On Weighting Clustering

Richard Nock and Frank Nielsen

Abstract—Recent papers and patents in iterative unsupervised learning have emphasized a new trend in clustering. It basically consists of penalizing solutions via weights on the instance points, somehow making clustering move toward the hardest points to cluster. The motivations come principally from an analogy with powerful supervised classification methods known as boosting algorithms. However, interest in this analogy has so far been mainly borne out from experimental studies only. This paper is, to the best of our knowledge, the first attempt at its formalization. More precisely, we handle clustering as a constrained minimization of a Bregman divergence. Weight modifications rely on the local variations of the expected complete log-likelihoods. Theoretical results show benefits resembling those of boosting algorithms and bring modified (weighted) versions of clustering algorithms such as k-means, fuzzy c-means, Expectation Maximization (EM), and k-harmonic means. Experiments are provided for all these algorithms, with a readily available code. They display the advantages that subtle data reweighting may bring to clustering.

Index Terms—Clustering, Bregman divergences, k-means, fuzzy k-means, expectation maximization, harmonic means clustering.

1 Introduction

Recently, a new methodology in the design of supervised learning algorithms has allowed us to obtain dramatic improvements of classification performances: the constrained minimization of Bregman divergences [1]. A Bregman divergence is, informally speaking, the tail of the Taylor expansion of a differentiable convex function. A famous problem in computational learning theory was addressed and solved by this technique [2], [3], [4]: *boosting*, that is, the problem of combining the outputs of moderately accurate classifiers to get, with high probability, a highly accurate ensemble [5]. Online learning has also benefited from this framework, as well as relevant applications in portfolio prediction, text categorization, and calendar management [1].

On the other hand, unsupervised learning algorithms have so far remained remarkably cut off from this line of work. A notable exception is the work of [6], [7], whose objective is the use of Bregman divergences to extend current clustering algorithms such as k-means and Expectation Maximization (EM) to densities, members of the exponential family. However, this work does not consider the modification of clustering algorithms by leveraging data, which is the crux of boosting. This is all the more interesting as is it well-known that Bregman divergence minimization brings weighted iterative solutions for learning [1] and there has recently been a growing attention around weighted iterative clustering algorithms in unsupervised learning, such as, harmonic means clustering [8], [9]. Recent approaches have even emphasized the benefits of weighting the instances in clustering [8], [10] and make first attempts to explain the quality of the experimental results by boosting analogies [8], [11], [12]; unfortunately, the analogy has so far remained quite loose, supported mainly by experimental results and the notice that weighting functions tend to give greater weights to points less efficiently clustered, thereby "attracting" the cluster centers.

It is the aim of this paper to formulate clustering as an abstract problem of constrained Bregman divergence minimization. The solution yields original extensions of clustering algorithms, whose application is given for four members:

1. k-means [13],
2. fuzzy c-means [14] (called, in this paper, fuzzy k-means for notational convenience),
3. Gaussian EM [15], and
4. harmonic means clustering [9].

This solution has attractive boosting related theoretical features [4]. It completes the previous ad hoc analogies on the behavior of weighted clustering [8], [9], [11]. Furthermore, its implementation does not require a significant implementation effort as it boils down to plugging a local module for weight modification in the abstract clustering scheme of [8]. Also, weighting does not exhibit a significant complexity increase. In fact, compared to clustering algorithms such as EM and harmonic means clustering, there is no additional complexity penalty. We have implemented and tested our applications. Numerous experiments assess the influence of various parameters on clustering, such as the cluster types, the balance between clusters, and the initialization of clustering. Among other comments emerges the fact that weighting is a worthwhile companion for clustering and, in particular, a convenient alternative to previous clustering with soft membership (fuzzy k-means, EM, harmonic means clustering), which seems to work better as the problems contain highly imbalanced clusters, i.e., as they become harder.

Section 2 presents some preliminaries on clustering. Section 3 details the theoretical aspects of clustering with Bregman divergences. Section 4 presents and discusses some experiments with a readily available implementation of our algorithms. Section 5 and the Acknowledgments conclude the paper and detail the code availability.

- *R. Nock is with the Département Scientifique Inter-facultaire/GRIMAAG Lab., Université Antilles-Guyane, B.P. 7209, 97278 Schoelcher, Martinique, France. E-mail: rnock@martinique.univ-ag.fr.*
- *F. Nielsen is with Sony Computer Science Laboratories Inc., 3-14-13 Higashi Gotanda, Shinagawa-Ku, Tokyo 141-0022, Japan. E-mail: Frank.Nielsen@acm.org.*

Manuscript received 6 Sept. 2005; revised 22 Nov. 2005; accepted 29 Dec. 2005; published online 13 June 2006.
Recommended for acceptance by L. Kuncheva.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0482-0905.

2 Definitions and Preliminaries

Our notations closely follow [6]. Bold-faced variables, such as $\mathbf{p}, \mathbf{x}$, denote (column) vectors. Sets are represented by calligraphic upper-case alphabets such as $\mathcal{X}$. The elements of $\mathcal{X}$ are enumerated x_i, for $i = 1, 2, \ldots, |\mathcal{X}|$ ($\mathbf{x}_i$ if they are vectors), where $|.|$ denotes the cardinal. Blackboard faces denote subsets of $\mathbb{R}$, the set of real numbers. $\langle ., . \rangle$ defines the inner product for real valued vectors, i.e., the dot product, and $\| \mathbf{z} \| = \langle \mathbf{z}, \mathbf{z} \rangle^{\frac{1}{2}}$ for some real-valued vector $\mathbf{z}$. A distribution $\mathbf{w}$ over $\mathcal{X}$ is some unit mass discrete measure and the uniform distribution over $\mathcal{X}$, $\mathbf{u}$, is such that $u_i = 1/n$ ($\forall 1 \leq i \leq n$). In the context of clustering, this set, $\mathcal{X}$, is a point set of n elements in a d-dimensional metric space.

Clustering aims at recovering a structure in data. We are given an integer $k > 0$ and wish to recover k *component models* that best fit sampling $\mathcal{X}$. Prominent approaches, including k-means [13] and EM [15], can be cast in a probabilistic framework. Each model is a density, $\mathbf{p}(.; \theta_j)$, for $1 \leq j \leq k$, where θ_j denotes its parameters. We note, for short, $p(\mathbf{x}_i; \theta_j) = p_i^j$. For any distribution $\mathbf{w}$ over $\mathcal{X}$, define:

$$\ell(\mathbf{w}) = \sum_{i=1}^{n} w_i \ell_i, \tag{1}$$

$$\ell_i = \sum_{j=1}^{k} -m(j,i) \log(p_i^j \pi_j), \forall 1 \leq i \leq n. \tag{2}$$

Here, $\boldsymbol{\pi}$ is a distribution over the set of models, the *mixing proportions*: π_j is the proportion of $\mathcal{X}$ that comes from model j. $m(j,i)$ is called the posterior responsibility (or probability), or *membership function*, as it generally defines a distribution over the models for each $1 \leq i \leq n$. $m(j,i)$ defines the proportion of element $\mathbf{x}_i$ that belongs to cluster j. Equation (1) is formulated in a general way for future purposes, but, if $\mathbf{w}$ were uniform, $(\mathbf{w} = \mathbf{u})$, $-\ell_t(\mathbf{u})$ would equal the expected complete log-likelihood, which is approximately locally maximized by k-means and EM in their respective frameworks. The main difference between k-means and EM comes from a longstanding debate on the assignments of points to the clusters. k-means chooses *hard* membership assignment since each point belongs to exactly one cluster (for each i, $m(j,i)$ is 1 for a single j and 0 for all others). On the other side, EM chooses *soft* membership of each point to all clusters.

More recently, some authors have begun to question the transfer to clustering of a well-known supervised learning principle. *Boosting* [2] has shown from both the theoretical and experimental standpoints that improvements in the performances of iterative supervised learning algorithms can be obtained when one makes subtle reweighting of the (labeled) points in $\mathcal{X}$. It seems natural to try to transfer this property to clustering: Whenever the loss function is essentially decreasing as a function to a cluster center (such as for Gaussian priors or, more generally, members of the exponential family [6]), points with higher weights should attract the cluster centers [8]. The iterative nature of popular clustering algorithms [13], [15], [11] is certainly another motivation for this transfer as the adaptive nature of boosting algorithms comes, in part, from the fact that they are iterative. Weighting a clustering algorithm boils down to defining a distribution $\mathbf{w}$ over $\mathcal{X}$. So far, however, exploiting this analogy has been mainly the subject of empirical studies only. A first example is Topchy et al. [10], who tailor resampling to make clustering adaptive, by favoring points that have been less efficiently clustered so far. A second example is Zhang [9], who replaces the k-means loss by a soft, differentiable loss which introduces the distance to *all* centers instead of just the distance to the closest of them: the harmonic mean.

Algorithm 1. `General Iterative Clustering` $\mathcal{X}, k$
Input: point set $\mathcal{X}$, integer $k > 0$
Initialize the models $\mathbf{p}_0^j, \forall 1 \leq j \leq k$ (plus eventual related parameters)
for $t = 0, 1, \ldots$ **do**
[1.] *Compute/update* the memberships: $m_t(j,i), \forall 1 \leq i \leq n, \forall 1 \leq j \leq k$;
[2.] *Compute/update* the weights: $w_{t,i}, \forall 1 \leq i \leq n$;
[3.] *Update* the models: $\mathbf{p}_{t+1}^j, \forall 1 \leq j \leq k$ (plus eventual related parameters);

Algorithm 1, which is inspired by [8], gives a useful abstract view of clustering from which we may derive all algorithms we consider in this paper. Remark that the algorithm's parameters (memberships, weights, models, and related parameters) are indexed first by an iteration number (e.g., t). With the help of Algorithm 1, Table 1 gives a synthesis of the four categories of algorithms we consider in this paper, including the loss that each algorithm strives to minimize. Here are some additional comments: There are two popular initializations for the centers $\boldsymbol{\mu}_{0,.}$: Forgy and random [8]. Forgy initializes the cluster centers by picking at random k points over the n. In random initialization, centers are computed by first assigning each point to a random cluster and then computing the cluster centers from the sets obtained. In EM, $\boldsymbol{\pi}_0$ is initialized to the uniform distribution (it remains uniform in k-means). We have modeled k-means with Gaussians having covariance matrices proportional to identity. Under the constraint of hard membership, it is easy to check that $-\ell_t(\mathbf{u})$ ((3)) is indeed maximized for the membership choices in Table 1. Furthermore, the update of the centers in Step 3 of Algorithm 1 boils down to a conventional least square minimization problem equivalent to the *quantization error* minimization [13]. In our context of clustering and after Kearns et al. [16], we define the current *KMN loss* on some point $\mathbf{x}_i$ as $\min_j \| \mathbf{x}_i - \boldsymbol{\mu}_{t,j} \|^2$. Since clustering algorithms do not always minimize the same loss (see Table 1), the KMN loss is sometimes used as the preferred comparison measure [8]. Finally, in EM, Step 1 would be the E-step and Step 3 would be the M-step.

3 An Abstract Weighting Scheme for Clustering Algorithms

Definition 1. *We call the* advantage *over distribution* $\mathbf{w}_t$ *at iteration* t *the quantity* $\gamma_t \in \mathbb{R}$ *that satisfies* $\ell_{t+1}(\mathbf{w}_t) - \ell_t(\mathbf{w}_t) = -\gamma_t$, $\forall t \geq 0$. *We also define vector* $\mathbf{d}_t$ *such that* $d_{t,i} = \ell_{t+1,i} - \ell_{t,i}, \forall 1 \leq i \leq n$.

