An Abstract Weighting Framework for Clustering Algorithms

Richard Nock[*] Frank Nielsen[†]

Abstract

Recent works in unsupervised learning have emphasized the need to understand a new trend in algorithmic design, which is to influence the clustering via weights on the instance points. In this paper, we handle clustering as a constrained minimization of a Bregman divergence. Theoretical results show benefits resembling those of boosting algorithms, and bring new modified weighted versions of clustering algorithms such as k-means, expectation-maximization (EM) and k-harmonic means. Experiments display the quality of the results obtained, and corroborate the advantages that subtle data reweightings may bring to clustering.

Keywords: Statistical/optimization methods, Clustering algorithms.

1 Introduction

Recently, a new methodology in the design of supervised learning algorithms has allowed 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]. On-line 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 remarkably remained cut off from this line of works. This is all the more interesting as is it well known that Bregman divergence minimization brings weighted iterative solutions [1], and there has recently been a growing attention around weighted iterative clustering algorithms in unsupervised learning, such as k-harmonic means (k-Hmeans for short) clustering [6]. Recent approaches have even emphasized the benefits of weighting the instances in clustering [6], and make first attempts to explain the quality of the experimental results by boosting analogies [6, 7]; unfortunately, the analogy has remained so far quite loose, supported mainly by experimental results and the notice that weighting functions tend to give bigger weights to points less efficiently clustered, thereby "attracting" the cluster centers.

It is the aim of this paper to formulate clustering as a problem of constrained Bregman divergence minimization. The solution obtained has attractive boosting related theoretical features [4] such as the very fast decrease of the loss function under mild assumptions, or the clustering optimization on both the weighted and unweighted (original) instances. We also present simple weighted modifications of commonly used clustering algorithms such as k-means, EM and k-Hmeans. In that last case, the weights obtained are different from those originally presented in [6, 7]. The weighting behavior, which respect the boosting analogy of [6], displays however an original pattern when applied to a non monotonous clustering algorithm [8]: whenever the clustering gets worse, bigger weights are given to the points *more efficiently* clustered, thereby tending to penalize the clustering, making it attracted by the previous, better solutions.

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. A last Section concludes the paper.

2 Definitions and Preliminaries

The task of clustering can be presented from the density estimation standpoint, using one of its most popular representatives: k-means [9]. We dispose of a point set S of m elements and an integer $k > 0$, and the task is to estimate k densities, each defining a cluster. Each point $x \in S$ is affected to one cluster: the density on x of the cluster to which x belongs is noted for short $p(x)$. The goodness-of-fit of the clustering is obtained through its likelihood, $\prod_{x\in S} p(x)$, which we want to maximize. Equivalently, we want to minimize the following loss: $\sum_{x\in S} -\ln p(x)$. In our context of clustering, we refer for short to $-\ln p(x)$ as the

[*] Grimaag-DSI, Univ. Antilles-Guyane, Schoelcher 97275, France. E-mail: rnock@martinique.univ-ag.fr.

[†] Sony CS Labs, FRL. 3-14-13 Higashi Gotanda. Tokyo 141-0022, Japan. E-mail: Nielsen@csl.sony.co.jp.

KMN loss (on some $x \in S$), after Kearns *et al.*[10]. [10] provide a convenient abstraction of the k-means clustering algorithm as a maximum likelihood iterative procedure, which comes in handy for the derivations of more complex clustering algorithms (see Algorithm 1 below). In this algorithm, p^j denotes the density associated to cluster j ($j = 1, 2, ..., k$).

```
Algorithm 1: k-means(S, k)
    Input: point set $S$, integer $k > 0$
    Initialization: first decomposition into $k$ clusters;
    for $t = 0, 1, ...$ do
        [1.] $\forall i = 1, 2, ..., k$ :
            $S_i \leftarrow \{x \in S : i = \arg\max_j p^j(x)\}$ ;
        [2.] $\forall i = 1, 2, ..., k$ :
            $p^i \leftarrow \arg\min_p -\frac{1}{m}\sum_{x \in S_i} \ln p(x)$ ;
```

A popular method (due to E. Forgy) to initialize the cluster centers is to pick at random k points over the m points of S [6]. The k-means algorithm, which is Newton-type (thus, monotonous [8]), is typically run until the loss decrease in absolute value does not exceed a small threshold. In the sequel, bold-faces denote vector notations.

Definition 2.1. *Let $\mathcal{P}_m$ be the m-dimensional probability simplex, and $\boldsymbol{u} \in \mathcal{P}_m$ the uniform vector. For any properly defined function f and m-dimensional vector $\boldsymbol{v}$, let $f\boldsymbol{v}$ be the vector whose j^{th} component is $f(v_j)$ ($1 \leq j \leq m$).*

From these definitions, the (uniform) KMN loss on S is just

$$-\frac{1}{m}\sum_{x \in S} \ln p(x) = -\sum_{j=1}^{m} u_j \ln(p_j) = -\mathbf{u}.\ln \mathbf{p} \quad (2.1)$$

Notice the slight abuse of notation, with which we replace for x, the j^{th} element of S, $p(x)$ by p_j ($j = 1, 2, ..., m$). It is worthwhile remarking that when densities are modeled by multivariate Gaussian densities with identity covariance matrices, the minimization of eq. (2.1) to choose the centers in step [2.] of Algorithm 1 boils down to a conventional least square minimization problem equivalent to the *quantization error* minimization [9]. The new centers are obtained as the per-cluster average of the points of S [10, 9].

The k-means algorithm is based upon two essential principles that have been later on discussed and relaxed. The first one is a longstanding debate on the assignments of points to the clusters (step [1.]). The k-means chooses *hard* membership assignment, since each point belongs exactly to one cluster. Another well-known approach has proned a fractional assignment, or *soft* membership, of each point $x \in S$ to all clusters: EM [11].

