Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Weighted Clustering

Margareta Ackerman
University of Waterloo
mackerma@cs.uwaterloo.ca

Shai Ben-David
University of Waterloo
shai@cs.uwaterloo.ca

Simina Brânzei
Aarhus University
simina@cs.au.dk

David Loker
University of Waterloo
dloker@cs.uwaterloo.ca

Abstract

We investigate a natural generalization of the classical clustering problem, considering clustering tasks in which different instances may have different weights. We conduct the first extensive theoretical analysis on the influence of weighted data on standard clustering algorithms in both the partitional and hierarchical settings, characterizing the conditions under which algorithms react to weights. Extending a recent framework for clustering algorithm selection, we propose intuitive properties that would allow users to choose between clustering algorithms in the weighted setting and classify algorithms accordingly.

Introduction

Many common applications of clustering, such as facility allocation and vector quantization, may naturally be cast as weighted clustering tasks - tasks in which some data points should have a greater effect on the utility of the clustering than others. Consider vector quantification that aims to find a compact encoding of signals that has low expected distortion. The accuracy of the encoding is most important for signals that occur frequently. With weighted data, such a consideration is easily captured by having the weights of the points represent signal frequencies.

When applying clustering to facility allocation, such as the placement of police stations in a new district, the distribution of the stations should enable quick access to most areas in the district. However, the accessibility of different landmarks to a station may have varying importance. The weighted setting enables a convenient method for prioritising certain landmarks over others.

Traditional clustering algorithms can be readily translated into the weighted setting. This leads to the following fundamental question: Given a specific weighted clustering task, how should a user select an algorithm for that task? Recently, a new approach for choosing a clustering algorithm has been proposed (see, for example, (Ackerman, Ben-David, and Loker 2010b)). This approach involves identifying significant properties that distinguish between clustering paradigms in terms of their input/output behavior. When such properties are relevant to the user's domain knowledge, they may be used to select which algorithms are appropriate for their specific applications.

In this paper, we formulate intuitive properties that may allow a user to select an algorithm based on how it treats weighted data. Based on these properties we obtain a classification of clustering algorithms into three categories: those that are affected by weights on all data sets, those that ignore weights, and those methods that respond to weights on some configurations of the data but not on others. Among the methods that always respond to weights are several well-known algorithms, such as k-means and k-median. On the other hand, algorithms such as single-linkage, complete-linkage, and min-diameter ignore weights.

Perhaps the most notable is the last category. We find that methods belonging to that category are robust to weights when data is sufficiently clusterable, and respond to weights otherwise. Average-linkage as well as the well-known spectral objective function, ratio cut, both fall into this category. We characterize the precise conditions under which these methods are influenced by weights.

Related Work

Clustering algorithms are usually analysed in the context of unweighted data. The only related work that we are aware of is from the early 1970s. (Fisher and Ness 1971) introduced several properties of clustering algorithms. Among these, they mention "point proportion admissibility", which requires that the output of an algorithm should not change if any points are duplicated. They then observe that a few algorithms are point proportion admissible. However, clustering algorithms can display a much wider range of behaviours on weighted data than merely satisfying or failing to satisfy point proportion admissibility. We carry out the first extensive analysis of clustering on weighted data, characterising the precise conditions under which algorithms respond to weight.

In addition, (Wright 1973) proposed a formalisation of cluster analysis consisting of eleven axioms. In two of these axioms, the notion of mass is mentioned. Namely, that points with zero mass can be treated as non-existent, and that multiple points with mass at the same location are equivalent to one point with weight the sum of the masses. The idea of mass has not been developed beyond stating these axioms in their work.

Our work falls within a recent framework for clustering algorithm selection. The framework is based on identifying properties that address the input/output behaviour of algorithms. Algorithms are classified based on intuitive, user-friendly properties, and the classification can then be used to assist users in selecting a clustering algorithm for their specific application. So far, research in this framework has focused on the unweighed partitional (Ackerman, Ben-David, and Loker 2010a), (Bosagh-Zadeh and Ben-David 2009), (Ackerman, Ben-David, and Loker 2010b) and hierarchical settings (Ackerman and Ben-David 2011). This is the first application of the framework to weighted clustering.