We thus have $\langle \mathbf{w}_t, \mathbf{d}_t \rangle = -\gamma_t$. EM and k-means are algorithms that work by a repetitive minimization of $\ell_t(\mathbf{u})$. They guarantee that, if $\mathbf{w}_t$ were uniform, then we would have $\gamma_t \geq 0$ so that the clustering would indeed get better with the iterations (hence, the name of γ_t).

TABLE 1
Synthesis of the Original Algorithms Considered in the Paper

Algorithm	k-means [13]	Fuzzy k-means [14]	(Gaussian) EM [15]	Harmonic means [9]
Model parameters	Gaussians $\mathcal{N}(\boldsymbol{\mu}_j, \sigma I)$	Centers $\boldsymbol{\mu}_j$	Gaussians $\mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)$	Centers $\boldsymbol{\mu}_j$
Loss:	$\ell_t(\mathbf{u})$ in eq. (3)			
	$\ell_{t,i}$: see eq. (4)	$\ell_{t,i}$: see eq. (5)	$\ell_{t,i}$: see eq. (4)	$\ell_{t,i}$: see eq. (6)
Membership	hard	soft	soft	soft
Step [1.]: $m_t(j,i) =$	1 for $\arg\min_{j'} \parallel \mathbf{x}_i - \boldsymbol{\mu}_{t,j'} \parallel^2$ 0 otherwise;	$1/\sum_{l=1}^k \lambda_{t,i,j,l}^{\left(\frac{2}{a-1}\right)}$; $\lambda_{t,i,j,l} = \frac{\parallel \mathbf{x}_i - \boldsymbol{\mu}_{t,j} \parallel}{\parallel \mathbf{x}_i - \boldsymbol{\mu}_{t,l} \parallel}$	$\lambda_{t,i,j}/\sum_{l=1}^k \lambda_{t,i,l}$; $\lambda_{t,i,j} = \pi_{t,j} p_{t,i}^j$	$\lambda_{t,i,j}/\sum_{l=1}^k \lambda_{t,i,l}$; $\lambda_{t,i,j} = \frac{1}{\parallel \mathbf{x}_i - \boldsymbol{\mu}_{t,j} \parallel^{a+2}}$
Step [2.]: $w_{t,i} =$	$1/n$			$\frac{\sum_{j=1}^k \lambda_{t,i,j}^{-2}}{z\left(\sum_{j=1}^k \lambda_{t,i,j}\right)^2}$ $\lambda_{t,i,j} = \frac{1}{\parallel \mathbf{x}_i - \boldsymbol{\mu}_{t,j} \parallel^a}$
Step [3.]: $\boldsymbol{\mu}_{t+1,j} =$	see (7)	see (8)	see (7)	
Step [3.] (misc.)			$\boldsymbol{\pi}_{t+1}$, $\Sigma_{t+1,j}$: see eqs (9) and (10)	
Misc.		$a \in [1, +\infty)$		$a \in [2, +\infty)$ z=normalization coeff.

$$\ell_t(\mathbf{w}) = \sum_{i=1}^n w_i \ell_{t,i} \ , \tag{3}$$

$$\ell_{t,i} = \sum_{j=1}^k -m_t(j,i) \log(p_{t,i}^j \pi_{t,j}) \ , \forall 1 \leq i \leq n \ , \tag{4}$$

$$\ell_{t,i} = \sum_{j=1}^k m_t^a(j,i) \parallel \mathbf{x}_i - \boldsymbol{\mu}_{t,j} \parallel^2 \ , \forall 1 \leq i \leq n \ , \tag{5}$$

$$\ell_{t,i} = k/\sum_{j=1}^k 1/ \parallel \mathbf{x}_i - \boldsymbol{\mu}_{t,j} \parallel^a \ , \forall 1 \leq i \leq n \ , \tag{6}$$

$$\boldsymbol{\mu}_{t+1,j} = \sum_{i=1}^n m_t(j,i) \mathbf{x}_i / \sum_{i=1}^n m_t(j,i) \ , \forall 1 \leq j \leq k \ , \tag{7}$$

$$\boldsymbol{\mu}_{t+1,j} = \sum_{i=1}^n m_t^a(j,i) \mathbf{x}_i / \sum_{i=1}^n m_t^a(j,i) \ , \forall 1 \leq j \leq k \ , \tag{8}$$

$$\pi_{t+1,j} = \frac{1}{n} \sum_{i=1}^n m_t(j,i) \ , \forall 1 \leq j \leq k \ , \tag{9}$$

$$\Sigma_{t+1,j} = \sum_{i=1}^n m_t(j,i) (\mathbf{x}_i - \boldsymbol{\mu}_{t+1,j})(\mathbf{x}_i - \boldsymbol{\mu}_{t+1,j})^\top / \sum_{i=1}^n m_t(j,i) \ , \forall 1 \leq j \leq k \ . \tag{10}$$

The problem is: What if we demand that the advantage be measured on other distributions? Is there something to gain over the initial uniform distribution $\mathbf{u}$? These questions may appear surprising at first glance because $\mathbf{u}$ is the natural distribution of data. Thereby, it is certainly the most direct way to minimize the loss $\ell_t(.)$. But, it appears not to be the only way to achieve this goal. Surprisingly, sometimes, it is also not the best.

The remainder of this section is now the investigation of a particular weighting scheme and its possible theoretical benefits on clustering. For this objective, we make two assumptions on Algorithm 1: First, $\mathbf{w}_0 = \mathbf{u}$. Second, in Step 3, we pick $\mathbf{p}_{t+1}^j (1 \leq j \leq k)$ such that:

$$\langle \mathbf{w}_t, \mathbf{d}_t \rangle = -\gamma_t. \tag{11}$$

This is equivalent to having: $\sum_{i:d_{t,i}<0} w_{t,i}|d_{t,i}| = \gamma_t + \sum_{i:d_{t,i}>0} w_{t,i}|d_{t,i}|$. By means of words, when $\gamma_t > 0$, the total amount of loss decrease *exceeds* the total amount of loss increase by quantity γ_t (with respect to $\mathbf{w}_t$) and the next clustering models are chosen so as to make the new clustering have at least a small gain on $\mathbf{w}_t$. In Boosting, there is a similar assumption of *biased weak* loss reduction, called the *weak learning assumption*, which turns out to be sufficient for significant *unbiased* global loss reductions too (i.e., over the initial distribution) [2], [4]. We now show that it is also within reach for clustering, provided the sole satisfaction of our *weak clustering assumption* ((11) with $\gamma_t > 0$). For this objective, we return to Step 2 of Algorithm 1. $\mathbf{w}_{t+1}$ is found by minimizing a convex functional under some particular constraints. This functional is a Bregman divergence: The *information divergence* [1], $\langle \mathbf{1}, \mathbf{i}_t \rangle$, with $\mathbf{i}_t$ the vector whose coordinate for some $\mathbf{x}_i \in \mathcal{X}$ is $i_{t,i} = w_{t+1,i} \ln(w_{t+1,i}/w_{t,i}) - w_{t+1,i} + w_{t,i}$. The problem we would like to solve for $\mathbf{w}_{t+1}$ is the following one:

$$\text{minimize} \quad \langle \mathbf{1}, \mathbf{i}_t \rangle, \tag{12}$$
$$\text{s.t.} \quad \langle \mathbf{1}, \mathbf{w}_{t+1} \rangle = 1, \tag{13}$$
$$\text{s.t.} \quad \langle \mathbf{w}_{t+1}, \mathbf{d}_t \rangle = 0. \tag{14}$$

With the fact that $\mathbf{w}_0 = \mathbf{u}$, constraint (13) is a sufficient condition to express the fact that $\mathbf{w}_{t+1}$ is a distribution (see below, (15), for an analytical expression). Constraint (14) states that $\mathbf{w}_{t+1}$ is decorrelated with respect to loss variations: $\ell_{t+1}(\mathbf{w}_{t+1}) = \ell_t(\mathbf{w}_{t+1})$. This has a consequence on the following step of the algorithm: Constraint (11) will force the models in $\mathbf{p}_{t+2}$ to be different from those in $\mathbf{p}_{t+1}$, thereby trying to learn something "new" out of the current clustering.

3.1 Dichotomic Approximations to (12)

The convexity of the information divergence makes that (12) is solved via the Lagrangian: $L(\mathcal{X}, b_t, c_t) = \langle \mathbf{1}, \mathbf{i}_t \rangle - b_t(1 - \langle \mathbf{1}, \mathbf{w}_{t+1} \rangle) - c_t(0 - \langle \mathbf{w}_{t+1}, \mathbf{d}_t \rangle)$ with b_t and c_t Lagrange multipliers for constraints (13) and (14), respectively. $\mathbf{w}_{t+1}$ is obtained after solving $\partial L(\mathcal{X}, b_t, c_t)/\partial w_{t+1,i} = \ln(w_{t+1,i}/w_{t,i}) - b_t - c_t d_{t,i} = 0$ ($\forall 1 \leq i \leq n$). Though it does not admit closed-form solutions, it can be simplified to yield:

$$w_{t+1,i} = \frac{w_{t,i} \exp(-c_t d_{t,i})}{\exp(b_t(c_t))}, \forall 1 \leq i \leq n. \tag{15}$$

In (15), $b_t(.)$ is called the cumulant function, whose expression is obtained with constraint (13): $b_t(c) = \ln \sum_{i=1}^n w_{t,i} \exp(-c d_{t,i})$ (with $c \in \mathbb{R}$). The term inside the "ln" is the normalization coefficient for $\mathbf{w}_{t+1}$: $Z_t(c) = \sum_{i=1}^n w_{t,i} \exp(-c d_{t,i})$. Finally, c_t is obtained using constraint (14), as the solution to:

$$\sum_{i=1}^n w_{t,i} d_{t,i} \exp(-c_t d_{t,i}) = 0. \tag{16}$$

This equation does not admit a closed-form solution in the general case. However, we need to compute at least an approximation with desirable properties for the weighted algorithm. The following lemma, whose proof is straightforward, states under which conditions (16) admits a unique solution.

Lemma 1. *Suppose* $\exists 1 \leq i \leq n : d_{t,i} > 0$ *and* $\exists 1 \leq i \leq n : d_{t,i} < 0$. *Then, (16) has a single solution.*

Lemma 1 does not state where c_t lies in $\mathbb{R}$. Without more information, even searching for approximations might represent a considerable complexity burden. Fortunately, we show that c_t lies on an interval of reduced measure. Define $\underline{d}_t = \min_{1 \leq i \leq n : d_{t,i} > 0} |d_{t,i}|$, $\overline{d}_t = \max_{1 \leq i \leq n} |d_{t,i}|$, $d_t^+ = \sum_{1 \leq i \leq n : d_{t,i} > 0} w_{t,i} |d_{t,i}|$, and $d_t^- = \sum_{1 \leq i \leq n : d_{t,i} < 0} w_{t,i} |d_{t,i}|$.

