for fuzzy clustering. In *AFSS'02* (pp. 332–338).

Färber, I., Günnemann, S., Kriegel, H., Kröger, P., Müller, E., Schubert, E., Seidl, T., & Zimek, A. (2010). On using class-labels in evaluation of clusterings. In *MultiClust: 1st international workshop on discovering, summarizing and using multiple clusterings*.

Fern, X. Z., & Brodley, C. E. (2004). Solving cluster ensemble problems by bipartite graph partitioning. In *Proc. ICML '04*.

Frank, A., & Asuncion, A. (2012). In *UCI machine learning repository*. http://archive.ics.uci.edu/ml.

Fred, A. (2001). Finding consistent clusters in data partitions. In J. Kittler & F. Roli (Eds.), *Multiple classifier systems* (Vol. 2096, pp. 309–318). Berlin: Springer.

Fred, A., & Jain, A. (2002). Data clustering using evidence accumulation. In *Proc. of the 16th int'l conference on pattern recognition* (pp. 276–280).

Fred, A., & Jain, A. (2005). Combining multiple clustering using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(6), 835–850.

Fred, A., & Jain, A. (2006). Learning pairwise similarity for data clustering. In *Proc. of the 18th int'l conference on pattern recognition (ICPR)*, Hong Kong (Vol. 1, pp. 925–928).

Ghosh, J., & Acharya, A. (2011). Cluster ensembles *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(4), 305–315.

Karypis, G., Aggarwal, R., Kumar, V., & Shekhar, S. (1997). Multilevel hypergraph partitioning: applications in VLSI domain. In *Proc. design automation conf.*

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(Suppl 1), 5228–5235.

Heller, K., & Ghahramani, Z. (2007). A nonparametric Bayesian approach to modeling overlapping clusters. In *Int. conf. AI and statistics*.

Jain, A. K., & Dubes, R. (1988). *Algorithms for clustering data*. New York: Prentice Hall.

Jardine, N., & Sibson, R. (1968). The construction of hierarchic and non-hierarchic classifications. *Computer Journal*, *11*, 177–184.

Kachurovskii, I. R. (1960). On monotone operators and convex functionals. *Uspehi Matematičeskih Nauk*, *15*(4), 213–215.

Karypis, G., & Kumar, V. (1998). Multilevel algorithms for multi-constraint graph partitioning. In *Proceedings of the 10th supercomputing conference*.

Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *NIPS* (pp. 556–562). Cambridge: MIT Press.

Lourenço, A., Fred, A., & Figueiredo, M. (2011). A generative dyadic aspect model for Evidence Accumulation Clustering. In *Proc. 1st int. conf. similarity-based pattern recognition, SIMBAD'11* (pp. 104–116). Berlin/Heidelberg: Springer.

Lourenço, A., Fred, A., & Jain, A. K. (2010). On the scalability of evidence accumulation clustering. In *Proc. 20th international conference on pattern recognition (ICPR)*, Istanbul, Turkey.

Mei, J. P., & Chen, L. (2010). Fuzzy clustering with weighted medoids for relational data. *Pattern Recognition*, *43*(5), 1964–1974.

Meila, M. (2003). Comparing clusterings by the variation of information. In Springer (Ed.), *Proc. of the sixteenth annual conf. of computational learning theory*, COLT.

Nepusz, T., Petróczi, A., Négyessy, L., & Bazsó, F. (2008). Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review A*, *77*, 016107.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: analysis and an algorithm. In *NIPS* (pp. 849–856). Cambridge: MIT Press.

Paatero, P., & Tapper, U. (1994). Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, *5*(2), 111–126.

Punera, K., & Ghosh, J. (2007). Soft consensus clustering. In *Advances in fuzzy clustering and its applications*. New York: Wiley.

Punera, K., & Ghosh, J. (2008). Consensus-based ensembles of soft clusterings. *Applied Artificial Intelligence*, *22*(7&8), 780–810.

Rota Bulò, S., Lourenço, A., Fred, A., & Pelillo, M. (2010). Pairwise probabilistic clustering using evidence accumulation. In *Proc. 2010 int. conf. on structural, syntactic, and statistical pattern recognition*, SSPR&SPR'10 (pp. 395–404).

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *22*(8), 888–905.

Steyvers, M., & Griffiths, T. (2007). Latent semantic analysis: a road to meaning. In *Probabilistic topic models*. Laurence Erlbaum.

Strehl, A., & Ghosh, J. (2003). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, *3*, 583–617.

Topchy, A., Jain, A., & Punch, W. (2003). Combining multiple weak clusterings. In *IEEE intl. conf on data mining*, Melbourne (pp. 331–338).

Topchy, A., Jain, A., & Punch, W. (2004). A mixture model of clustering ensembles. In *Proc. of the SIAM conf. on data mining*.

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.

Wang, H., Shan, H., & Banerjee, A. (2009). Bayesian cluster ensembles. In *9th SIAM int. conf. on data mining*.

Wang, H., Shan, H., & Banerjee, A. (2011). Bayesian cluster ensembles. *Statistical Analysis and Data Mining*, *4*(1), 54–70.

Wang, P., Domeniconi, C., & Laskey, K. B. (2010). Nonparametric Bayesian clustering ensembles. In *ECML PKDD'10* (pp. 435–450).