Preliminaries

A *weight function* w over X is a function $w : X \rightarrow R^+$. Given a domain set X, denote the corresponding weighted domain by $w[X]$, thereby associating each element $x \in X$ with weight $w(x)$. A *distance function* is a symmetric function $d : X \times X \rightarrow R^+ \cup \{0\}$, such that $d(x, y) = 0$ if and only if $x = y$. We consider *weighted data sets* of the form $(w[X], d)$, where X is some finite domain set, d is a distance function over X, and w is a weight function over X.

A *k-clustering* $C = \{C_1, C_2, \ldots, C_k\}$ of a domain set X is a partition of X into $1 < k < |X|$ disjoint, non-empty subsets of X where $\cup_i C_i = X$. A *clustering* of X is a k-clustering for some $1 < k < |X|$. To avoid trivial partitions, clusterings that consist of a single cluster, or where every cluster has a unique element, are not permitted.

Denote the *weight of a cluster* $C_i \in C$ by $w(C_i) = \sum_{x \in C_i} w(x)$. For a clustering C, let $|C|$ denote the number of clusters in C. For $x, y \in X$ and clustering C of X, write $x \sim_C y$ if x and y belong to the same cluster in C and $x \not\sim_C y$, otherwise.

A *partitional weighted clustering algorithm* is a function that maps a data set $(w[X], d)$ and an integer $1 < k < |X|$ to a k-clustering of X.

A *dendrogram* $\mathcal{D}$ of X is a pair (T, M) where T is a binary rooted tree and $M : leaves(T) \rightarrow X$ is a bijection. A *hierarchical weighted clustering algorithm* is a function that maps a data set $(w[X], d)$ to a dendrogram of X. A set $C_0 \subseteq X$ is a cluster in a dendrogram $\mathcal{D} = (T, M)$ of X if there exists a node x in T so that $C_0 = \{M(y) \mid y \text{ is a leaf and a descendent of } x\}$. For a hierarchical weighted clustering algorithm $\mathcal{A}$, $\mathcal{A}(w[X], d)$ outputs a clustering $C = \{C_1, \ldots, C_k\}$ if C_i is a cluster in $\mathcal{A}(w[X], d)$ for all $1 \leq i \leq k$. A partitional algorithm $\mathcal{A}$ outputs clustering C on $(w[X], d)$ if $\mathcal{A}(w[X], d, |C|) = C$.

For the remainder of this paper, unless otherwise stated, we will use the term "clustering algorithm" for "weighted clustering algorithm".

Finally, given clustering algorithm $\mathcal{A}$ and data set (X, d), let $range(\mathcal{A}(X, d)) = \{C \mid \exists w$ such that $\mathcal{A}$ outputs C on $(w[X], d)\}$, i.e. the set of clusterings that $\mathcal{A}$ outputs on (X, d) over all possible weight functions.

Basic Categories

Different clustering algorithms respond differently to weights. We introduce a formal categorisation of clustering algorithms based on their response to weights. First, we define what it means for a partitional algorithm to be weight responsive on a clustering. We present an analogous definition for hierarchical algorithms when we study hierarchical algorithms below.

Definition 1 (Weight responsive). *A partitional clustering algorithm $\mathcal{A}$ is* weight-responsive *on a clustering C of (X, d) if*

1. *there exists a weight function w so that $\mathcal{A}(w[X], d) = C$, and*
2. *there exists a weight function w' so that $\mathcal{A}(w'[X], d) \neq C$.*

Weight-sensitive algorithms are weight-responsive on all clusterings in their range.

Definition 2 (Weight Sensitive). *An algorithm $\mathcal{A}$ is* weight-sensitive *if for all (X, d) and all $C \in range(\mathcal{A}(X, d))$, $\mathcal{A}$ is weight-responsive on C.*

At the other extreme are clustering algorithms that do not respond to weights on any data set. This is the only category that has been considered in previous work, corresponding to "point proportion admissibility"(Fisher and Ness 1971).

Definition 3 (Weight Robust). *An algorithm $\mathcal{A}$ is* weight-robust *if for all (X, d) and all clusterings C of (X, d), $\mathcal{A}$ is not weight-responsive on C.*

Finally, there are algorithms that respond to weights on some clusterings, but not on others.