The second one draws on a recent history of supervised learning and the theory of learning pioneered by Valiant [12]. A breakthrough has recently shown from both the theoretical and experimental standpoints that dramatic improvements in the performances of iterative learning algorithms are obtained when one makes subtles reweighting of the problem's instance. Probably because of the increasing popularity of these so-called *boosting* algorithms [2], some authors have recently begun to question the transfer of this property to unsupervised learning, debating on the interest of weighting the points in S to influence the choice of the next clusters (step [2.]) [6]. In the context of unsupervised learning, the main analogy motivating this question is that whenever the loss function is essentially decreasing as a function to a cluster center (such as for Gaussian priors), points with higher weights should attract the cluster centers [6]. The iterative nature of popular clustering algorithms [9, 11, 7] is certainly another motivation for this analogy, as the adaptive nature of boosting algorithms comes in part from the fact that they are iterative. These possible connexions with boosting are explored in the next Section.

3 Weighted Clustering

Let us slightly shift our view of Algorithm 1 and see it from a more general standpoint. Before convergence, the new cluster partition at step $t + 1$ ensures that $-\mathbf{u}.\ln \mathbf{p}_{t+1} < -\mathbf{u}.\ln \mathbf{p}_t$ (if we replace k-means by a still iterative but non monotonous algorithm [8], we may consider that we add a small positive penalty to the right-hand side). Here, $\mathbf{p}_t$ is the per-point density values vector after iteration t (eq. (2.1)); $\mathbf{p}_0$ is that of the algorithm's initialization step (*Cf* Algorithm 1).

Definition 3.1. *Consider a sequence of reals γ_t ($t = 0, 1, ...$) such that $\boldsymbol{u}.(\ln \boldsymbol{p}_t - \ln \boldsymbol{p}_{t+1}) = -\gamma_t$. We call γ_t the advantage at time t.*

This definition makes sense, since when $\gamma_t > 0$, the clustering gets indeed better. The distribution is absent from the γ notation, but it should be clear from context. Define

$$\forall x \in S, d_t(x) = \ln \frac{p_t(x)}{p_{t+1}(x)} \ , \quad (3.2)$$

so that we have $\mathbf{u}.\mathbf{d}_t = -\gamma_t$. From this standpoint, Algorithm 1 (as well as others, even non monotonous) is the naive procedure which takes benefit of the advantages over the uniform distribution to drive down the loss.

The problem is: what if we demand that the advantage be measured on other distributions ? is there something to gain over the uniform distribution $\mathbf{u}$? These questions may appear surprising at first glance, because $\mathbf{u}$ is the distribution which is used to measure the loss throughout t. Thereby, it is certainly the most direct way to control it. But it appears to be not the unique way. Surprisingly, sometimes, it is also not the best.

3.1 General Scheme We consider an abstraction of clustering, which is in particular a generalization of Algorithm 1. After the initialization of the first configuration (that is, the first per-point density values vector $\mathbf{p}_0$), we also fix an initial distribution $\mathbf{w}_0 = \mathbf{u}$. Algorithm 2 below shows our abstraction of Algorithm 1.

```
Algorithm 2: Adaptive-Clustering($S, k$)
    Input: point set $S$, integer $k > 0$
    Initialization:
    — first decomposition into $k$ clusters ($\mathbf{p}_0$);
    — initialization of the weights ($\mathbf{w}_0 = \mathbf{u}$);
    for $t = 0, 1, ...$ do
        [1.] pick $\mathbf{p}_{t+1}$ having advantage $\gamma_t$ over $\mathbf{w}_t$;
        [2.] compute $\mathbf{w}_{t+1}$ from $\mathbf{w}_t$;
```

Exiting the t-loop may be obtained when t reaches a threshold T, or when no computationally available $\mathbf{p}_{t+1}$ has advantage significantly different from zero. Notice that Algorithm 1 is indeed a particular case of Algorithm 2, in which $\forall t \geq 0, \mathbf{w}_t = \mathbf{u}$, and finding $\mathbf{p}_{t+1}$ is obtained by steps [1.] and [2.] of Algorithm 1. Furthermore, a non-negative advantage γ_t is always guaranteed for Algorithm 1 by the fact that it is Newton-type [8].

Let us detail and explain the loop step of Algorithm 2. In step [1.], we want to pick $\mathbf{p}_{t+1}$ such that

$$(3.3) \qquad \mathbf{w}_t.\mathbf{d}_t \quad = \quad -\gamma_t \ .$$

This is equivalent to stating that

$$\sum_{x \in S : d_t(x) < 0} w_t(x)|d_t(x)| \quad = \quad \gamma_t + \sum_{x \in S : d_t(x) > 0} w_t(x)|d_t(x)| \ .$$

By means of words, when $\gamma_t > 0$, the explanation of 3.3 is simple : in absolute value for each $x \in S$, the weighted sum of loss decrease on $\mathbf{w}_t$ is the weighted sum of loss increase *plus a positive advantage.* Thus, the KMN loss, when measured on $\mathbf{w}_t$ (and not on $\mathbf{u}$ like in eq. (2.1)), decreases, and $\mathbf{p}_{t+1}$ is chosen so as to make the next clustering have at least a small gain on $\mathbf{w}_t$. Such an assumption of guaranteed small gain in a weighted iterative learning algorithm is the cornerstone of *boosting.* Its related name is the *weak learning assumption* [2, 4]. Let us recall that boosting is primarily a methodology for the iterative combination of classifiers: the weak learning assumption states that each local classifier has accuracy only slightly better than random on the sample's weights on which it is built. The power of boosting is to make subtle updates in the weights so as to obtain, from each of these locally weak classifiers, a combination of arbitrary high accuracy, as measured on the *initial* distribution (*e.g.* $\mathbf{u}$, [2, 4]). Provided similar behaviors can be obtained in the context of clustering, a natural name for eq. (3.3) when $\gamma_t > 0$ should thus be a *weak clustering assumption.*

Let us shift back to Algorithm 2, and detail step [2.], the computation of $\mathbf{w}_{t+1}$. We wish to find $\mathbf{w}_{t+1}$ which satisfies two constraints. The first one is straightforward: it expresses the fact that $\mathbf{w}_{t+1}$ is a distribution ($\mathbf{w}_{t+1} \in \mathcal{P}_m$):

$$(3.4) \qquad \mathbf{1}.\mathbf{w}_{t+1} - 1 \quad = \quad 0 \ .$$

The second constraint expresses its decorrelation with respect to the variations of the weighted KMN loss:

$$(3.5) \qquad \mathbf{w}_{t+1}.\mathbf{d}_t \quad = \quad 0 \ .$$

Informally, after the computation of the new weights in $\mathbf{w}_{t+1}$, because of the fact that the weighted KMN loss on $\mathbf{p}_t$ and $\mathbf{p}_{t+1}$ remains the same when measured on $\mathbf{w}_{t+1}$ (eq. (3.5)), we are somewhat forcing the choice of the next densities in $\mathbf{p}_{t+2}$ (eq. (3.3) with $t \rightarrow t+1$) to learn something "new" from S.