Lemma 2. *The solution to (16) satisfies*

$$|c_t| \in \left[\frac{1}{\overline{d}_t} \left| \ln \sqrt{d_t^- / d_t^+} \right|, \frac{1}{\underline{d}_t} \left| \ln \sqrt{d_t^- / d_t^+} \right| \right].$$

Proof. Define $g(c)$ as the sigma of (16) in which c_t is replaced by variable $c \in \mathbb{R}$. We consider three cases, depending on the value of γ_t. We have $g(0) = \langle \mathbf{w}_t, \mathbf{d}_t \rangle = -\gamma_t$ from constraint (11). When $\gamma_t = 0$, $g(0) = 0$ and, thus, $c_t = 0$. In this case, since $\langle \mathbf{w}_t, \mathbf{d}_t \rangle = d_t^+ - d_t^- = 0$, the left and right bounds in Lemma 2 are both zero and the lemma is satisfied. Suppose now that $\gamma_t > 0$. $g(0) < 0$ easily yields $c_t < 0$ and we have:

$$\sum_{1 \leq i \leq n : d_{t,i} > 0} w_{t,i} d_{t,i} \exp(-c_t d_{t,i}) \leq \exp(-c_t \overline{d}_t) d_t^+, \tag{17}$$

$$\sum_{1 \leq i \leq n : d_{t,i} < 0} w_{t,i} |d_{t,i}| \exp(-c_t d_{t,i}) \geq \exp(c_t \overline{d}_t) d_t^-. \tag{18}$$

Furthermore,

$$g(c_t) = \sum_{1 \leq i \leq n : d_{t,i} > 0} w_{t,i} d_{t,i} \exp(-c_t d_{t,i}) - \sum_{1 \leq i \leq n : d_{t,i} < 0} w_{t,i} |d_{t,i}| \exp(-c_t d_{t,i}) = 0,$$

which yields the identity of the two sigmas in (17) and (18) and, finally, yields $\exp(-c_t \overline{d}_t) d_t^+ \geq \exp(c_t \overline{d}_t) d_t^-$, from which we obtain $c_t \leq -(1/\overline{d}_t) \ln \sqrt{d_t^- / d_t^+}$. Because $d_t^- > d_t^+$, this upper bound is also strictly negative. Now, we also have the following: $\sum_{1 \leq i \leq n : d_{t,i} > 0} w_{t,i} d_{t,i} \exp(-c_t d_{t,i}) \geq \exp(-c_t \underline{d}_t) d_t^+$ and $\sum_{1 \leq i \leq n : d_{t,i} < 0} w_{t,i} |d_{t,i}| \exp(-c_t d_{t,i}) \leq \exp(c_t \underline{d}_t) d_t^-$, from which we get $\exp(c_t \underline{d}_t) d_t^- \geq \exp(-c_t \underline{d}_t) d_t^+$ and $c_t \geq -(1/\underline{d}_t) \ln \sqrt{d_t^- / d_t^+}$. The case for $\gamma_t < 0$ is obtained in the same way. □

Lemma 2 shows that c_t may be approximated through a simple dichotomic search, with reduced complexity. Fix

$$\underline{c}_t = \min \left\{ -(1/\overline{d}_t) \ln \sqrt{1 + (\gamma_t / d_t^+)}, -(1/\underline{d}_t) \ln \sqrt{1 + (\gamma_t / d_t^+)} \right\}$$

and

$$\overline{c}_t = \max \left\{ -(1/\overline{d}_t) \ln \sqrt{1 + (\gamma_t / d_t^+)}, -(1/\underline{d}_t) \ln \sqrt{1 + (\gamma_t / d_t^+)} \right\}.$$

Then, Lemma 2 also shows that:

$$c_t \in [\underline{c}_t, \overline{c}_t]. \tag{19}$$

Let $\hat{c}_t$ denote our approximation to c_t. Suppose we wish $|c_t - \hat{c}_t| / |c_t| \leq \varepsilon$. Then, the number of dichotomic steps to beat this relative error ε is only $\mathcal{O}(\ln(\overline{d}_t / \underline{d}_t) + \ln(1/\epsilon))$: For this to hold, observe that a sufficient condition is to have $((b-a)/(2^\tau a)) \leq \varepsilon$, with $[a, b]$ the interval in Lemma 2 and τ the number of dichotomic steps. Solving for τ yields the bound. Before going on further, it is worthwhile noticing that the weighting behavior respects the boosting analogy of [8] when the clustering gets better, i.e., when $c_t < 0$. In this case, points that "attract" centers on the next models have their weight decreasing. Thus, they somehow "leave space" for the next clustering rounds, for the points whose clustering degrades. Our weighting scheme displays, however, an original pattern previously not reported: Whenever the clustering gets worse, greater weights are given to the points *more efficiently* clustered. This tends to penalize the current clustering, making it somehow "attracted" toward the previous, better solutions. Finally, Lemma 2 also shows that the sign of c_t and $\hat{c}_t$ is the same. Lemma 2 does not say, however, why we should carry out such computations for weight modification.

3.2 Closed-Form Approximations to (12) and Conditions of Improvement of Clustering

This section is split into two. We first show that there are different ways to compute $\hat{c}_t$ that bring a convenient result on the normalization coefficient, namely, $Z_t(\hat{c}_t)$ is smaller than one by a *significant* amount. In a second part, we show that the normalization coefficient is the key to the improvement of clustering, i.e., the decrease of the losses used in Table 1.

3.2.1 An Upper Bound on $Z_t(\hat{c}_t)$

First, since $\partial Z_t(c)/\partial c = -g(c)$, we see that $Z_t(c)$ is strictly decreasing on $[-\infty, c_t]$ and strictly increasing on $[c_t, +\infty]$. We also have $Z_t(0) = \langle \mathbf{1}, \mathbf{w}_t \rangle = 1$. Thus, without any more derivation, if we pick for $\hat{c}_t$ the bound of (19) which is the closest to zero, then we would already have $Z_t(.) < 1$. The amount by which $Z_t(.)$ is < 1 still remains to be given. For this objective, we make use of the following simple Lemma.

Lemma 3.

$$\exp(-xz) \leq ((1+x)/2)\exp(-z) + ((1-x)/2)\exp(z), \quad \forall x \in [-1,1], \forall z \in \mathbb{R}.$$

Proof. Left is a convex curve crossing points $(-1, \exp(z))$ and $(1, \exp(-z))$ and right is the equation of the line crossing these two points. □

The following lemma states the upper bound on the normalization coefficient.

Lemma 4. *Suppose* $|\gamma_t| < \overline{d}_t$. *Then,* $Z_t(c) \leq \exp\left(-\gamma_t^2/(2\overline{d}_t^2)\right)$ *whenever* $\gamma_t \geq 0$ *and* $c \in [c_t, \overline{c}_t]$ *and whenever* $\gamma_t \leq 0$ *and* $c \in [\underline{c}_t, c_t]$.

Proof. $\forall 1 \leq i \leq n$, if we fix $x = d_{t,i}/\overline{d}_t$ and $z = c\overline{d}_t$ ($\forall c \in \mathbb{R}$), then we obtain from Lemma 3:

$$\exp(-cd_{t,i}) \leq ((\overline{d}_t + d_{t,i})/(2\overline{d}_t))\exp(-c\overline{d}_t) + ((\overline{d}_t - d_{t,i})/(2\overline{d}_t))\exp(c\overline{d}_t).$$

Plugging this into $Z_t(c)$ and using (11), we obtain:

$$\begin{aligned} Z_t(c) &= \sum_{i=1}^n w_{t,i}\exp(-cd_{t,i}) \\ &\leq ((\overline{d}_t + \langle \mathbf{w}_t, \mathbf{d}_t \rangle)/(2\overline{d}_t))\exp(-c\overline{d}_t) \\ &\quad + ((\overline{d}_t - \langle \mathbf{w}_t, \mathbf{d}_t \rangle)/(2\overline{d}_t))\exp(c\overline{d}_t) \\ &= ((\overline{d}_t - \gamma_t)/(2\overline{d}_t))\exp(-c\overline{d}_t) \\ &\quad + ((\overline{d}_t + \gamma_t)/(2\overline{d}_t))\exp(c\overline{d}_t). \end{aligned}$$

Fix $h(c)$ as this last function for short. We have $Z_t(c) \leq h(c), \forall c \in \mathbb{R}$ and, therefore, $Z_t(c_t) \leq \inf_{c \in \mathbb{R}} h(c)$. We have $|\gamma_t| \leq \overline{d}_t$, and provided this inequality is strict, $h(c)$ is strictly convex with $+\infty$ limits when $|c| \to +\infty$. Its minimizing c is easily obtained as:

$$\tilde{c}_t = -\frac{1}{\overline{d}_t}\ln\sqrt{1 + \frac{2\gamma_t}{\overline{d}_t - \gamma_t}}, \tag{20}$$

which yields

$$h(\tilde{c}_t) = \sqrt{1 - (\gamma_t/\overline{d}_t)^2} \leq \exp(-\gamma_t^2/(2\overline{d}_t^2)).$$

To finish the proof of the lemma, it is sufficient to prove that $\tilde{c}_t$ is in between 0 and the interval $[\underline{c}_t, \overline{c}_t]$. First, notice that the lemma holds trivially when $\gamma_t = 0$ (since $\underline{c}_t = \overline{c}_t = \tilde{c}_t = 0$). Suppose that $\gamma_t > 0$. In this case, we have

$$\overline{c}_t = -(1/\overline{d}_t)\ln\sqrt{1 + (\gamma_t/d_t^+)} < 0;$$

furthermore, since $\gamma_t < \overline{d}_t$ and $\gamma_t = d_t^- - d_t^+$, we easily obtain that $\overline{c}_t < \tilde{c}_t$ and, so, the statement of the lemma holds. Suppose now that $\gamma_t < 0$. In this case,

$$\underline{c}_t = (1/\overline{d}_t)\ln\sqrt{1 - (\gamma_t/d_t^-)} > 0$$

and we can also rewrite $\tilde{c}_t$ as

$$\tilde{c}_t = (1/\overline{d}_t)\ln\sqrt{1 - 2\gamma_t/(\overline{d}_t + \gamma_t)} > 0.$$

Using again $\gamma_t < \overline{d}_t$ and $\gamma_t = d_t^- - d_t^+$, we easily obtain $\tilde{c}_t < \underline{c}_t$ and, so, the statement of the lemma holds. □

Lemma 4 says that, if we want a normalization coefficient $Z_t(\hat{c}_t)$ explicitly smaller than one up to the bound stated, then we have three strategies, in decreasing order of the upperbound on $Z_t(\hat{c}_t)$:

- The simplest: We fix $\hat{c}_t = \tilde{c}_t$. This brings $Z_t(\hat{c}_t)$ the closest to the bounds.
- A more efficient: We use for $\hat{c}_t$ the bound of the interval (19) which is the closest to zero.
- The most efficient: We run the dichotomic search inside (19).

3.2.2 Improvement of the Clustering

From now on, we write, for short, $\hat{Z}_t = Z_t(\hat{c}_t)$. We suppose we have run Algorithm 1 for $T+1$ clustering rounds. The next lemma is immediate from the proof of Lemma 4, but it is useful to emphasize the results to follow.

Lemma 5. $\gamma_t \hat{c}_t \leq 0$, $\forall t \geq 0$, *with inequality iff* $\gamma_t = \hat{c}_t = 0$.