Definition 4 (Weight Considering). *An algorithm $\mathcal{A}$ is* weight-considering *if*

- *There exists an (X, d) and a clustering C of (X, d) so that $\mathcal{A}$ is weight-responsive on C.*
- *There exists an (X, d) and $C \in range(\mathcal{A}(X, d))$ so that $\mathcal{A}$ is not weight-responsive on C.*

To formulate clustering algorithms in the weighted setting, we consider their behaviour on data that allows duplicates. Given a data set (X, d), elements $x, y \in X$ are duplicates if $d(x, y) = 0$ and $d(x, z) = d(y, z)$ for all $z \in X$. In a Euclidean space, duplicates correspond to elements that occur at the same location. We obtain the weighted version of a data set by de-duplicating the data, and associating every element with a weight equaling the number of duplicates of that element in the original data. The weighted version of an algorithm partitions the resulting weighted data in the same manner that the unweighted version partitions the original data. As shown throughout the paper, this translation leads to natural formulations of weighted algorithms.

Partitional Methods

In this section, we show that partitional clustering algorithms respond to weights in a variety of ways. Many popular partitional clustering paradigms, including k-means, k-median, and min-sum, are weight sensitive. It is easy to see that methods such as min-diameter and k-center are weight-robust. We begin by analysing the behaviour of a spectral objective function ratio cut, which exhibits interesting behaviour on weighted data by responding to weight unless data is highly structured.

Ratio-Cut Spectral Clustering

We investigate the behaviour of a spectral objective function, ratio-cut (Von Luxburg 2007), on weighted data. Instead of a distance function, spectral clustering relies on a *similarity function*, which maps pairs of domain elements to non-negative real numbers that represent how alike the elements are.

The ratio-cut of a clustering C is $\text{rcut}(C, w[X], s) =$

$$\frac{1}{2}\sum_{C_i \in C} \frac{\sum_{x \in C_i, y \in X \setminus C_i} s(x,y) \cdot w(x) \cdot w(y)}{\sum_{x \in C_i} w(x)}.$$

The ratio-cut clustering function is $\text{rcut}(w[X], s, k) = \arg\min_{C;|C|=k} \text{rcut}(C, w[X], s)$. We prove that this function ignores data weights only when the data satisfies a very strict notion of clusterability. To characterise precisely when ratio-cut responds to weights, we first present a few definitions.

A clustering C of $(w[X], s)$ is *perfect* if for all $x_1, x_2, x_3, x_4 \in X$ where $x_1 \sim_C x_2$ and $x_3 \not\sim_C x_4$, $s(x_1, s_2) > s(x_3, x_4)$. C is *separation-uniform* if there exists λ so that for all $x, y \in X$ where $x \not\sim_C y$, $s(x,y) = \lambda$. Note that neither condition depends on the weight function.

We show that whenever a data set has a clustering that is both perfect and separation-uniform, then ratio-cut uncovers that clustering, which implies that ratio-cut is not weight-sensitive. Note that these conditions are satisfied when all between-cluster similarities are set to zero. On the other hand, we show that ratio-cut does respond to weights when either condition fails.

Lemma 1. *Given a clustering C of (X, s) where every cluster has more than one point, if C is not separation-uniform then ratio-cut is weight-responsive on C.*

Proof. We consider two cases.

Case 1: There is a pair of clusters with different similarities between them. Then there exist $C_1, C_2 \in C$, $x \in C_1$, and $y \in C_2$ so that $s(x,y) \geq s(x,z)$ for all $z \in C_2$, and there exists $a \in C_2$ so that $s(x,y) > s(x,a)$.

Let w be a weight function such that $w(x) = W$ for some sufficiently large W and weight 1 is assigned to all other points in X. Since we can set W to be arbitrarily large, when looking at the cost of a cluster, it suffices to consider the dominant term in terms of W. We will show that we can improve the cost of C by moving a point from C_2 to C_1. Note that moving a point from C_2 to C_1 does not affect the dominant term of clusters other than C_1 and C_2. Therefore, we consider the cost of these two clusters before and after rearranging points between these clusters.

Let $A = \sum_{a \in C_2} s(x,a)$ and let $m = |C_2|$. Then the dominant term, in terms of W, of the cost of C_2 is $W\frac{A}{m}$. The cost of C_1 approaches a constant as $W \to \infty$.

Now consider clustering C' obtained from C by moving y from cluster C_2 to cluster C_1. The dominant term in the cost of C_2 becomes $W\frac{A - s(x,y)}{m-1}$, and the cost of C_1 approaches a constant as $W \to \infty$. By choice of x and y, if $\frac{A - s(x,y)}{m-1} < \frac{A}{m}$ then C' has lower loss than C when W is large enough. $\frac{A - s(x,y)}{m-1} < \frac{A}{m}$ holds when $\frac{A}{m} < s(x,y)$, and the latter holds by choice of x and y.