Remember that $m.\mathbf{u} = \mathbf{1}$. Under constraints (3.4) and (3.5), $\mathbf{w}_{t+1}$ is chosen so as to minimize a Bregman divergence with respect to $\mathbf{w}_t$: the *information divergence* [1], $\mathbf{1}.\mathbf{i}(\mathbf{w}_{t+1}, \mathbf{w}_t)$, with $\mathbf{i}(.,.)$ the vector whose component for some $x \in S$ is:

$$\begin{aligned} \mathbf{i}(\mathbf{w}_{t+1}, \mathbf{w}_t)(x) \quad &= \quad w_{t+1}(x) \ln \frac{w_{t+1}(x)}{w_t(x)} \\ &\quad -w_{t+1}(x) + w_t(x) \ . \end{aligned}$$

The information divergence is convex in $\mathbf{w}_{t+1}$: its minimization under constraints (3.4) and (3.5) is obtained as the solution to ($\forall x \in S$):

$$\begin{aligned} (3.6) \qquad & \partial_{\mathbf{w}_{t+1}} \mathbf{i}(\mathbf{w}_{t+1}, \mathbf{w}_t)(x) \\ + \left[b_t \partial_{\mathbf{w}_{t+1}} (\mathbf{1}.\mathbf{w}_{t+1} - 1) + c_t \partial_{\mathbf{w}_{t+1}} (\mathbf{w}_{t+1}.\mathbf{d}_t) \right] (x) \quad &= \quad 0 \end{aligned}$$

with b_t and c_t Lagrange multipliers. Solving (3.6) for $\mathbf{w}_{t+1}$ brings

$$(3.7)\ \forall x \in S, w_{t+1}(x) \quad = \quad \frac{w_t(x)\exp(-c_t d_t(x))}{\exp(b_t(c_t))} \quad .$$

In (3.7), $b_t(.)$ is called the cumulant function, whose expression is obtained with constraint (3.4):

$$(3.8)\ b_t(c) = \ln \sum_{x\in S} w_t(x)\exp(-cd_t(x)) \quad (c \in I\!\!R) \ .$$

The term inside the "ln" is the normalization coefficient for $\mathbf{w}_{t+1}$:

$$(3.9) \qquad Z_t(c) \quad = \quad \sum_{x\in S} w_t(x)\exp(-cd_t(x)) \ .$$

The last unknown, c_t, is obtained from constraint (3.5) as the unique solution to

$$(3.10) \quad \sum_{x\in S} w_t(x)d_t(x)\exp(-c_t d_t(x)) \quad = \quad 0 \ .$$

The next Subsection shows some properties of Algorithm 2, the first of which is the proof that eq. (3.10) has indeed a single solution.

3.2 Properties of the solution to eq. (3.10)

Define for short $g(c) = -\partial Z_t/\partial c$. Eq. (3.10) is equivalent to stating:

$$g(c_t) \quad = \quad 0 \ .$$

Lemma 3.1. *If $\exists x \in S : d_t(x) > 0$ and $\exists x \in S : d_t(x) < 0$, then eq. (3.10) has a single solution.*

Proof. We have $\forall c \in I\!\!R$:

$$(3.11)\ g'(c) \quad = \quad -\sum_{x\in S} w_t(x) d_t^2(x)\exp(-cd_t(x)) \\ < \quad 0 \ .$$

Since $\lim_{c\rightarrow -\infty} g(c) = +\infty$ (provided at least one x has $d_t(x) > 0$), and $\lim_{c\rightarrow +\infty} g(c) = -\infty$ (provided at least one x has $d_t(x) < 0$), there is indeed a single solution to (3.10). This ends the proof of Lemma 3.1.

Lemma (3.1) shows that eq. (3.10) has a single solution, but it does not states where c_t lies in $I\!\!R$. Without more information, searching for even approximate solutions might represent a considerable complexity burden at the data mining scale. Fortunately, we show that c_t lies on an interval of reasonable measure, with efficient and simple approximation algorithms. $\forall \ell \in \{+,-\}$, define

$$\begin{aligned}
\underline{d}_t^\ell &= \min_{x\in S:\ell d_t(x)>0} |d_t(x)| \\
\overline{d}_t^\ell &= \max_{x\in S:\ell d_t(x)>0} |d_t(x)| \\
D_t^\ell &= \sum_{x\in S:\ell d_t(x)>0} w_t(x)|d_t(x)| \\
\underline{c}_t &= -\frac{1}{\underline{d}_t^- + \underline{d}_t^+}\ln\frac{D_t^-}{D_t^+} \\
\overline{c}_t &= -\frac{1}{\overline{d}_t^- + \overline{d}_t^+}\ln\frac{D_t^-}{D_t^+}
\end{aligned}$$

Lemma 3.2.

$$(3.12) \qquad c_t \quad \in \quad [\min\{\underline{c}_t, \overline{c}_t\}, \max\{\underline{c}_t, \overline{c}_t\}] \ .$$

Proof. Remark that $g(0) = \mathbf{w}_t.\mathbf{d}_t = -\gamma_t$ from constraint (3.3). When $\gamma_t = 0$, $g(0) = 0$ and thus $c_t = 0 = \underline{c}_t = \overline{c}_t$. Suppose now that $\gamma > 0$. Since $g(0) < 0$, $c_t < 0$ and We have

$$\begin{aligned}
\sum_{x\in S:d_t(x)>0} w_t(x)d_t(x)\exp(-c_t d_t(x)) &\leq \exp(-c_t\overline{d}_t^+)D_t^+ \\
\sum_{x\in S:d_t(x)<0} w_t(x)|d_t(x)|\exp(-c_t d_t(x)) &\geq \exp(c_t\overline{d}_t^-)D_t^- \ .
\end{aligned}$$