We now let $A_0 = \hat{c}_0$ and $A_T = \hat{c}_0\ell_0(\mathbf{u}) + \sum_{t=1}^T (\hat{c}_t - \hat{c}_{t-1})\ell_t(\mathbf{u})$, $\forall T > 0$. We also let $B_T = \sum_{t=0}^T \ln(1/\hat{Z}_t)$, $\forall T \geq 0$, and $D_T = (\max_i w_{T,i} - \min_i w_{T,i})^2/(4\max_i w_{T,i}\min_i w_{T,i})$, $\forall T > 0$. Remark that provided each $\hat{c}_t$ is picked according to one of the three methods above, each summand in B_T is positive, so that $B_T > 0$. Furthermore, D_T is positive; it quantifies a discrepancy between the maximal and the minimal weight in $\mathbf{w}_T$. We need the following result, a reverse of the AGH inequality due to [17]:

Lemma 6. *Consider* n *reals* $0 < x_1 \leq x_2 \leq \ldots \leq x_n$ *and a distribution* $\mathbf{w}$ *over these reals. Then, we have*

$$\sum_i w_i x_i \leq \prod_i x_i^{w_i}\exp((x_n - x_1)^2/(4x_1x_n)).$$

The following lemma bounds loss $\ell_{T+1}(\mathbf{u})$ as a function of the three parameters A_T, B_T and D_T.

Lemma 7. $A_T + B_T \leq \hat{c}_T\ell_{T+1}(\mathbf{u}) \leq A_T + B_T + D_{T+1}$, $\forall T \geq 0$.

Proof. Consider some iteration $T > 0$. We unravel the update rule in (15) and obtain $w_{T+1,i} = (1/(n\prod_{t=0}^T \hat{Z}_t))\exp(-\sum_{t=0}^T \hat{c}_t d_{t,i})$, $\forall 1 \leq i \leq n$. Summing over all $1 \leq i \leq n$

and rearranging, we obtain $(1/n)\sum_{i=1}^{n}\exp(-\sum_{t=0}^{T}\hat{c}_t d_{t,i}) = \prod_{t=0}^{T}\hat{Z}_t$. Now, we lower bound the exponential average using Jensen's inequality, and upper bound the same average via Lemma 6. We can simplify the term appearing in the exponential penalty of (6) and we finally get the following inequalities:

$$\exp\left(-\sum_{t=0}^{T}\hat{c}_t\langle\mathbf{u},\mathbf{d}_t\rangle\right) \le \frac{1}{n}\sum_{i=1}^{n}\exp\left(-\sum_{t=0}^{T}\hat{c}_t d_{t,i}\right) \le \exp\left(-\sum_{t=0}^{T}\hat{c}_t\langle\mathbf{u},\mathbf{d}_t\rangle\right)\times\exp(D_{T+1}).$$

There remains to use the unraveled update rule to replace the central exponential and solve using the relationship $\langle\mathbf{u},\mathbf{d}_t\rangle = \ell_{t+1}(\mathbf{u}) - \ell_t(\mathbf{u})$. This yields the statement of Lemma 7. □

Lemma 7 brings immediate upper bounds on $\ell_{T+1}(\mathbf{u})$, depending on the sign of $\hat{c}_T$. For example, when $\hat{c}_T \le 0$ (i.e., $\gamma_T \ge 0$), we obtain:

$$\ell_{T+1}(\mathbf{u}) \le (A_T/\hat{c}_T) + (B_T/\hat{c}_T). \tag{21}$$

Note that $\hat{c}_0/\hat{c}_T + \sum_{t=1}^{T}(\hat{c}_t - \hat{c}_{t-1})/\hat{c}_T = 1$ so that $A_T/\hat{c}_T$ describes a weighted sum of losses, each loss computed over the initial distribution $\mathbf{u}$ (and not on the skewed distributions $\mathbf{w}_t$). We get that $\ell_{T+1}(\mathbf{u})$ is no more than this weighted sum plus an additional term. Provided the leveraging coefficients $\hat{c}_t$ are picked according to whichever of the three methods above is used, this additional term, $B_T/\hat{c}_T$, is < 0, also tends to decrease with T. Under some particular circumstances, we can obtain an upper bound on $\ell_{T+1}(\mathbf{u})$ much more explicit, as shown by the following Lemma 8:

Lemma 8. *Let $T > 0$. Suppose that $\hat{c}_t \le \hat{c}_{t-1} \le 0, \forall 0 < t \le T$. Then, $(A_T + B_T)/\hat{c}_T \le \ell_0(\mathbf{u}) + \sum_{t=0}^{T}(1/\hat{c}_t)\ln(1/\hat{Z}_t)$.*

Proof. The proof is obtained by induction on T. Case $T = 0$ is immediate as both sides coincide. To prove case $T = 1$, remark that we can write:

$$\frac{A_1 + B_1}{\hat{c}_1} = \ell_0(\mathbf{u}) + \left(\frac{\hat{c}_1 - \hat{c}_0}{\hat{c}_1}\right)\left[\ell_1(\mathbf{u}) - \ell_0 - \frac{1}{\hat{c}_0}\ln\frac{1}{\hat{Z}_0}\right] + \frac{1}{\hat{c}_0}\ln\frac{1}{\hat{Z}_0} + \frac{1}{\hat{c}_1}\ln\frac{1}{\hat{Z}_1}. \tag{22}$$

Lemma 7 yields $\ell_1(\mathbf{u}) \le (A_0 + B_0)/\hat{c}_0$, which, after the induction hypothesis, brings $\ell_1(\mathbf{u}) \le \ell_0(\mathbf{u}) - (1/\hat{c}_0)\ln(1/\hat{Z}_0)$. Thus, the expression inside brackets is ≤ 0. Furthermore, factor $(\hat{c}_1 - \hat{c}_0)/\hat{c}_0 \ge 0$, which yields that the right-hand side of (22) is $\le \ell_0 + (1/\hat{c}_0)\ln(1/\hat{Z}_0) + (1/\hat{c}_1)\ln(1/\hat{Z}_1)$, as claimed. Case $T > 1$ is obtained by writing again:

$$\frac{A_T + B_T}{\hat{c}_T} = \left(1 - \frac{\hat{c}_{T-1}}{\hat{c}_T}\right)\ell_T(\mathbf{u}) + \frac{1}{\hat{c}_T}\ln\frac{1}{\hat{Z}_T} + \frac{\hat{c}_{T-1}}{\hat{c}_T}\left[\frac{A_{T-1} + B_{T-1}}{\hat{c}_{T-1}}\right]. \tag{23}$$

Since $\hat{c}_{T-1}/\hat{c}_T > 0$, we may use the induction's hypothesis inside the brackets of (23) and get:

$$\frac{A_T + B_T}{\hat{c}_T} \le \left(1 - \frac{\hat{c}_{T-1}}{\hat{c}_T}\right)\ell_T(\mathbf{u}) + \frac{\hat{c}_{T-1}}{\hat{c}_T}\ell_0(\mathbf{u}) + \frac{\hat{c}_{T-1}}{\hat{c}_T}\sum_{t=0}^{T-1}\frac{1}{\hat{c}_t}\ln\frac{1}{\hat{Z}_t} + \frac{1}{\hat{c}_T}\ln\frac{1}{\hat{Z}_T}. \tag{24}$$

Equation (24) can be written as follows:

$$\frac{A_T + B_T}{\hat{c}_T} \le \ell_0(\mathbf{u}) + \left(\frac{\hat{c}_T - \hat{c}_{T-1}}{\hat{c}_T}\right)\left[\ell_T(\mathbf{u}) - \ell_0 - \sum_{t=0}^{T-1}\frac{1}{\hat{c}_t}\ln\frac{1}{\hat{Z}_t}\right] + \sum_{t=0}^{T}\frac{1}{\hat{c}_t}\ln\frac{1}{\hat{Z}_t}. \tag{25}$$

To finish up, Lemma 7 yields $\ell_T \le (A_{T-1} + B_{T-1})/\hat{c}_{T-1}$, which, after the induction hypothesis, brings $\ell_T(\mathbf{u}) \le \ell_0(\mathbf{u}) - \sum_{t=0}^{T-1}(1/\hat{c}_t)\ln(1/\hat{Z}_t)$. Thus, the term inside brackets in (25) is ≤ 0 and, since its factor is ≥ 0, we obtain the proof of the induction's general case and that of the lemma as well. □

What is interesting about Lemma 8 is that, under its assumptions (that also imply $\gamma_t \ge 0$), $A_T/\hat{c}_T$ becomes a (weighted) *average* of losses as all factors of the ℓ_ts are ≥ 0 in A_T. Under the conditions of Lemma 8, if we pick $\hat{c}_t$ as in (20), then we easily obtain the following bound (with the fact that $\ln(1+x) \le x$):

$$\ell_{T+1}(\mathbf{u}) \le \ell_0(\mathbf{u}) - \sum_{t=0}^{T}\frac{\gamma_t^2}{\overline{d}_t\ln\left(\frac{\overline{d}_t+\gamma_t}{\overline{d}_t-\gamma_t}\right)} \le \ell_0(\mathbf{u}) - \sum_{t=0}^{T}\frac{\gamma_t}{2}\left(1 - \frac{\gamma_t}{\overline{d}_t}\right). \tag{26}$$

Since $\gamma_t \ge 0$ (Lemma 5) and $\overline{d}_t \ge \gamma_t$, we see that even when Algorithm 1 relies on a distribution repeatedly skewed with our weighting scheme, it can also bring a decrease of the loss on the *initial* distribution too.

So far, we have seen the ways we can tackle a direct, *global* minimization of $\ell_\cdot(\mathbf{u})$, which is basically distribution dependent as it takes into account $\mathbf{w}_t$ ($t \ge 0$) and $\mathbf{u}$. There is a second way to cope with the reduction which does not take into account the distribution since it focuses *locally* on the reduction of the pointwise losses, as defined in (4), (5), and (6). Its principle is that a pointwise decrease of this loss on a sufficiently large number of points may also bring a reduction of $\ell_\cdot(\mathbf{u})$ as well, regardless of distributions $\mathbf{w}_t$. The following lemma is the key to show that it is indeed possible.

Lemma 9. *$\forall T > 0$, suppose that $\gamma_T > 0$. Then,*

$$\frac{1}{n}\left|\left\{\mathbf{x}_i \in \mathcal{X} : \exp -\ell_{T+1,i} \le \left(\exp -\ell_{0,i}\right)^{\frac{\hat{c}_0}{\hat{c}_T}}\prod_{t=1}^{T}\left(\exp -\ell_{t,i}\right)^{\frac{\hat{c}_t - \hat{c}_{t-1}}{\hat{c}_T}}\right\}\right| \le \prod_{t=0}^{T}\hat{Z}_t. \tag{27}$$

Proof. Define $[\![\pi]\!]$, the function which returns the truth value $\in \{0,1\}$ of predicate π; clearly, $[\![q \le 0]\!] \le \exp(-q), \forall q \in \mathbb{R}$.