Case 2: The similarities between every pair of clusters are the same. However, there are clusters $C_1, C_2, C_3 \in C$, so that the similarities between C_1 and C_2 are greater than the ones between C_1 and C_3. Let a and b denote the similarities between C_1, C_2 and C_1, C_3, respectively.

Let $x \in C_1$ and w a weight function, such that $w(x) = W$ for large W, and weight 1 is assigned to all other points in X. The dominant term comes from clusters going into C_1, specifically edges that include point x. The dominant term of the contribution of cluster C_3 is Wb and the dominant term of the contribution of C_2 is Wa, totalling $Wa + Wb$.

Now consider clustering C' obtained from clustering C by merging C_1 with C_2, and splitting C_3 into two clusters (arbitrarily). The dominant term of the clustering comes from clusters other than $C_1 \cup C_2$, and the cost of clusters outside $C_1 \cup C_2 \cup C_3$ is unaffected. The dominant term of the cost of the two clusters obtained by splitting C_3 is Wb for each, for a total of $2Wb$. However, the factor of Wa that C_2 previously contributed is no longer present. This replaces the coefficient of the dominant term from $a + b$ to $2b$, which improved the cost of the clustering because $b < a$. □

Lemma 2. *Given a clustering C of (X, s) where every cluster has more than one point, if C is not perfect than ratio-cut is weight-responsive on C.*

The proof of the lemma is included in the appendix (Anonymous 2012).

Lemma 3. *Given any data set $(w[X], s)$ that has a perfect, separation-uniform k-clustering C, ratio-cut$(w[X], s, k) = C$.*

Proof. Let $(w[X], s)$ be a weighted data set, with a perfect, separation-uniform clustering $C = \{C_1, \ldots, C_k\}$. Recall that for any $Y \subseteq X$, $w(Y) = \sum_{y \in Y} w(y)$. Then,

$$\begin{aligned}
\text{rcut}(C, w[X], s) &= \frac{1}{2}\sum_{i=1}^{k} \frac{\sum_{x \in C_i}\sum_{y \in \overline{C_i}} s(x,y)w(x)w(y)}{\sum_{x \in C_i} w(x)} \\
&= \frac{1}{2}\sum_{i=1}^{k} \frac{\sum_{x \in C_i}\sum_{y \in \overline{C_i}} \lambda w(x)w(y)}{\sum_{x \in C_i} w(x)} \\
&= \frac{\lambda}{2}\sum_{i=1}^{k} \frac{\sum_{y \in \overline{C_i}} w(y)\sum_{x \in C_i} w(x)}{\sum_{x \in C_i} w(x)} = \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{y \in \overline{C_i}} w(y) \\
&= \frac{\lambda}{2}\sum_{i=1}^{k} w(\overline{C_i}) = \frac{\delta}{2}\sum_{i=1}^{k}[w(X) - w(C_i)] \\
&= \frac{\lambda}{2}\left(kw(X) - \sum_{i=1}^{k} w(C_i)\right) = \frac{\lambda}{2}(k-1)w(X).
\end{aligned}$$

Consider any other clustering, $C' = \{C'_1, \ldots, C'_k\} \neq C$. Since C is both perfect and separation-uniform, all between-cluster similarities in C equal λ, and all within-cluster similarities are greater than λ. From here it follows that all pairwise similarities in the data are at least λ. Since C is a k-clustering different from C, it must differ from C on at least one between-cluster edge, so that edge must be greater than λ.

So the cost of $C^{'}$ is,

$$\text{rcut}(C^{'}, w[X], s) = \frac{1}{2}\sum_{i=1}^{k} \frac{\sum_{x \in C_i^{'}} \sum_{y \in \overline{C_i^{'}}} s(x,y)w(x)w(y)}{\sum_{x \in C_i^{'}} w(x)}$$
$$> \frac{1}{2}\sum_{i=1}^{k} \frac{\sum_{x \in C_i^{'}} \sum_{y \in \overline{C_i^{'}}} \lambda w(x)w(y)}{\sum_{x \in C_i^{'}} w(x)}$$
$$= \frac{\lambda}{2}(k-1)w(X) = \text{rcut}(C).$$

So clustering $C^{'}$ has a higher cost than C. □

We can now characterise the precise conditions under which ratio-cut responds to weights. Ratio-cut responds to weights on all data sets but those where cluster separation is both very large and highly uniform. Formally,

Theorem 1. *Given a clustering C of (X, s) where every cluster has more than one point, ratio-cut is weight-responsive on C if and only if either C is not perfect, or C is not separation-uniform.*

Proof. The result follows by Lemmas 1, 2, and 3. □

K-Means

Many popular partitional clustering paradigms, including k-means, k-median, and min-sum, are weight sensitive. Moreover, these algorithms satisfy a stronger condition. By modifying weights, we can make these algorithms separate any set of points. We call such algorithms weight-separable.