Furthermore,

$$\begin{aligned}
g(c_t) \quad = & \sum_{x\in S:d_t(x)>0} w_t(x)d_t(x)\exp(-c_t d_t(x)) \\
& - \sum_{x\in S:d_t(x)<0} w_t(x)|d_t(x)|\exp(-c_t d_t(x)) \\
= & \ 0 \ .
\end{aligned}$$

We obtain $\exp(-c_t\overline{d}_t^+)D_t^+ \geq \exp(c_t\overline{d}_t^-)D_t^-$, thus $c_t \leq -(1/(\overline{d}_t^+ + \overline{d}_t^-))\ln(D_t^-/D_t^+)$. We also have

$$\begin{aligned}
\sum_{x\in S:d_t(x)>0} w_t(x)d_t(x)\exp(-c_t d_t(x)) &\geq \exp(-c_t\underline{d}_t^+)D_t^+ \\
\sum_{x\in S:d_t(x)<0} w_t(x)|d_t(x)|\exp(-c_t d_t(x)) &\leq \exp(c_t\underline{d}_t^-)D_t^- \ ,
\end{aligned}$$

from which we get $\exp(c_t\underline{d}_t^-)D_t^- \geq \exp(-c_t\underline{d}_t^+)D_t^+$, and $c_t \geq -(1/(\underline{d}_t^+ + \underline{d}_t^-))\ln(D_t^-/D_t^+)$. The cases when $\gamma_t < 0$ are obtained in the same way (end of the proof of Lemma 3.2).

Lemma 3.2 shows that c_t may be approximated through a simple dichotomic search. Its computational complexity is very reasonable, as we now explain. Suppose that we wish to approximate c_t by some $\hat{c}_t$ such that $|c_t - \hat{c}_t|/|c_t| \leq \epsilon$. Then, the number of dichotomic steps to beat error ϵ is only $\mathcal{O}(\log((\overline{d}_t^+ + \overline{d}_t^-)/(\underline{d}_t^+ + \underline{d}_t^-)) + \log(1/\epsilon))$.

3.3 Properties of Algorithm 2 We now explain why the clustering bias, as obtained through distribution $\mathbf{w}_t$, may also help to obtain substantial gains over $\mathbf{u}$. An interest in minimizing the information divergence subject to the decorrelation with the $\mathbf{d}_t$'s is explained in the next Lemma. Its proof is straightforward once we remark that

$$\begin{aligned}(3.13)\quad -c_t\mathbf{w}_t.\mathbf{d}_t &= -\mathbf{1}.\mathbf{i}(\mathbf{w}_t, \mathbf{w}_{t+1}) \\ &\quad -\mathbf{1}.\mathbf{i}(\mathbf{w}_{t+1}, \mathbf{w}_t)\ , \\ (3.14)\quad -c_t\mathbf{u}.\mathbf{d}_t &= \mathbf{1}.\mathbf{i}(\mathbf{u}, \mathbf{w}_t) - \mathbf{1}.\mathbf{i}(\mathbf{u}, \mathbf{w}_{t+1}) \\ &\quad -\mathbf{1}.\mathbf{i}(\mathbf{w}_{t+1}, \mathbf{w}_t)\ .\end{aligned}$$

In the sequel, we replace for the sake of readability $Z_t(c_t)$ by Z_t (eq. (3.9)). We also suppose that Algorithm 2 is ran for $T > 0$ clustering rounds.

Lemma 3.3.

$$\begin{aligned}(3.15)\quad \boldsymbol{1}.\boldsymbol{i}(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) &= \ln(1/Z_t)\ , \\ (3.16)\quad (-c_t)\boldsymbol{w}_t.\boldsymbol{d}_t &\leq -\ln(1/Z_t)\ , \\ (3.17)\quad \sum_{t\leq T}(-c_t)\boldsymbol{u}.\boldsymbol{d}_t &\leq \sum_{t\leq T} -\ln(1/Z_t)\ .\end{aligned}$$

Let us name in the sequel $\mathbf{1}.\mathbf{i}(\mathbf{w}_{t+1}, \mathbf{w}_t)$ the information divergence "remainder", as it is what remains after minimization to find $\mathbf{w}_{t+1}$. Lemma 3.3 shows the equivalence between maximizing the advantage γ_t (eq. (3.3)), minimizing the normalization coefficient Z_t, and maximizing this information divergence remainder. Let us concentrate on Z_t. From Lemma 3.3, if Z_t is small (say, < 1), then we may indeed expect gains on both $\mathbf{w}_t$ and $\mathbf{u}$.

Lemma 3.4. *$\gamma_t \neq 0$ implies $Z_t < 1$. Furthermore, $\gamma_t c_t \leq 0$ with equality iff $c_t = \gamma_t = 0$.*

More than the fact that $Z_t < 1$, an upperbound on its value is also useful as we shall see that the quantity $\prod_{t\leq T} Z_t$ is meaningful to appreciate the fraction of "bad" points in S, *i.e.* those for which the loss does not decrease significantly. We now give such an indication on how slowly Z_t approaches 1.

Lemma 3.5. *If $\forall x \in S, |\bar{c}_t d_t(x)|$ is small enough, then $\exists k > 0$ a constant such that*

$$(3.18)\quad Z_t(c_t) \leq 1 + k\gamma_t\bar{c}_t\ .$$

Proof. We use the fact that $Z_t(c_t) \leq Z_t(\bar{c}_t)$ and $\forall d_t(x) \in \mathbb{R}$, $\exp(-\bar{c}_t d_t(x)) = 1 - \bar{c}_t d_t(x) + (\bar{c}_t d_t(x))^2/2! - (\bar{c}_t d_t(x))^3/3! + \ldots$ (end of the proof of Lemma 3.5).