TABLE 2
Synthesis of the Weighted Algorithms Considered in the Paper

Algorithm	k-means [13]	Fuzzy k-means [14]	(Gaussian) EM [15]	Harmonic means [9]
Step [1.]:	see Table I			
Step [2.]: $w_{t,i} =$	solve (12)			
Step [3.]: $\boldsymbol{\mu}_{t+1,j} =$	see (29)	see (30)	see (29)	see (31)
Step [3.] (misc.)			$\boldsymbol{\pi}_{t+1}$, $\Sigma_{t+1,j}$: see eqs (9) and (33)	

$$\boldsymbol{\mu}_{t+1,j} = \sum_{i=1}^{n} w_{t,i} m_t(j,i) \mathbf{x}_i / \sum_{i=1}^{n} w_{t,i} m_t(j,i) \ , \forall 1 \leq j \leq k \ , \tag{29}$$

$$\boldsymbol{\mu}_{t+1,j} = \sum_{i=1}^{n} w_{t,i} m_t^a(j,i) \mathbf{x}_i / \sum_{i=1}^{n} w_{t,i} m_t^a(j,i) \ , \forall 1 \leq j \leq k \ , \tag{30}$$

$$\boldsymbol{\mu}_{t+1,j} = \sum_{i=1}^{n} w_{t,i} g_{t,j,i} \mathbf{x}_i / \sum_{i=1}^{n} w_{t,i} g_{t,j,i} \ , \forall 1 \leq j \leq k \ , \tag{31}$$

$$g_{t,j,i} = \| \mathbf{x}_i - \boldsymbol{\mu}_{t,j} \|^{-a-2} / \left(\sum_{l=1}^{k} \| \mathbf{x}_i - \boldsymbol{\mu}_{t,l} \|^{-a} \right)^2 \ , \forall 1 \leq j \leq k, \forall 1 \leq i \leq n \ , \tag{32}$$

$$\Sigma_{t+1,j} = \sum_{i=1}^{n} w_{t,i} m_t(j,i) (\mathbf{x}_i - \boldsymbol{\mu}_{t+1,j})(\mathbf{x}_i - \boldsymbol{\mu}_{t+1,j})^\top / \sum_{i=1}^{n} w_{t,i} m_t(j,i) \ , \forall 1 \leq j \leq k \tag{33}$$

Thus, we have $[\![\sum_{t=0}^{T} \hat{c}_t d_{t,i} \leq 0]\!] \leq \exp(-\sum_{t=0}^{T} \hat{c}_t d_{t,i}), \forall 1 \leq i \leq n$. Since $\gamma_T > 0$, we have $\hat{c}_T < 0$ (Lemma 5) and the predicate can be reformulated as $-\ell_{T+1,i} \leq \sum_{t=1}^{T} (\frac{\hat{c}_t - \hat{c}_{t-1}}{\hat{c}_T})(-\ell_{t,i}) + \frac{\hat{c}_0}{\hat{c}_T}(-\ell_{0,i})$. Taking the exponential, the expectation of the predicate value and using $(1/n) \sum_{i=1}^{n} \exp(-\sum_{t=0}^{T} \hat{c}_t d_{t,i}) = \prod_{t=0}^{T} \hat{Z}_t$ (proof of Lemma 7), we obtain the statement of the lemma. □

Note that $\exp -\ell_{t,i}$ is the local likelihood of point $\mathbf{x}_i$ at iteration t. The right-hand side of the inequality in the cardinal is a geometric combination of the preceeding likelihoods (the sum of exponents is 1). Lemma 9 may be read as follows: The proportion of "bad" points for which the last $((T+1)\text{th})$ likelihood is no more than this geometric combination of the preceeding, is vanishing at least as fast as the product of the normalization coefficients. Consider the setting of Lemma 8. In this case, since each exponent is also nonnegative, the upper bound for $p_{T+1,i}$ is exactly a weighted geometric *average* of the preceeding ones and it resembles a likelihood as well. If we pick $\hat{c}_T$ following one of the three methods, the upper bound on $\hat{Z}_t$ of Lemma 4 yields that the set of these bad points is vanishing *exponentially fast*. Furthermore, since the geometric average is highly concave, points that are not bad may enjoy a fast local increase of their likelihood. Outside the setting of Lemma 8, the situation may be even more dramatic as multiplicands with negative exponents are equivalent to multiplications by potentially very large numbers. From the global standpoint, these phenomena may help to bring a convenient decrease of $\ell_.(\mathbf{u})$. Furthermore, if we use densities for $\mathbf{p}^j$ $(1 \leq j \leq k)$ like the members of the exponential family that are parameterized by centers (e.g., Gaussians), we may also expect a very fast spreading of these centers. To finish, Lemma 9 is much simplified when all $\hat{c}_t$ are identical and picked according to whichever of the three available methods. In this case, we obtain indeed:

$$\frac{|\{\mathbf{x}_i \in \mathcal{X} : \exp -\ell_{T+1,i} \leq \exp -\ell_{0,i}\}|}{n} \leq \prod_{t=0}^{T} \hat{Z}_t \leq \exp\left(-\sum_{t=0}^{T} \frac{\gamma_t^2}{2\overline{d}_t^2} \right). \tag{28}$$

To conclude this section, it tends to show that there are at least two basic strategies for weighted clustering, one rather local and one rather global:

- We pick $\hat{c}_t$ according to whichever of the three available methods. We are guaranteed that each $\hat{Z}_t < 1$ by a sufficient amount, so that Lemma 9 brings a fast local improvement of likelihoods. Lemma 7 also brings a direct decrease of $\ell_{T+1}(\mathbf{u})$, unless too many of the $\hat{c}_t$ are positive, in which case the upper bound on $\ell_{T+1}(\mathbf{u})$ becomes more difficult to read.
- We force a sequence of $\hat{c}_t$ following, e.g., Lemma 8. This boils down to using one of the three methods above and accept a new value for $\hat{c}_t$ if it is not higher than the last one. In this case, the bound for Lemma 8 is easy to read and, unless the skewed $\hat{Z}_t$ are too large, we may expect a convenient decrease of $\ell_{T+1}(\mathbf{u})$. On the other hand, Lemma 9 is this time not as easy to read since some $\hat{Z}_t$ may be > 1.

3.3 Finishing Up: Step 3 of Algorithm 1 and Synthesis

So far, we have seen the way we solve Step 2 of Algorithm 1. While Step 1 does not change, Step 3 is crucial since we have to find $\mathbf{p}_{t+1}$ out of $\mathbf{p}_t$ so as to ensure, whenever possible, a positive advantage γ_t in constraint (11).

For algorithms that can be cast into an approximate maximization of some expected complete log-likelihood (k-means, EM), for algorithms whose parameter update fits into conventional least square minimization (Fuzzy k-means), there is no great change in Step 3. Table 2 gives these updates

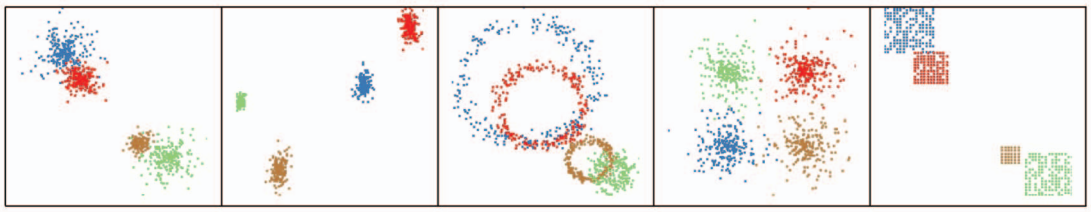

Fig. 1. The five types of data sets, with $d = 2$ and $K = 4$ theoretical clusters (from the left to the right: Gaussians, Gaussians with less overlaps, ring Gaussians, BIRCH, and cubic; see text for details).

without proofs. The case of harmonic means clustering is a little bit more involved to find the new centers out of the minimization of the harmonic mean (6). Table 1 displays the updates we have obtained by following [9].

4 Experiments

We report experiments comparing our weighted versions of clustering algorithms to the original algorithms. Notice that original algorithms may also be weighted (such as for k-harmonic means), but we keep the term "original" for these algorithms in order to avoid confusion with our modified weighted versions. There are various data sets used for our experimental comparisons, but, in order to make fair comparisons, the weighted and original versions we run are initialized with the *same* parameters (empirical centers, covariance matrices, etc.). Thus, any difference in the results stems from the differences in the weighting strategies. We have compared the original and weighted versions of all four algorithms of Table 1. We have fixed $a = 2$ for fuzzy k-means and harmonic means clustering. We have tested both Forgy and random initialization for the clusters centers and considered synthetic data sets with numerous possible characteristics, as detailed below.

4.1 Data Sets

The data sets differ according to the following parameters:

4.1.1 Types of Theoretical Clusters

We have implemented five different types of data set. In the first, we generate multidimensional spherical Gaussians, i.e., with covariance matrix $\approx \sigma^2 I$, with σ picked at random for each cluster. In the second, we do the same, but for clusters with less overlap (i.e., larger distance between cluster centers and smaller σs). In the third, we generate multidimensional ring Gaussians: A point for the jth cluster is generated by $\boldsymbol{\mu}_j + (r_j + \mathcal{N}(0, \sigma_j))\mathbf{u}$, where r_j is the ring radius of cluster j and $\mathbf{u}$ is a unit norm vector picked uniformly at random. The fourth case is considered only when $d = 2$; it was previously used to compare various clustering algorithms [8]. This data set, called BIRCH, consists of a set of 2D clusters whose centers are located on a $\lceil\sqrt{K}\rceil \times \lceil\sqrt{K}\rceil$ grid. Here, K denotes the number of theoretical clusters. The distance between two adjacent cluster means on the grid is $4\sqrt{2}$, with cluster radius of $\sqrt{2}$ (i.e., the variance in each direction is 1). Finally, in the fifth case, we generate multidimensional cubic uniform clusters with various degrees of overlap. Fig. 1 presents examples for each of the five types of data sets.

4.1.2 Theoretical Balance of Clusters

The experimental data sets are generated from theoretical clusters specified by models, but also by mixing proportions ($\pi_j, 1 \leq j \leq n$, with $\sum_{j=1}^{k} \pi_j = 1$, see Section 2). According to the *balance* between theoretical clusters, some of them may have very few points with respect to others and this may influence the result of clustering. In order to test the influence of the balance, we have tested three ways of fixing $\boldsymbol{\pi}$. In the first, $\boldsymbol{\pi} = \mathbf{u}$, i.e., the cluster proportions are theoretically uniform. In the second, $\boldsymbol{\pi}$ is a random distribution vector. In the third, the clusters have exponentially decreasing mixing proportions, i.e., $\pi_{j+1} = \pi_j/2$ (and the two last clusters have the same mixing proportion, to ensure that $\boldsymbol{\pi}$ yields unit mass). Fig. 2 presents examples of the three types of mixing proportions.

4.2 Weighted Clustering: An Example

To catch a glimpse into the way weighted clustering is weighting points and influencing clustering, Fig. 3 displays an example of run for weighted k-means on an "8"-like shape described by two ring Gaussians. t is the iteration number and the configuration shown is that immediately preceeding the run for t in Algorithm 1. Thus, the configuration for $t = 0$ is the initialization (first) configuration. The colors of points for $t > 0$ indicate their membership in a cluster. This explains why $t = 0$ does not have different colors for clusters as the membership has not yet been computed for the configuration displayed. Recall also that modifying weights can only occur when at least two configurations have been computed, hence the weights that remain constant for $t = 0$ and $t = 1$ in Fig. 3. Fig. 3 shows that points far from the initial centers tend to attract them at the beginning of the iterations; hence, their weights rapidly become smaller. On the other hand, the points at the intersection of rings, near the initial centers, see these center spreading around the rings and, thus, have their weights increasing. Everything isas if they were trying to "keep" some centers in their vicinity.

4.3 Overall Results

The experimental setting is as follows: We have crossed all initialization and data set parameters, for each possible value of $k \in \{2, 4, 8, 16, 32\}$ experimental clusters, $K \in \{2, 4, 8, 16, 32\}$ theoretical clusters, $d \in \{2, 5, 25, 50\}$ dimensions, and

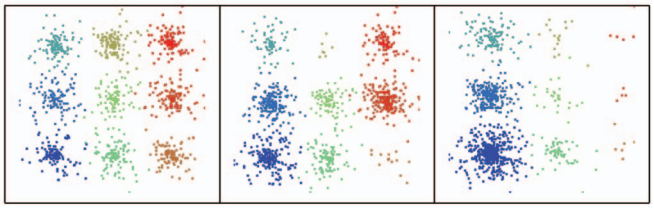

Fig. 2. The three types of mixing proportions, illustrated with $d = 2$ and $K = 9$ theoretical clusters on a BIRCH data set (from the left to the right: uniform, random, and exponentially decreasing; see text for details).