Definition 5 (Weight Separable). *A partitional clustering algorithm $\mathcal{A}$ is* weight-separable *if for any data set (X, d) and any $S \subset X$, where $2 \leq |S| \leq k$, there exists a weight function w so that $x \not\sim_{\mathcal{A}(w[X],d,k)} y$ for all disjoint pairs $x, y \in S$.*

Note that every weight-separable algorithm is also weight-sensitive.

Lemma 4. *If a clustering algorithm $\mathcal{A}$ is weight-separable, then $\mathcal{A}$ is weight-sensitive.*

Proof. Given any $(w[X], d)$, let $C = \mathcal{A}(w[X], d, k)$. Select points x and y where $x \sim_C y$. Since $\mathcal{A}$ is weight-separable, there exists w' so that $x \not\sim_{\mathcal{A}(w'[X],d,k)} y$, and so $\mathcal{A}(w'[X], d, k) \neq C$. □

K-means is perhaps the most popular clustering objective function, with cost: $k\text{-means}(C, w[X], d) = \sum_{C_i \in C} \sum_{x \in C_i} d(x, cnt(C_i))^2$, where $cnt(C_i)$ denotes the center of mass of cluster C_i. The k-means optimizing function finds a clustering with minimal k-means cost. We show that k-means is weight-separable, and thus also weight-sensitive.

Theorem 2. *The k-means optimizing function is weight-separable.*

Proof. Consider any $S \subseteq X$. Let w be a weight function over X where $w(x) = W$ if $x \in S$, for large W, and $w(x) = 1$ otherwise. As shown by (Ostrovsky et al. 2006), the k-means objective function is equivalent to $\frac{\sum_{x,y \in C_i} d(x,y)^2 \cdot w(x) \cdot w(y)}{w(C_i)}$. Let $m_1 = \min_{x,y \in X} d(x,y)^2 > 0$, $m_2 = \max_{x,y \in X} d(x,y)^2$, and $n = |X|$. Consider any k-clustering C where all the elements in S belong to distinct clusters. Then $k\text{-means}(C, w[X], d) < km_2(n + \frac{n^2}{W})$. On the other hand, given any k-clustering C' where at least two elements of S appear in the same cluster, $k\text{-}means(C', w[X], d) \geq \frac{W^2 m_1}{W+n}$. Since $\lim_{W \to \infty} \frac{k\text{-means}(C', w[X], d)}{k\text{-means}(C, w[X], d)} = \infty$, k-means separates all the elements in S for large enough W. □

It can also be shown that the well-known min-sum objective function is also weight-separable.

Theorem 3. *Min-sum, which minimises the objective function $\sum_{C_i \in C} \sum_{x,y \in C_i} d(x,y) \cdot w(x) \cdot w(y)$, is weight-separable.*

Proof. The proof is similar to that of the previous theorem. □

Several other objective functions similar to k-means, namely k-median and k-mediods are also weight-separable. The details appear in the appendix (Anonymous 2012).

Hierarchical Algorithms

Similarly to partitional methods, hierarchical algorithms also exhibit a wide range of responses to weights. We show that Ward's method, a successful linkage-based algorithm, as well as popular divisive heirarchical methods, are weight sensitive. On the other hand, it is easy to see that the linkage-based algorithms single-linkage and complete-linkage are both weight robust, as was observed in (Fisher and Ness 1971).

Average-linkage, another popular linkage-based method, exhibits more nuanced behaviour on weighted data. When a clustering satisfies a reasonable notion of clusterability, then average-linkage detects that clustering irrespective of weights. On the other hand, this algorithm responds to weights on all other clusterings. We note that the notion of clusterability required for average-linkage is a lot weaker than the notion of clusterability used to characterise the behaviour of ratio-cut on weighted data.

Hierarchical algorithms output dendrograms, which contain multiple clusterings. Please see the preliminary section for definitions relating to the hierarchical setting. Weight-responsive for hierarchical algorithms is defined analogously to Definition 1.