Even when the assumption of Lemma 3.5 is strong, notice that $\bar{c}_t$ is itself not very large, as we have $|\bar{c}_t| \leq \gamma_t/(D_t^+(\overline{d}_t^- + \overline{d}_t^+))$. Furthermore, $\gamma_t\bar{c}_t < 0$ for any non-zero advantage (Lemmata 3.2 and 3.4), which may lead indeed to Z_t small enough to guarantee a very fast vanishing of $\prod_{t\leq T} Z_t$.

The weak clustering assumption is basically local, since it postulates that at step t, there will be at least a slim gain for clustering, say for some fraction of S. What Lemma 3.3 brings is that when all $\gamma_t > 0$, these small gains cannot cancel each other, as they are guaranteed to sum up and bring significant gains over each $\mathbf{w}_t$, but also over $\mathbf{u}$.

Now, what happens when some $\gamma_t < 0$, *i.e.* when the clustering gets locally worse on $\mathbf{w}_t$? We now show that its degradation on $\mathbf{u}$ is actually *smaller*. We have [1]

$$\begin{aligned}(3.19)\quad \mathbf{1}.\mathbf{i}(\mathbf{u}, \mathbf{w}_t) &\geq \mathbf{1}.\mathbf{i}(\mathbf{u}, \mathbf{w}_{t+1}) \\ &\quad +\mathbf{1}.\mathbf{i}(\mathbf{w}_{t+1}, \mathbf{w}_t)\ .\end{aligned}$$

The proof of the next Lemma is based on Lemma 3.4, and (3.13), (3.14), (3.19).

Lemma 3.6. *$\gamma_t \leq 0 \Rightarrow \boldsymbol{u}.\boldsymbol{d}_t \leq \boldsymbol{w}_t.\boldsymbol{d}_t - (\boldsymbol{1}.\boldsymbol{i}(\boldsymbol{w}_t, \boldsymbol{w}_{t+1}) + \boldsymbol{1}.\boldsymbol{i}(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t))/c_t$.*

Fix for short $L_t = -\mathbf{u}.\ln\mathbf{p}_t$ $(t \geq 0)$ as the KMN loss at time t, and $\delta_t = c_t - c_{t-1}$ $(t > 0)$. The next Lemma extends ineq. (3.17) by providing an upperbound on the KMN loss for some time $T+1 > 0$ with $\gamma_T > 0$.

Lemma 3.7. *$\forall T > 0$, if $\gamma_T > 0$ then*

$$L_{T+1} \leq L_0 + \sum_{t=1}^{T}\frac{\delta_t(L_t - L_0)}{c_T} + \frac{1}{c_T}\sum_{t=0}^{T}\ln\frac{1}{Z_t}\ .$$

Let us write this bound as $L_{T+1} \leq L_0 + a_1(T) + a_2(T)$. Since $\gamma_T > 0$, $c_T < 0$, and thus $a_2(T) < 0$. Furthermore, since each $Z_t \leq 1$, each iteration may help to improve the clustering through $a_2(T)$ (each term in the sum is ≤ 0). Let us concentrate on $a_1(T)$. Eqs. (3.14) and (3.19) bring $-c_t\mathbf{u}.\mathbf{d}_t \leq 0$. Consider for the sake of simplicity that each $\delta_t \leq 0$ (thus, $c_t \leq c_{t-1} \leq 0$). This yields $\mathbf{u}.\mathbf{d}_t \leq 0$, thus, $L_t - L_0 \leq 0$. We obtain that $a_1(T)$ is also negative, and each step also contributes to the KMN loss decrease.

Let us compare this decrease to that of the unweighted (naive) clustering which would repeatedly minimizes L_t. We have $L_{T+1} = L_0 + \sum_{t=1}^{T+1}(L_t - L_{t-1})$, which we write for short $L_{T+1} = L_0 + a_3(T)$. Notice that Lemma (3.7) exhibits two sources of loss decrease ($a_1(T)$ and $a_2(T)$) instead of only one for the naive approach ($a_3(T)$). Consider for the sake of simplicity that there is a constant, strictly negative loss decrease $L_t - L_{t-1} = \eta < 0$ for both approaches. Fix $t^* = \arg\min_{1\leq t\leq T} \delta_t$ as the iteration of the worst weighted decrease. Then, we would have $a_1(T) < a_3(T)$ provided $T(T+1)\delta_{t^*}\eta/(2c_T) \leq (T+1)\eta$, that is,

$$c_{t^*} \quad \leq \quad c_{t^*-1} + 2\frac{c_T}{T} \ \ .$$

We also have $\lim_{T\rightarrow +\infty} c_T/T = 0$; since all δ_t are non positive, the constraint for the weighted clustering to be better than the unweighted clustering vanishes as T increases. Notice that we did not even make use of $a_2(T)$, which is also < 0.

We now focus on boosting-like properties that Algorithm 2 may exhibit. The next Lemma gives, on our unsupervised learning setting, the equivalent of a well known boosting Theorem (Th. 1 in [4]).

Lemma 3.8. *$\forall T > 0$, if $\gamma_T > 0$, then*

$$\frac{|\{x \in S : p_{T+1}(x) \leq (p_0)^{\frac{c_0}{c_T}}(x) \prod_{t=1}^{T} (p_t)^{\frac{\delta_t}{c_T}}(x)\}|}{m} \leq \prod_{t\leq T} Z_t \ \ .$$

Under the hypothesis that $\gamma_t \geq 0, \forall t < T$ and $\delta_t \leq 0, \forall t \leq T$, we get

$$\begin{aligned}
\forall t \leq T, \frac{\delta_t}{c_T} \quad &\in \quad [0,1] \\
\frac{c_0}{c_T} \quad &\in \quad [0,1] \\
\frac{c_0}{c_T} + \sum_{t\leq T} \frac{\delta_t}{c_T} \quad &= \quad 1
\end{aligned}$$

Fix for short $\alpha_0 = c_0/c_T$ and $\alpha_t = \delta_t/c_T$ $(1 \leq t \leq T)$. Lemma 3.8 can be reformulated to integrate the loss functions L_t, as:

$$\frac{|\{x \in S : L_{T+1}(x) \geq \sum_{t=0}^{T} \alpha_t L_t(x)\}|}{m} \quad \leq \quad \prod_{t\leq T} Z_t \ \ .$$

Therefore, Lemma 3.8 states that there is a very fast (exponential, *Cf* Lemma 3.5) decrease of the number of points in S for which the loss at time $T+1$ is not smaller than the weighted average of all their previous losses. Note that the weights α_t emphasize the best iterations (for which the decrease of c_t is the largest). Thus, it tends to strengthen this phenomenon.