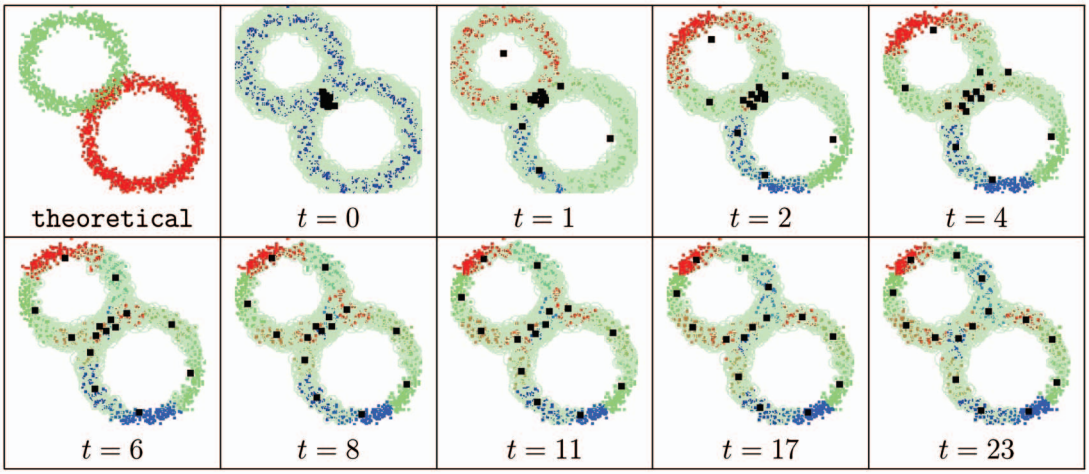

Fig. 3. Running weighted k-means on a set of $K = 2$ theoretical ring Gaussian clusters (upper left), with $k = 16$ experimental clusters and random initialization. Light-blue circles indicate the weights of the points (smaller radius means smaller weight). Big black dots are the empirical clusters centers. See text for details.

$n \in \{1,000, 2,000, 3,000, 6,000, 9,000, 12,000\}$ points. Basically, this represents more than 15,000 runs for each algorithm, weighted or not. There is an exception for EM, for which additional computational complexities (such as for matrix computations) made it convenient to reduce the set of sizes for $\mathcal{X}$ to $n \in \{500, 1,000, 1,500, 2,000\}$. Furthermore, even when we have implemented arbitrary Gaussian EM, the results presented here use diagonal covariance matrices, i.e., whose principal axes are parallel to the canonical axes. This reduces the numerical instabilities when computing inverses and yields more reliable comparisons. Finally, to still reduce the risk of numerical instabilities, we have chosen (for weighted EM only) to approximate $\ell_{t,i}$ in (4) by $\ell_{t,i} = \sum_{j=1}^{k} -(1/k) \log(p_{t,i}^{j})$, which boils down to locally approximating the memberships by the uniform distribution and has the benefit of canceling the mixing proportions in $d_{t,i}$ (Definition 1). Each algorithm was run for $T = 20$ clustering rounds and the last configuration was kept; while, in most cases, this was sufficient for the original algorithm to converge, this also gives an indication on the rapidity of the loss decrease. The weighted algorithm is run with the simplest computation for $\hat{c}_t$: $\hat{c}_t = \tilde{c}_t$ (Section 3.2).

The comparison of the algorithm's performances is based on three parameters:

- Following [8], $\ell_T(\mathbf{u})$ is the k-means KMN loss (see the k-means column in Table 1).
- miss_T is the proportion of theoretical clusters that are missed, i.e., that do not have an empirical cluster in their Voronoi cell. Such a criterion helps to estimate the rapidity in spreading the empirical centers [8].
- prop_T is the proportion of points whose final KMN loss is not smaller than their initial KMN loss. This is like comparing the final and initial likelihoods in Lemma 9 and in (28) and yields insights into the way the algorithms "share" the centers, i.e., spread the centers so as to make the greatest number of points benefit from this spreading.

Each of these parameters can be used as a loss measure to compare the algorithms. Due to the very large number of runs, the results have been synthesized to fit in the paper as follows: For each dimension and each K, we compute the proportion of wins and losses for the weighted algorithm. α estimates the proportion of random win/ties/losses sequences that present a ratio win/lose more favorable than that of the weighted algorithm against the original. For example, if α is small enough (say, 1 percent), we can reject the hypothesis that the advantage of weighted against original is to random: The weighted algorithm is significantly better. For computational and numerical reasons, we have chosen to compute an estimation of α instead of its true value. This estimation is computed via a standard concentration inequality ([18, p. 123]), with which we compute the minimum number of random sequences yielding an estimator $\hat{\alpha}$ no different from α by more than β, with probability $\geq 1 - \delta$. We have fixed $\beta = 10^{-4}$ and $\delta = 1\%$.

Clearly, our weighting scheme may be fit to many clustering algorithms: We have experienced this fact for four of them. We do not intend to demonstrate in this paper that there exists a clustering algorithm whose weighting scheme outperforms all clustering algorithms. Such a "free lunch" result is far beyond its scope. Rather, it is well known that clustering algorithms are very often highly sensitive to the fluctuations of some parameters (such as the initialization). In this context, any method that can be plugged into virtually many such algorithms, without significant computation load, without significant complexity load, while guaranteeing relevant theoretical results on the output's loss, is clearly of interest to try to improve, from the experimental standpoint, the algorithm's results. This is why we compare each original algorithm to its weighted version and not to the other (un)weighted clustering algorithms. Provided an improvement can be observed (and the experiments display this), our objective is much more to observe under which conditions does the best improvement occur.

Tables 3 and 4 present the results obtained for the four algorithms. Fig. 4 displays examples of configurations for which the weighted algorithm beats the original, for the four types of algorithms.

The following patterns emerge from the comparison of weighted versus original algorithms. First, weighted tends to perform better ($\ell_t(\mathbf{u})$) as the dimension increases. This is true for the three classes of clustering algorithms except for fuzzy

TABLE 3
Synthesis of the Results of Two Algorithms (Unweighted versus Weighted)

	K	$d=2$			$d=5$			$d=25$			$d=50$		
		Wins%	Lose%	α%	Wins%	Lose%	α%	Wins%	Lose%	α%	Wins%	Lose%	α%
$\ell_T(\mathbf{u})$	2	8.38	72.20	≈ 100	14.07	50.86	≈ 100	19.88	69.88	≈ 100	19.67	63.08	≈ 100
	4	6.64	79.60	≈ 100	8.15	72.84	≈ 100	24.07	73.33	≈ 100	17.53	76.55	≈ 100
	8	6.46	91.31	≈ 100	6.17	87.53	≈ 100	15.68	83.58	≈ 100	17.27	79.48	≈ 100
	16	7.48	91.62	≈ 100	6.67	91.60	≈ 100	18.64	81.36	≈ 100	15.32	87.77	≈ 100
	32	7.78	92.02	≈ 100	8.39	91.11	≈ 100	23.95	75.81	≈ 100	21.75	73.25	≈ 100
miss_T	2	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A
	4	0.51	0.40	0.21	0.37	0	≈ 0	0	0	N/A	0	0	N/A
	8	0.81	0.91	93.51	0.61	0.74	98.27	0.81	1.35	≈ 100	1.30	0.90	≈ 0
	16	3.43	2.12	≈ 0	0.49	1.35	≈ 100	3.11	2.43	0.35	1.56	1.17	0.05
	32	2.52	3.03	94.44	0.12	1.06	≈ 100	2.84	3.65	99.72	1.69	1.30	0.16
prop_T	2	67.07	11.62	≈ 0	51.98	10.25	≈ 0	64.67	11.40	≈ 0	47.95	10.00	≈ 0
	4	74.24	10.30	≈ 0	58.89	4.87	≈ 0	65.81	13.92	≈ 0	42.27	12.89	≈ 0
	8	71.92	12.63	≈ 0	60.00	10.49	≈ 0	69.46	10.41	≈ 0	58.05	7.66	≈ 0
	16	72.93	14.24	≈ 0	63.21	10.62	≈ 0	72.84	9.32	≈ 0	61.82	9.87	≈ 0
	32	74.95	14.24	≈ 0	77.41	7.47	≈ 0	78.65	10.14	≈ 0	68.44	7.79	≈ 0
$\ell_T(\mathbf{u})$	2	27.33	11.67	≈ 0	38.33	15.42	≈ 0	33.75	24.58	1.17	47.50	26.67	≈ 0
	4	26.00	11.66	≈ 0	26.25	21.25	8.17	22.08	34.58	99.42	30.83	30.83	50.00
	8	26.67	15.00	≈ 0	23.33	26.67	78.40	20.83	21.67	62.26	20.00	20.83	55.84
	16	23.33	12.67	≈ 0	16.25	36.67	≈ 100	10.83	18.75	≈ 100	15.83	21.67	90.47
	32	21.67	10.00	≈ 0	19.58	36.25	≈ 100	10.42	22.92	≈ 100	17.50	22.50	88.52
miss_T	2	0	0	N/A	0	0	N/A	0	0.83	≈ 100	0	6.67	≈ 100
	4	0.33	0	≈ 0	0	0	N/A	1.25	19.58	≈ 100	1.67	12.50	≈ 100
	8	0	0	N/A	0.42	0.83	≈ 100	4.17	7.92	≈ 100	2.50	7.50	≈ 100
	16	0	0.33	≈ 100	2.50	5.41	≈ 100	0	4.58	≈ 100	0.83	7.50	≈ 100
	32	0	0	94.44	7.92	10.00	≈ 100	0	5.83	≈ 100	0	4.17	≈ 100
prop_T	2	6.33	4.33	0.19	11.67	7.50	0.19	12.50	18.75	99.22	11.67	10.00	20.82
	4	5.67	6.33	78.98	7.08	5.42	5.44	39.58	9.58	0	33.33	4.17	≈ 0
	8	5.00	6.00	89.18	9.17	6.25	0.19	21.25	18.75	20.82	22.50	8.33	≈ 0
	16	5.67	5.33	31.52	25.42	10.00	≈ 0	14.17	15.83	74.32	8.33	16.67	≈ 100
	32	8.00	6.67	11.28	40.00	10.83	≈ 0	13.75	13.33	43.77	11.67	10.83	38.52

Top table: k-means; bottom table: fuzzy k-means. See text for details.