Definition 6 (Weight responsive). *A clustering algorithm $\mathcal{A}$ is* weight-responsive *on a clustering C of (X, d) if (1) there exists a weight function w so that $\mathcal{A}(w[X], d)$ outputs C, and (2) there exists a weight function w' so that $\mathcal{A}(w'[X], d)$ does not output C.*

Weight-sensitive, weight-considering, and weight-robust are defined as in the preliminaries section, with the above definition for weight-responsive.

Average Linkage

Linkage-based algorithms start by placing each element in its own cluster, and proceed by repeatedly merging the "closest" pair of clusters until the entire dendrogram is constructed. To identify the closest clusters, these algorithms use a linkage function that maps pairs of clusters to a real number. Formally, a *linkage function* is a function $\ell : \{(X_1, X_2, d, w) \mid d, w \text{ over } X_1 \cup X_2\} \rightarrow R^+$.

Average-linkage is one of the most popular linkage-based algorithms (commonly applied in bioinformatics under the name UPGMA). Recall that $w(X) = \sum_{x \in X} w(x)$. The average-linkage linkage function is

$$\ell_{AL}(X_1, X_2, d, w) = \frac{\sum_{x \in X_1, y \in X_2} d(x, y) \cdot w(x) \cdot w(y)}{w(X_1) \cdot w(X_2)}.$$

To study how average-linkage responds to weights, we give a relaxation of the notion of a perfect clustering.

Definition 7 (Nice). *A clustering C of $(w[X], d)$ is* nice *if for all $x_1, x_2, x_3 \in X$ where $x_1 \sim_C x_2$ and $x_1 \not\sim_C x_3$, $d(x_1, x_2) < d(x_1, x_3)$.*

Data sets with nice clusterings correspond to those that satisfy the "strict separation" property introduced by Balcan *et al.* (Balcan, Blum, and Vempala 2008). As for a perfect clustering, being a nice clustering is independent of weights.

We present a complete characterisation of the way that average-linkage (AL) responds to weights, showing that it ignores weights on nice clusterings, but responds to weights on all other clusterings.

Theorem 4. *For any data set (X, d) and clustering $C \in range(AL(X, d))$, average-linkage is weight robust on clustering C if and only if C is a nice clustering.*

Theorem 4 follows from the two lemmas below.

Lemma 5. *If a clustering $C = \{C_1, \ldots, C_k\}$ of (X, d) is not nice, then either $C \notin range(AL(X, d))$ or average-linkage is weight-responsive on C.*

Proof. Assume that there exists some w so that $C \in AL(w[X], d)$. If it does not exist then we are done. We construct w' so that $C \notin AL(w'[X], d)$.

Since C is not nice, there exist $1 \leq i, j \leq k$, $i \neq j$, and $x_1, x_2 \in C_i$, $x_1 \neq x_2$, and $x_3 \in C_j$, so that $d(x_1, x_2) > d(x_1, x_3)$.

Now, define weigh function w' as follows: $w'(x) = 1$ for all $x \in X \setminus \{x_1, x_2, x_3\}$, and $w'(x_1) = w'(x_2) = w'(x_3) = W$, for some large value W. We argue that when W is sufficiently large, C is not a clustering in $AL(w'[X], d)$.

By way of contradiction, assume that C is a clustering in $AL(w'[X], d)$ for any setting of W. Then there is a step in the algorithm where clusters X_1 and X_2 merge, where $X_1, X_2 \subset C_i$, $x_1 \in X_1$, and $x_2 \in X_2$. At this point, there is some cluster $X_3 \subseteq C_j$ so that $x_3 \in X_3$.

We compare $\ell_{AL}(X_1, X_2, d, w')$ and $\ell_{AL}(X_1, X_3, d, w')$. $\ell_{AL}(X_1, X_2, d, w') = \frac{W^2 d(x_1,x_2) + \alpha_1 W + \alpha_2}{W^2 + \alpha_3 W + \alpha_4}$, for some non-negative real α_is. Similarly, $\ell_{AL}(X_1, X_3, d, w') = \frac{W^2 d(x_1,x_3) + \beta_1 W + \beta_2}{W^2 + \beta_3 W + \beta_4}$ for some non-negative real β_is.