Lemma 3.8 brings a geometric mean inequality on the densities. Using the AGH-inequality, it immediately implies the same result for the harmonic mean [13]. It is also possible to obtain a more natural arithmetic mean inequality, using a reverse of the AGH-inequality [13]: denote $\underline{\alpha}_T = \min_{0\leq t\leq T:\alpha_t>0} \alpha_t$ and $\overline{\alpha}_T = \max_{0\leq t\leq T} \alpha_t$. Then we also have:

$$\frac{|\{x \in S : p_{T+1}(x) \leq e^{-\frac{(\overline{\alpha}_T - \underline{\alpha}_T)^2}{4\underline{\alpha}_T\overline{\alpha}_T}} \sum_{t=0}^{T} \alpha_t p_t(x)\}|}{m} \leq \prod_{t\leq T} Z_t \ \ .$$

3.4 Clustering under constant advantage The next Lemma is a direct, simple consequence of Lemmata 3.7, 3.8. It holds under an assumption resembling that of the constant learning rate in supervised learning [1].

Lemma 3.9. *Fix some $T > 0$ for which the following holds: $\gamma_T > 0$, $\gamma_t \geq 0$ ($\forall 0 \leq t < T$) and $c_t = c$ ($\forall 0 \leq t \leq T$). Then we have*

$$L_T \quad \leq \quad L_0 + \frac{1}{c} \sum_{t\leq T-1} \ln\frac{1}{Z_t} \ \ , \tag{3.20}$$

and

$$\frac{|\{x \in S : L_T(x) \geq L_0(x)\}|}{m} \quad \leq \quad \prod_{t\leq T} Z_t \ \ . \tag{3.21}$$

Since $c < 0$, this Lemma says that there is a significant global decrease of the loss function L_t through the iterations (eq. (3.20)), but also the fraction of points for which this loss does not decrease vanishes very rapidly ($Z_t < 1, \forall t$, see Lemma 3.5).

3.5 Hard and Soft Membership Assignments for $\mathbf{p}_{t+1}$ Choosing $\mathbf{p}_{t+1}$ having (whenever possible) positive advantage over $\mathbf{p}_t$ on $\mathbf{w}_t$ (constraint (3.3)) may be easily obtained when treating it as multivariate Gaussian densities with identical covariance matrix. Since the last partition ($\mathbf{p}_t$) is fixed, we only have to minimize $-\mathbf{w}_t.\mathbf{p}_{t+1}$, and this amounts to a least square solution for the cluster centers [10]. The hard membership (k-means) solution is:

$$\arg\min_{\mu_1,...,\mu_k\in I\!\!R^n} \sum_{i=1}^{k} \sum_{x\in S_i} w_t(x) \parallel x - \mu_i \parallel^2 \ \ . \tag{3.22}$$

Solving (3.22) yields

$$\begin{aligned}
&\forall i = 1, 2, ..., k, \\
&\mu_i \quad = \quad \frac{\sum_{x\in S_i} w_t(x).x}{\sum_{x\in S_i} w_t(x)} \ \ . \qquad (3.23)
\end{aligned}$$

If we integrate the weight update between steps [1.] and [2.] in Algorithm 1, then we obtain a *weighted k-means* clustering algorithm, which is *not* monotonous, and therefore not Newton-type (because between each weight modification, both the points of S and the cluster centers get reallocated). Thus, there are two possible behaviors. If $c_t < 0$, the new weights put greater emphasis on points less efficiently clustered so far. However, if $c_t > 0$ (the clustering gets worse), the new weights put greater emphasis on points more efficiently clustered so far, thereby tending to correct the smaller quality of the clustering found.

The same reasoning allows to obtain the soft membership (EM) solution [10]. If we put a fractional assignment of each $x \in S$ to a cluster $i \in \{1, 2, ..., k\}$ with density $q(i|x)$ (with $\sum_i q(i|x) = 1$), then we replace problem (3.22) by:

$$(3.24)\quad \arg \min_{\mu_1,...,\mu_k \in I\!R^n} \sum_{i=1}^{k} \sum_{x \in S} w_t(x) q(i|x) \parallel x - \mu_i \parallel^2 \ .$$

Solving (3.24) brings our *weighted EM* solution for the cluster centers:

$$\forall i = 1, 2, ..., k,$$

$$(3.25)\qquad \mu_i \quad = \quad \frac{\sum_{x \in S} w_t(x) q(i|x).x}{\sum_{x \in S} w_t(x) q(i|x)} \ .$$

Finally, notice that $\mathbf{p}_t$ does not need to be a density: we only need $\mathbf{d}_t$ to be defined on S. For instance, suppose that the loss on one $x \in S$ depends on the location of all cluster centers instead of just one [6]. In that case, the squared Euclidean distance may be replaced by the harmonic mean [6] ($\forall a \in I\!R^{+,*}$):

$$(3.26)\qquad \forall x \in S, f_a(x) \quad = \quad \frac{k}{\sum_{i=1}^{k} \frac{1}{\|x - \mu_i\|^a}} \ .$$

The density on x can be replaced by $\exp - f_a(x)$: even when this is not a proper density, we can still solve (3.22), and get the corresponding center update. Define

$$\forall i = 1, 2, ..., k, \forall x \in S$$

$$g_i(x) = \frac{1}{\|x - \mu_i\|^{a+2} \left(\sum_{j=1}^{k} \frac{1}{\|x - \mu_j\|^a} \right)^2} \ .$$

Then, we get

$$\forall i = 1, 2, ..., k,$$

$$(3.27)\qquad \mu_i \quad = \quad \frac{\sum_{x \in S} w_t(x) g_i(x).x}{\sum_{x \in S} w_t(x) g_i(x)} \ .$$

This brings our *weighted k-Hmeans* algorithm for the cluster centers, whose weighting scheme appears to be much different from the original k-Hmeans [6, 7].