TABLE 4
Synthesis of the Results of Two Algorithms (Unweighted versus Weighted)

	K	$d=2$			$d=5$			$d=25$			$d=50$		
		Wins%	Lose%	α%	Wins%	Lose%	α%	Wins%	Lose%	α%	Wins%	Lose%	α%
$\ell_T(\mathbf{u})$	2	16.36	83.64	≈ 100	34.20	65.80	≈ 100	28.52	71.11	≈ 100	39.41	57.62	99.36
	4	13.94	85.76	≈ 100	25.65	74.35	≈ 100	27.51	72.49	≈ 100	39.23	69.38	99.74
	8	10.00	90.00	≈ 100	21.64	78.36	≈ 100	29.74	70.26	≈ 100	34.35	64.89	99.99
	16	12.73	86.97	≈ 100	28.36	71.64	≈ 100	34.07	65.93	99.99	35.55	64.45	99.99
	32	11.82	87.27	≈ 100	31.11	68.89	≈ 100	47.21	52.79	77.26	49.23	50.77	57.55
miss_T	2	0	0	N/A	0	0	N/A	1.85	8.89	≈ 100	4.83	16.73	≈ 100
	4	0.60	12.42	≈ 100	0	1.49	≈ 100	11.52	15.99	98.49	11.54	23.08	99.99
	8	10.91	33.03	≈ 100	4.85	12.69	≈ 100	13.38	20.45	99.74	12.21	16.41	97.35
	16	12.73	31.82	≈ 100	7.46	10.82	99.29	11.11	21.11	99.99	10.16	19.53	99.99
	32	6.97	37.27	≈ 100	11.90	16.36	98.30	11.52	21.93	99.99	9.61	13.46	99.99
prop_T	2	30.61	44.24	99.65	17.47	12.27	0.98	21.11	14.07	0.36	18.96	13.01	0.65
	4	33.94	42.73	95.59	21.87	19.14	22.43	11.52	23.05	≈ 100	14.62	11.92	9.14
	8	30.30	45.76	99.86	14.55	22.01	99.67	15.61	11.15	1.32	14.89	12.60	13.53
	16	26.36	42.12	99.97	14.93	17.91	88.52	13.70	7.04	≈ 0	8.59	8.98	60.68
	32	26.06	35.76	98.98	12.64	11.90	34.10	10.91	10.11	48.55	9.40	9.23	45.47
$\ell_T(\mathbf{u})$	2	23.64	75.76	≈ 100	26.17	72.84	≈ 100	44.57	39.01	6.08	58.77	38.15	≈ 0
	4	26.26	73.74	≈ 100	30.67	68.93	≈ 100	54.81	44.69	0.91	49.51	50.37	57.60
	8	31.21	68.69	≈ 100	50.25	49.75	45.66	50.53	43.60	4.98	43.20	49.07	92.25
	16	34.19	65.81	≈ 100	52.84	47.16	9.42	48.52	38.27	0.30	44.50	32.75	≈ 0
	32	39.89	60.11	≈ 100	44.67	55.33	99.14	48.80	27.07	≈ 0	38.58	28.25	≈ 0
miss_T	2	0	0	N/A	0	0	N/A	0.74	3.70	≈ 100	0	0	N/A
	4	2.02	1.31	≈ 0	0.13	1.20	≈ 100	2.35	0.49	≈ 0	2.72	5.68	≈ 100
	8	5.86	4.85	0.80	6.05	4.44	0.02	7.20	6.93	33.49	6.27	5.47	6.37
	16	6.88	9.89	≈ 100	8.03	6.91	4.23	12.22	11.36	19.73	14.64	8.55	≈ 0
	32	9.89	10.75	0.85	6.14	9.73	≈ 100	15.07	7.60	≈ 0	16.35	8.41	≈ 0
prop_T	2	65.05	10.71	≈ 0	60.74	8.89	≈ 0	32.96	26.30	0.44	17.90	21.48	98.26
	4	67.47	16.57	≈ 0	50.67	22.27	≈ 0	48.02	37.28	0.16	38.52	27.78	≈ 0
	8	75.45	18.18	≈ 0	55.68	29.01	≈ 0	55.87	25.20	≈ 0	42.80	28.67	≈ 0
	16	76.24	16.99	≈ 0	59.26	32.84	≈ 0	43.46	32.10	0.03	48.41	19.42	≈ 0
	32	76.56	19.68	≈ 0	52.94	41.33	0.31	44.80	29.20	≈ 0	49.53	14.76	≈ 0

Top table: EM; bottom table: harmonic means clustering. See text for details.

k-means, whose weighting scheme already significantly beats the original on low dimensions. On the contrary, there does not seem to be any difference as significant when the number of theoretical clusters increases. Second, the advantage for weighted against original algorithms seems to be more dominant for soft membership functions. This is clear from the results of k-means (hard membership) and fuzzy k-means (soft membership). Finally, there is a clear advantage on prop_T

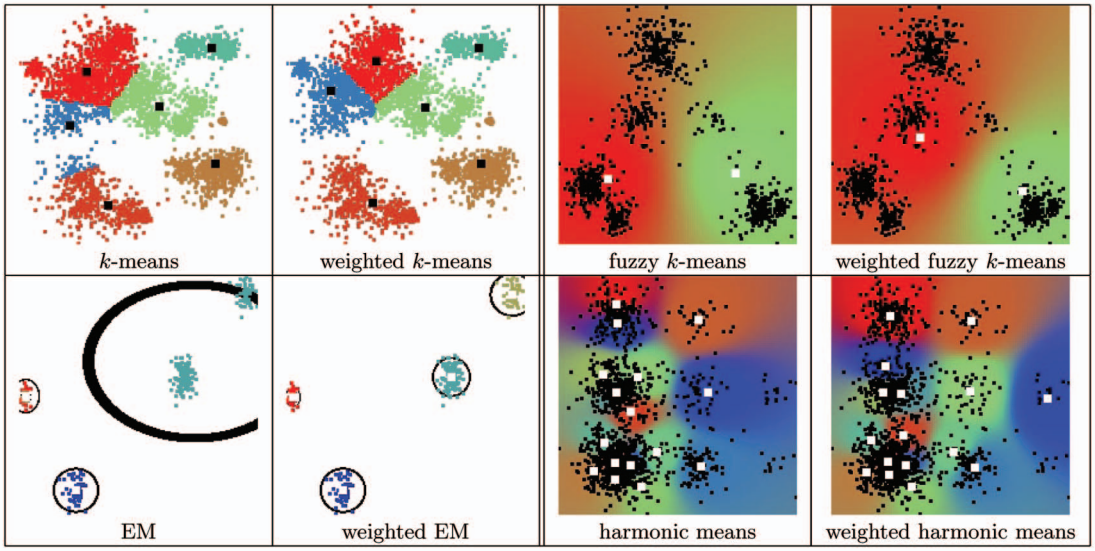

Fig. 4. Examples of configurations that are better for the weighted algorithms. Upper-left: The loss for weighted k-means is smaller by more than 10 percent than that of the original (clusters are Gaussians, mixing proportions are uniform, and initialization is random). Upper-right: The loss for fuzzy k-means is smaller by more than 13 percent than that of the original (clusters are Gaussians, mixing proportions are uniform, and initialization is random). Lower-left: The loss for weighted EM is smaller by more than 10 percent of that of the original (clusters are Gaussians, mixing proportions are exponentially decreasing and initialization is random). Lower-right: The loss for weighted harmonic means is smaller by more than 5 percent of that of the original and the fraction of theoretical centers missed is also smaller (configurations are BIRCH, mixing proportions are exponentially decreasing and initialization is random). In the right tables, white dots are the empirical clusters centers and colors show the soft membership function.

TABLE 5
Synthesis of the Percentages of Wins for Weighted Algorithms against Their Original Version as a Function of the Type of Cluster (for $\ell_t(\mathbf{u})$, See Text for Details)

type	value	KM			FKM			EM			HM			
		%	r	s	%	r	s	%	r	s	%	r	s	avg. rank
Cluster:	Gaussians	13.03	2	**2**	25.05	2	**2**	27.24	3	**2**	41.92	2	**2**	2.25
	Ring Gaussians	16.48	1	**1**	17.81	5	**4**	23.13	4	**3**	41.24	3	**2**	3.25
	BIRCH ($d = 2$)	11.56	4	**3**	21.67	4	**3**	13.33	5	**4**	33.22	5	**3**	4.50
	Gauss. less ov.	7.63	5	**4**	28.76	1	**1**	33.70	1	**1**	40.64	4	**2**	2.75
	Cubic	12.51	3	**2**	23.62	3	**2**	32.30	2	**1**	45.14	1	**1**	2.25
Balance:	Uniform	12.91	2	**1**	18.47	3	**3**	26.86	3	**1**	38.51	3	**2**	2.75
	Random	12.54	3	**1**	20.00	2	**2**	28.30	1	**1**	39.32	2	**2**	2.50
	Expo. decreas.	12.94	1	**1**	32.53	1	**1**	27.70	2	**1**	46.52	1	**1**	1.25
Init.:	Random	11.73	2	**2**	38.80	1	**2**	27.07	2	**1**	39.39	2	**2**	1.75
	Forgy	14.11	1	**1**	8.53	2	**1**	28.30	1	**1**	44.13	1	**1**	1.25

KM = k-means, FKM = fuzzy k-means, EM = (Gaussian) Expectation Maximization, and HM = harmonic clustering. Here, r is the rank and s is the statistical rank (see text for details).

for the weighted algorithms on k harmonic means clustering and k-means, but not for Gaussian EM and fuzzy k-means (and this phenomenon is not as evident for miss_T). Overall, this may display an improved facility for spreading the centers for the two first algorithms. To conclude these overall results, while there are sometimes only sparse improvements among runs, such as for k-means, the fact that the weighted schemes require only reduced implementation/computational efforts makes these modifications worth trying to "escape" the locality of search for the original algorithms. Some dramatic improvement can also be obtained for fuzzy k-means using our weighting scheme.

4.4 Drilling Down into Controlled Parameters

Since the homoscedasticity assumption is not satisfied for our experimental results, instead of carrying out variance analyses, we have drilled down the controlled parameters using either the same technique as in the preceeding section or multiple statistical proportion comparisons with confidence intervals built using the same concentration inequality as above ([18, p. 123]). For the confidence intervals, we have used $\delta = 1\%$.

As a first set of experiments, we have merged the dimension-dependent results, and crossed them as a function of the three basic controlled parameters: type of clusters, balance between theoretical clusters, and initialization type for the clustering algorithms. Table 5 presents the results for the weighted versions of clustering algorithms and $\ell_t(\mathbf{u})$. This is basically aimed at displaying for which types of parameters the weighted versions perform the best. In the table, the statistical rank (s) is computed as follows: First, we compute we confidence interval for each win proportion. Then, we put win proportions in decreasing order. Starting from the highest, we add each following proportion that fits into its confidence interval into the current cluster. As soon as the current proportion does not fit in, we start a new cluster with the same procedure until all values are processed. This procedure was chosen to avoid multiple overlapping clusters that would have followed from standard statistical analyses and that would have impaired the readability of the results.