Dividing W^2, we see that $\ell_{AL}(X_1, X_3, d, w') \rightarrow d(x_1, x_3)$ and $\ell_{AL}(X_1, X_2, d, w') \rightarrow d(x_1, x_2)$ as $W \rightarrow \infty$, and so the result holds since $d(x_1, x_3) < d(x_1, x_2)$. Therefore average linkage merges X_1 with X_3, so cluster C_i is never formed, and so C is not a clustering in $AL(w'[X], d)$. □

Finally, average-linkage outputs all nice clusterings present in a data set, regardless of weights.

Lemma 6. *Given any weighted data set $(w[X], d)$, if C is a nice clustering of (X, d), then C is in the dendrogram produced by average-linkage on $(w[X], d)$.*

Proof. Consider a nice clustering $C = \{C_1, \ldots, C_k\}$ over $(w[X], d)$. It suffices to show that for any $1 \leq i < j \leq k$, $X_1, X_2 \subseteq C_i$ where $X_1 \cap X_2 = \emptyset$ and $X_3 \subseteq C_j$, $\ell_{AL}(X_1, X_2, d, w) < \ell_{AL}(X_1, X_3, d, w)$.

It can be show that $\ell_{AL}(X_1, X_2, d, w) \leq \frac{\sum_{x_1 \in X_1} w(x_1) \cdot \max_{x_2 \in X_2} d(x_1, x_2)}{w(X_1)}$ and $\ell_{AL}(X_1, X_3, d, w) \geq \frac{\sum_{x_1 \in X_1} w(x_1) \cdot \min_{x_3 \in X_3} d(x_1, x_3)}{w(X_1)}$.

Since C is nice, $\min_{x_3 \in X_3} d(x_1, x_3) > \max_{x_2 \in X_2} d(x_1, x_2)$, thus $\ell_{AL}(X_1, X_3) > \ell_{AL}(X_1, X_2)$. □

Ward's Method

Ward's method is a highly effective clustering algorithm (Everitt 1993), which, at every step, merges the clusters that will yield the minimal increase to the k-means cost. Let $ctr(X, d, w)$ be the center of mass of the data set $(w[X], d)$. Then, the linkage function for Ward's method is $\ell_{Ward}(X_1, X_2, d, w) = \frac{w(X_1) \cdot w(X_2) \cdot d(ctr(X_1, d, w), ctr(X_2, d, w))^2}{w(X_1) + w(X_2)}$.

Theorem 5. *Ward's method is weight sensitive.*

The proof is included in the appendix (Anonymous 2012).

Divisive Algorithms

The class of divisive clustering algorithms is a well-known family of hierarchical algorithms, which construct the dendrogram by using a top-down approach. This family of algorithms includes the popular bisecting k-means algorithm. We show that a class of algorithms that includes bisecting k-means consists of weight-sensitive methods.

Given a node x in dendrogram (T, M), let $\mathcal{C}(x)$ denote the cluster represented by node x. Formally, $\mathcal{C}(x) = \{M(y) \mid y \text{ is a leaf and a descendent of } x\}$. Informally, a $\mathcal{P}$-Divisive algorithm is a hierarchical clustering algorithm that uses a partitional clustering algorithm $\mathcal{P}$ to recursively divide the data set into two clusters until only single elements remain. Formally,

Definition 8 ($\mathcal{P}$-Divisive). *A hierarchical clustering algorithm $\mathcal{A}$ is* $\mathcal{P}$-Divisive *with respect to a partitional clustering algorithm $\mathcal{P}$, if for all (X, d), we have $\mathcal{A}(w[X], d) = (T, M)$, such that for all non-leaf nodes x in T with children x_1 and x_2, $\mathcal{P}(w[\mathcal{C}(x)], d, 2) = \{\mathcal{C}(x_1), \mathcal{C}(x_2)\}$.*

	Partitional	Hierarchical
Weight Sensitive	k-means, k-medoids k-median, Min-sum	Ward's method Bisecting k-means
Weight Considering	Ratio-cut	Average-linkage
Weight Robust	Min-diameter k-center	Single-linkage Complete-linkage

Table 1: Classification of weighted clustering algorithms.

We obtain bisecting k-means by setting $\mathcal{P}$ to k-means. Other natural choices for $\mathcal{P}$ include min-sum, and exemplar-based algorithms such as k-median. As shown above, many of these partitional algorithms are weight-separable. We show that whenever $\mathcal{P}$ is weight-separable, then $\mathcal{P}$-Divisive is weight-sensitive. The proof of the next theorem appears in the appendix (Anonymous 2012).