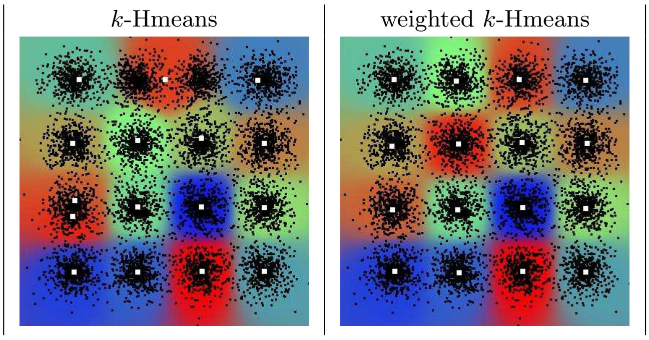

Figure 2: BIRCH configurations ($K = k = 16$) for soft memberships after 26 iterations: our weighted-k-Hmeans hits all theoretical clusters, while the usual k-Hmeans does not (see text for graphical conventions).

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-Hmeans), but we keep the term "original" for these algorithms in order to avoid confusion with our modified weighted versions. There are various datasets used for our experimental comparisons, but in order to make fair comparisons, the weighted and original versions are run on the same datasets, and initialized with the *same* set of empirical cluster centers. Thus, the differences in results stem from differences in the weighting strategies, and since the algorithms are deterministic, even relatively small variations in the results may actually denote different behaviors and significant variations in the results quality.

4.1 k-Hmeans vs weighted k-Hmeans We ran the original k-Hmeans and our weighted version on a simulated dataset of choice for the evaluation of harmonic clustering, BIRCH [6, 7]. The dataset consists of a set of 2D clusters, whose centers are located on a $\sqrt{K} \times \sqrt{K}$ grid. Here, K denotes the number of theoretical clusters. The distance between two adjacent cluster means is $4\sqrt{2}$ with cluster radius of $\sqrt{2}$ (*i.e.* the variance in each direction is 1). We fix $k = K$, and pick K to be a squared between 3^2 and 20^2. We also chose $m = 10000$. After [6, 7], the comparison measure is the number of theoretical clusters "hit" by empirical centers [6, 7]. The bigger this measure and the better the algorithms. The motivation for this choice stems from [6, 7]: the harmonic loss function experimentally helps to spread more rapidly the centers; the hits appreciate both its speed and efficiency in spreading. Figure 2 displays graphically two examples of configurations for

K	9	16	25	36	49	64	81	100	121	144	169	196	324	361	400
$C_{K-\text{Hmeans}}$	9	15	24	33	46	60	74	92	113	129	158	181	298	324	371
$I_{K-\text{Hmeans}}$	25	-	15	21	-	**16**	**24**	-	30	-	-	-	-	-	-
$C_{\mathbf{w}-K-\text{Hmeans}}$	9	**16**	24	33	**47**	60	74	**93**	113	**130**	**159**	**182**	**302**	**325**	**372**
$I_{\mathbf{w}-K-\text{Hmeans}}$	**16**	-	**9**	**8**	-	27	**24**	-	**15**	-	-	-	-	-	-

Figure 1: Number (C) of theoretical clusters hit by a center after 50 iterations, for K-Hmeans and weighted-k-Hmeans ($\mathbf{w} - K -$ Hmeans). When $C_{K-\text{Hmeans}} = C_{\mathbf{w}-K-\text{Hmeans}}$, the value $I_.$ gives the smallest iteration for which $C_.$ is obtained. Boldfaces denote the best results.

both the original algorithm and our modified weighted harmonic clustering. Because harmonic clustering uses a soft-membership function, each cluster center found is associated to a random color, and each pixel of the image displays the value of the soft membership function computed on its coordinates. The thick white dots show the center of the empirical clusters. The graphical result of Figure 2 clearly displays that weighted-k-Hmeans outperforms K-Hmeans on this BIRCH configuration, as it hits all 16 theoretical clusters after 26 iterations, while K-Hmeans does not (and still does not hit them all after 50 iterations). Synthetic results are presented on Figure 1 for experiments on 15 BIRCH configurations. They display the superiority of our weighted-k-Hmeans algorithm. Indeed, it is beaten by the original k-Hmeans only on 1 configuration out of the 15.

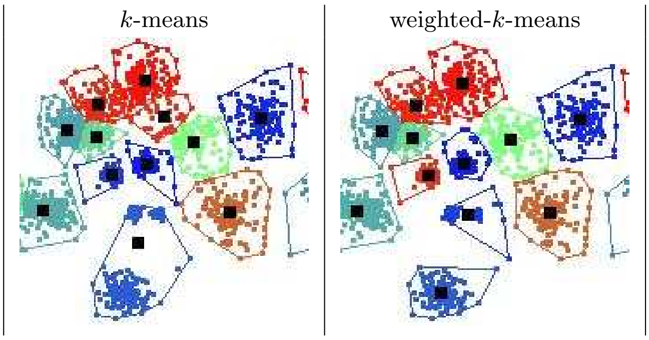

Figure 3: Crop of a clustering ($K = 60, k = 50$) for k-means vs weighted-k-means: the KMN loss is 19.64% smaller for our weighted algorithm.

4.2 k-means vs weighted k-means We first run k-means and weighted k-means on datasets with $m = 10000$ points, with $K \in \{10, 20, ..., 400\}$ theoretical clusters. The clusters are generated by spherical Gaussians with varying covariance matrix. The data are 2D. For each K, we pick $k \in \{10, 20, ..., 100\}$ experimental clusters (this makes 400 runs for each algorithm). The KMN loss of each algorithm is computed after $T = 20$ iterations, and we keep all configurations for which the relative difference in the KMN loss is $> 1\%$ in absolute value (otherwise, there is no significant visible difference). We obtain 33 configurations, 13 of which are in disfavor of the weighted algorithm. A simple sign test reveals a threshold probability $p = 14.81\%$ for rejecting the hypothesis that the weighted algorithm is not better.