TABLE 6
Synthesis of the Results of Two Algorithms for an Alternative Choice of $\hat{c}_t$ ($\hat{c}_{t+1} \leq \hat{c}_t$)

	K	$d=2$			$d=5$			$d=25$			$d=50$		
		Wins%	Lose%	α%	Wins%	Lose%	α%	Wins%	Lose%	α%	Wins%	Lose%	α%
$\ell_T(\mathbf{u})$	2	4.64	75.76	≈ 100	7.65	57.28	≈ 100	14.69	75.06	≈ 100	12.59	66.48	≈ 100
	4	5.35	80.71	≈ 100	5.06	75.93	≈ 100	19.38	78.02	≈ 100	17.59	76.30	≈ 100
	8	4.55	93.23	≈ 100	5.31	88.15	≈ 100	11.73	87.65	≈ 100	11.85	84.07	≈ 100
	16	5.76	93.33	≈ 100	6.67	91.85	≈ 100	11.36	88.64	≈ 100	11.48	87.59	≈ 100
	32	5.66	94.14	≈ 100	6.05	93.46	≈ 100	12.72	87.16	≈ 100	9.82	90.19	≈ 100
miss_T	2	0	0	N/A	0	0	N/A	0	0	N/A	0	0	N/A
	4	0.71	0.50	≈ 0	0.37	0	≈ 0	0	0	N/A	0	0.19	≈ 100
	8	0.81	1.62	≈ 100	0.74	0.86	96.37	1.11	1.24	88.85	1.85	0.93	≈ 0
	16	4.34	2.42	≈ 0	0.74	2.10	≈ 100	3.46	3.83	88.16	2.96	1.67	≈ 0
	32	2.22	3.23	≈ 100	0.49	2.47	≈ 100	2.93	3.46	96.33	1.67	1.30	0.85
prop_T	2	66.77	12.22	≈ 0	52.84	9.14	≈ 0	66.54	10.74	≈ 0	46.30	9.63	≈ 0
	4	74.24	11.41	≈ 0	60.00	9.63	≈ 0	65.43	12.96	≈ 0	40.56	13.89	≈ 0
	8	73.54	11.41	≈ 0	58.77	11.60	≈ 0	70.62	9.75	≈ 0	55.56	9.44	≈ 0
	16	70.81	16.87	≈ 0	64.32	11.98	≈ 0	75.43	8.39	≈ 0	61.11	6.30	≈ 0
	32	75.76	14.85	≈ 0	78.52	8.02	≈ 0	80.25	9.75	≈ 0	68.52	6.11	≈ 0
$\ell_T(\mathbf{u})$	2	21.88	78.12	≈ 100	26.02	73.98	≈ 100	45.52	53.36	85.55	42.08	57.92	98.10
	4	14.33	85.37	≈ 100	24.44	75.56	≈ 100	34.46	65.54	99.99	38.89	61.11	99.81
	8	13.68	86.32	≈ 100	22.22	77.78	≈ 100	36.80	63.20	99.98	38.19	59.45	99.78
	16	18.90	81.79	≈ 100	26.02	73.98	≈ 100	40.38	59.62	99.46	41.80	56.98	97.47
	32	33.03	66.67	≈ 100	26.30	73.70	≈ 100	40.23	59.77	99.52	43.90	56.10	93.95
miss_T	2	0.63	2.19	≈ 100	0	0	N/A	2.99	5.60	99.99	5.41	6.56	89.81
	4	3.05	20.43	≈ 100	1.48	3.70	≈ 100	17.60	10.86	0.09	19.05	14.29	3.29
	8	11.85	30.70	≈ 100	7.78	8.52	73.07	15.61	11.90	3.48	19.29	16.93	19.71
	16	17.68	28.05	99.95	8.92	14.13	99.87	13.21	15.47	85.19	18.03	27.87	99.68
	32	30.58	25.69	10.05	13.70	18.89	98.39	16.17	19.55	89.58	16.67	23.17	98.10
prop_T	2	32.19	52.50	99.97	19.33	17.10	20.53	8.58	14.18	99.94	18.15	15.06	11.18
	4	36.28	54.57	99.85	23.33	24.44	62.59	15.36	23.60	99.75	16.67	13.49	8.67
	8	37.39	48.02	96.71	22.96	20.00	17.79	13.38	11.15	11.32	12.20	12.60	57.76
	16	35.67	48.78	98.91	21.56	21.56	48.40	12.08	9.81	8.43	21.72	10.66	≈ 0
	32	40.38	40.98	54.09	15.19	15.56	57.32	13.53	10.53	4.77	16.67	7.31	≈ 0

Top table: k-means; bottom table: EM. See text, Table 3, and Table 4 for conventions and details.

From Table 5, we can say that cubic and Gaussian clusters are those for which our weighting scheme performs the best, followed by Gaussians with less overlaps and ring Gaussians. BIRCH are those for which weighting performs the worst. These last types of clusters have the reduced degree of overlap between clusters in common. Somehow, we can thus say that weighting does not significantly reduce the tendency of iterative clustering algorithms to be sensitive to the overlapping degree between theoretical clusters [8], even when a close look at the results of the original versions tends to display the fact that this tendency is reduced for weighting.

Furthermore, from the balance type of theoretical cluster, we can say that there is a clear tendency for weighting to perform better as the theoretical clusters are highly imbalanced. This are good news as data sets with such highly imbalanced clusters are frequently hard data sets. In Fig. 4, the weighted version of harmonic clustering clearly displays this ability to retrieve clusters among minorities. Fig. 4 is almost as clear for weighted EM.

Finally, there is also a clear tendency for weighting to perform better for Forgy initialization. This is not really surprising, as it is well-known that the original algorithms tend to perform better for random initialization [8], thus leaving room space for winning for the weighting algorithms on Forgy initialization. Comparatively, the results we have observed tend to display that the dependence of the performances for our weighting scheme is reduced with respect to the initialization type.

We have also drilled down these results to explore the influence of the dimension. First, for harmonic means clustering, the rankings observed for the types of clusters tend to become more dramatic as d increases. This is exactly the same observation that follows when changing the balance or the type of initialization. In that last case, for example, wins for weighting represent approximatively 60 percent for $d = 50$. On the other hand, there seems to be an inverse pattern for Gaussian EM: The wins tend to be more uniform as d increases for the type of cluster. This uniformization is also visible for the balance and initialization types. The case of k-means is more interesting. As d increases, the type of clusters which clearly makes weighting perform much worse is Gaussians with fewer overlaps. While weighted k-means wins on roughly 10 percent for them, it wins on more than 20 percent of the runs for each other type of cluster. Comparatively, there are no differences among balance and initialization types.

4.5 Alternative Choices for $\hat{c}_t$

In this section, we test the second alternative to compute the coefficients $\hat{c}_t$ outlined at the end of Section 3.2. Here, we compute a nonincreasing sequence of $\hat{c}_t$ following the same choice for new coefficients as in the preceeding section. Table 6 presents the results obtained. For EM, we have not upper bounded the sequence, which thus starts with $\hat{c}_0$ being computed as usual. From the tables, it seems that taking into account the sequences does not really bring a decrease in the loss functions with respect to Table 3 and Table 4. However, there seems to be a slight positive effect of the decreasing sequence for EM as the results for $d = 25$ are better than those of Table 4.

5 Conclusion

Recent papers in unsupervised learning have put a great emphasis on trying to bring to clustering the recent breakthrough of a supervised learning technique, boosting, that has allowed us to obtain dramatic improvements in performances. In the context of unsupervised learning, this represents the ability to make subtle reweighting of the points of a data set, with the hope of getting better final solutions and getting them faster than without reweighting. In fact, some of the essential reasons for this motivation are purely conceptual but quite appealing as it indeed seems natural that points less efficiently clustered so far may "attract" the clusters on the next rounds and, thus, receive greater weights [8], [9], [11], [12].

The main contribution of this paper is to adopt an insight from classification to improve the performance of unsupervised learning algorithms by making more precise this analogy to boosting algorithms. We have proposed a generic iterative clustering scheme that, coupled with some particular reweighting scheme, may indeed bring improvements over "classical" clustering from the theoretical standpoint. This iterative clustering scheme can be specialized to bring weighted variants of k-means, fuzzy k-means, Expectation Maximization, and harmonic means clustering [14], [8], [11], among others. The experimental results clearly display differences in the benefits of the weighting scheme depending on the original clustering algorithms. While the improvements seem to be much more significant for soft membership clustering (such as for fuzzy k-means), we think that our weighting scheme is worth trying for hard membership clustering algorithms as well because it may yield better solutions, at little implementation/computational expenses. In the near future, we plan to test our weighting scheme on more complex soft membership clustering algorithms, including variational Bayesian extensions of EM [19], [20].

Acknowledgments

The authors would like to thank the reviewers for insightful comments that helped to improve this manuscript. R. Nock would like to warmly thank Sony Computer Science Laboratories, Inc., Tokyo, for a visiting grant during which part of this work was done. The binaries used for this paper (data set generation, clustering algorithms, postprocessing files) are available from the authors.

References

[1] C. Gentile and M. Warmuth, "Proving Relative Loss Bounds for On-Line Learning Algorithms Using Bregman Divergences," *Proc. Tutorials 13th Int'l Conf. Computational Learning Theory,* 2000.

[2] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences,* vol. 55, pp. 119-139, 1997.

[3] J. Kivinen and M. Warmuth, "Boosting as Entropy Projection," *Proc. 12th Ann. Conf. Computational Learning Theory,* pp. 134-144, 1999.

[4] R.E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-Rated Predictions," *Proc. 11th Int'l Conf. Computational Learning Theory,* pp. 80-91, 1998.

[5] M.J. Kearns, "Thoughts on Hypothesis Boosting," ML class project, 1988.

[6] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman Divergences," *Proc. Fourth SIAM Int'l Conf. Data Mining,* pp. 234-245, 2004.

[7] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman Divergences," *J. Machine Learning Research,* vol. 6, pp. 1705-1749, 2005.

[8] G. Hammerly and C. Elkan, "Alternatives to the k-Means Algorithm that Find Better Clusterings," *Proc. 11th ACM Int'l Conf. Information and Knowledge Management,* pp. 600-607, 2002.

[9] B. Zhang, "Generalized k-Harmonic Means," Technical Report TR-HPL-2000-137, Hewlett Packard Labs, 2000.

[10] A. Topchy, B. Minaei-Bidgoli, A.-K. Jain, and W.-F. Punch, "Adaptive Clustering Ensembles," *Proc. 17th Int'l Conf. Pattern Recognition,* pp. 272-275, 2004.

[11] B. Zhang, M. Hsu, and U. Dayal, "k-Harmonic Means—A Spatial Clustering Algorithm with Boosting," *Temporal, Spatial, and Spatio-Temporal Data Mining,* pp. 31-45, 2000.

[12] B. Zhang, M. Hsu, and U. Dayal, "Harmonic Average Based Clustering Method and System," US Patent 6,584,433, 2000.

[13] J. McQueen "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Statistics and Probability,* pp. 281-297, 1967.

[14] J.-C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, 1981.

[15] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, pp. 1-38, 1977.

[16] M.J. Kearns, Y. Mansour, and A.Y. Ng, "An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering," *Proc. 13th Int'l Conf. Uncertainty in Artificial Intelligence,* pp. 282-293, 1997.

[17] I. Budimir, S. Dragomir, and J. Pecaric, "Further Reverse Results for Jensen's Discrete Inequality and Applications in Information Theory," *J. Inequalities in Pure and Applied Math.,* vol. 3, 2000.

[18] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition.* Springer, 1996.

[19] H. Attias, "A Variational Bayesian Framework for Graphical Models," *Advances in Neural Information Processing Systems 12,* pp. 209-215, 1999.

[20] M.-J. Beal and Z. Ghahramani, "The Variational Bayesian EM Algorithm for Incomplete Data: With Application to Scoring Graphical Models," *Bayesian Statistics,* vol. 7, pp. 453-464, 2003.

Richard Nock received the agronomical engineering degree from the Ecole Nationale Superieure Agronomique de Montpellier, France (1993), the PhD degree in computer science (1998), and an accreditation to lead research (HDR, 2002) from the University of Montpellier II, France. Since 1998, he has been a faculty member at the Universite Antilles-Guyane in Guadeloupe and in Martinique, where his primary research interests include machine learning, data mining, computational complexity, and image processing.

Frank Nielsen defended his PhD thesis on adaptive computational geometry prepared at INRIA Sophia-Antipolis (France). In 1997, he served in the army as a scientific member in the computer science laboratory of the Ecole Polytechnique. In 1998, he joined Sony Computer Science Laboratories Inc., Tokyo, as a researcher. His current research interests include geometry, vision, graphics, learning, and optimization. He recently published the book *Visual Computing: Geometry, Graphics, and Vision* (Charles River Media/Thomson Delmar Learning, 2005).