Theorem 6. *If $\mathcal{P}$ is weight-separable then the $\mathcal{P}$-Divisive algorithm is weight-sensitive.*

Conclusions

We study the behaviour of clustering algorithms on weighted data, presenting three fundamental categories that describe how such algorithms respond to weights and classifying several well-known algorithms according to these categories. Our results are summarized in Table 1. We note that all of our results immediately translate to the standard setting, by mapping each point with integer weight to the same number of unweighted duplicates.

Our results can be used to aid in the selection of a clustering algorithm. For example, in the facility allocation application discussed in the introduction, where weights are of primal importance, a weight-sensitive algorithm is suitable. Other applications may call for weight-considering algorithms. This can occur when weights (i.e. number of duplicates) should not be ignored, yet it is still desirable to identify rare instances that constitute small but well-formed outlier clusters. For example, this applies to patient data on potential causes of a disease, where it is crucial to investigate rare instances. While we do not argue that these considerations are always sufficient, they can provide valuable guidelines when clustering data that is weighted or contains element duplicates.

Our analysis also reveals the following interesting phenomenon: algorithms that are known to perform well in practice (in the classical, unweighted setting), tend to be more responsive to weights. For example, k-means is highly responsive to weights while single linkage, which often performs poorly in practice (Hartigan 1981), is weight robust.

We also study several k-means heuristics, specifically the Lloyd algorithm with several methods of initialization and the PAM algorithm. These results were omitted due to a lack of space, but they are included in the appendix (Anonymous 2012). Our analysis of these heuristics lends further support to the hypothesis that the more commonly applied algorithms are also more responsive to weights.

Acknowledgements

This work supported in part by the Sino-Danish Center for the Theory of Interactive Computation, funded by the Danish National Research Foundation and the National Science Foundation of China (under the grant 61061130540). The authors acknowledge support from the Center for research in the Foundations of Electronic Markets (CFEM), supported by the Danish Strategic Research Council. This work was also supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Alexander Graham Bell Canada Graduate Scholarship.

References

Ackerman, M., and Ben-David, S. 2011. Discerning linkage-based algorithms among hierarchical clustering methods. In *IJCAI*.

Ackerman, M.; Ben-David, S.; and Loker, D. 2010a. Characterization of linkage-based clustering. In *COLT*.

Ackerman, M.; Ben-David, S.; and Loker, D. 2010b. Towards property-based classification of clustering paradigms. In *NIPS*.

Agarwal, P. K., and Procopiuc, C. M. 1998. Exact and approximation algorithms for clustering. In *SODA*.

Anonymous. 2012. Weighted Clustering Appendix. http://wikisend.com/download/573754/clustering_appendix.pdf.

Arthur, D., and Vassilvitskii, S. 2007. K-means++: The advantages of careful seeding. In *SODA*.

Balcan, M. F.; Blum, A.; and Vempala, S. 2008. A discriminative framework for clustering via similarity functions. In *STOC*.

Bosagh-Zadeh, R., and Ben-David, S. 2009. A uniqueness theorem for clustering. In *UAI*.

Dasgupta, S., and Long, P. M. 2005. Performance guarantees for hierarchical clustering. *J. Comput. Syst. Sci.* 70(4):555–569.

Everitt, B. S. 1993. *Cluster Analysis*. John Wiley & Sons Inc.

Fisher, L., and Ness, J. V. 1971. Admissible clustering procedures. *Biometrika* 58:91–104.

Hartigan, J. 1981. Consistency of single linkage for high-density clusters. *J. Amer. Statist. Assoc.* 76(374):388–394.

Jain, A. K.; Murty, M. N.; and Flynn, P. J. 1999. Data clustering: a review. *ACM Comput. Surv.* 31(3):264–323.

Kaufman, L., and Rousseeuw, P. J. 2008. *Partitioning Around Medoids (Program PAM)*. John Wiley & Sons, Inc. 68–125.

Ostrovsky, R.; Rabani, Y.; Schulman, L. J.; and Swamy, C. 2006. The effectiveness of Lloyd-type methods for the k-means problem. In *FOCS*.

Talagrand, M. 1996. A new look at independence. *Ann. Probab.* 24(1):1–34.

Vapnik, V. 1998. *Statistical Learning Theory*. New York: Wiley.

Von Luxburg, U. 2007. A tutorial on spectral clustering. *J. Stat. Comput.* 17(4):395–416.

Wright, W. E. 1973. A formalization of cluster analysis. *J. Pattern Recogn.* 5(3):273–282.