Then, we run again the algorithms with the same parameters, but on clusters with less overlap. It is indeed well known that less overlaps tend to "trap" k-means on worse local optima [6]. Figure 3 shows a crop of a dataset we obtained. Thick black dots denote the cluster centers, and convex hulls delimit the clusters found. To summarize, *109* iterations display a difference $> 1\%$, *42* of which are in disfavor of the weighted algorithm. The threshold probability is now $p \approx 1.05\%$, which tends to display the ability of the weighted algorithm to reduce the hardness of clustering problems with less overlaps.

Recall that the starting point for both the weighted and the unweighted k-means are exactly the same: same dataset *and* same starting clusters. Furthermore, the KMN loss for Gaussian priors and hard membership function boils down to computing for each point the squared Euclidean distance to its cluster center, and then averaging over all points of S. Thus, a 1% difference threshold in the KMN loss is in fact significant to account for a difference between the algorithms, from both the visual and statistical standpoints.

We have experimented the efficiency in spreading the empirical cluster centers, using the hit numbers, in the same way as for harmonic clustering above. We have generated 3D ring Gaussians, that is, spherical Gaussians whose points are translated by some radius r (fixed for each theoretical cluster) on some random axis passing on the cluster's expectation. We have generated datasets with $m = 3000$ points and $K \in \{5, 10, ..., 55\}$. For each $k \in \{5, 15, 25, 35\}$, we have ran both k-means and weighted-k-means. We have computed the average number of theoretical clusters hit by a cluster center found, and have computed the number of couples

k	win/tie/lose
5	2/9/0
15	6/4/1
25	3/3/5
35	7/3/1
Tot.	18/19/7

Figure 4: Score of weighted-k-means against k-means for the hit numbers on 3D ring Gaussians (see text for details).

k	win/tie/lose
35	3/7/1
40	4/7/0
45	5/6/0
50	5/5/1
55	3/8/0
Tot.	20/33/2

Figure 5: Score of weighted-k-means against k-means for the hit numbers on 4D spherical Gaussians (see text for details).

(k, K) for which this score is in favor of weighted-k-means ("win"), is in disfavor ("lose"), or for which both algorithms maintain on average the same number of hits ("tie"). Figure 4 shows the results obtained. From these results, it comes that on more than 40% of the runs, the weighted version of k-means succeeds in making a better attraction of the cluster centers, while it fails on less than 16% of them. To complete this experiment, we have ran again weighted-k-means and k-means on 4D spherical Gaussians, with larger values for k, to see what happens when one requests more clusters. Figure 5 shows the results obtained. In this case, weighted-k-means wins on 36.4% of the runs, while it is beaten only on 3.6% of them. This ratio of *10* in favor of weighted-k-means tends to prove that the weighting scheme tends to spread more rapidly the centers of clusters, in the same way as it does for harmonic clustering (*Cf* Subsection 4.1). This also explains partially why the difference between weighted-k-means and k-means is sharpened on datasets with less overlaps between theoretical clusters: the rapidity in spreading the centers may help to escape the poor local optima on which the k-means are trapped.

5 Conclusion

Recent papers in unsupervised learning have put a great emphasis in trying to bring to clustering the recent breakthrough of a supervised learning technique that has allowed 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 dataset, with the hope to get better final solutions, and get them faster than without reweighting. In fact, some of the essential reasons for this motivation are purely conceptual but quite appealing, as it seems indeed natural that points less efficiently clustered so far may "attract" the clusters on the next rounds, and thus receive bigger weights [6, 7].

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 in supervised learning. We have proposed an abstract iterative clustering scheme that, coupled to some particular reweighting scheme, may indeed bring significant improvements on unweighted clustering from the theoretical standpoint. This iterative clustering scheme can be specialized to bring weighted variants of k-means, EM, and even harmonic means clustering [6, 7]. Our experiments display the ability of the weighted algorithms to obtain better solutions, but also to obtain them faster than for the non-modified clustering algorithms.

6 Acknowledgments

R. Nock gratefully acknowledges a visiting grant from Sony CSL Tokyo. The authors would like to thank J.-C. Janodet and T. Boudellal for comments on this paper.

References

[1] C. Gentile and M. Warmuth, "Proving relative loss bounds for on-line learning algorithms using Bregman divergences," in *Tutorials of the 13th International Conference on Computational Learning Theory*, 2000.

[2] Y. Freund and R. E. Schapire, "A Decision-Theoretic generalization of on-line learning and an application to Boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[3] J. Kivinen and M. Warmuth, "Boosting as entropy projection," in *Proc. of the 12th International Conference on Computational Learning Theory*, 1999, pp. 134–144.

[4] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," in *Proc. of the 11th International Conference on Computational Learning Theory*, 1998, pp. 80–91.

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

[6] G. Hammerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proc. of the 11th International Conference on Information and Knowledge Management*, 2002, pp. 600–607.

[7] B. Zhang, M. Hsu, and U. Dayal, "k-harmonic means - a spatial clustering algorithm with boosting," in *Temporal, Spatial, and Spatio-Temporal Data Mining*, J. F.

Roddick and K. Hornsby, Eds., pp. 31–45. Springer Verlag, 2000.

[8] L. Bottou and S. Bengio, "Convergence properties of the k-means algorithm," in *Advances in Neural Information Processing Systems 7*, 1995, pp. 585–592.

[9] J. McQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the 5th Berkeley symposium on mathematical statistics and probability*, 1967, pp. 281–297.

[10] M. J. Kearns, Y. Mansour, and A. Y. Ng, "An information-theoretic analysis of hard and soft assignment methods for clustering," in *Proc. of the 13 th International Conference on Uncertainty in Artificial Intelligence*, 1997, pp. 282–293.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. of the Royal Stat. Soc. B*, vol. 39, pp. 1–38, 1977.

[12] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, pp. 1134–1142, 1984.

[13] I. Budimir, S. S. Dragomir, and J. Pečarič, "Further reverse results for Jensen's discrete inequality and application in information theory," *Journal of Inequalities in Pure and Applied Mathematics*, vol. 2, 2001, article 5.