Best of Both: A Hybridized Centroid-Medoid Clustering Heuristic

Nizar Grira GRIRA@NII.AC.JP
Michael E.Houle MEH@NII.AC.JP
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

Abstract

Although each iteration of the popular k-MEANS clustering heuristic scales well to larger problem sizes, it often requires an unacceptably-high number of iterations to converge to a solution. This paper introduces an enhancement of k-MEANS in which local search is used to accelerate convergence without greatly increasing the average computational cost of the iterations. The local search involves a carefully-controlled number of swap operations resembling those of the more robust k-MEDOIDS clustering heuristic. We show empirically that the proposed method improves convergence results when compared to standard k-MEANS.

1. Introduction

The data clustering problem arises in a wide variety of fields, including data mining, pattern recognition, computer vision, and bioinformatics. In general, methods for clustering aim to organize a collection of data items into groups (the clusters), so as to obtain the highest possible association among items sharing the same group, and the greatest possible differentiation between items from different groups. Often, but not always, the groupings sought are disjoint, with each item assigned to exactly one cluster. The degree of association is typically expressed in terms of a similarity measure between pairs of items, chosen according to the purpose of the application, to domain-specific assumptions and to prior knowledge of the problem. Unlike classification, which requires a learning phase with respect to a user-supplied training set, clustering is regarded as a form of *unsupervised learning*, and is usually performed when little or no information is available concerning the membership of data items to predefined classes. The uses of clustering are extensive, and are supported by detailed surveys in the literature of many research communities (Jain & Dubes, 1988; Duda et al., 2001; Fukunaga, 1990).

Of the various clustering methods proposed to date, most can be categorized into one of two main styles: partitional or hierarchical (agglomerative). Hierarchical methods typically form clusterings in a bottom-up manner, building up larger groups by progressively merging smaller groups according to some local similarity-based linkage criterion. Generally speaking, hierarchical algorithms are static in the sense that the assignment of items to clusters cannot be revisited and refined, with subsequent membership changes arising solely from the merge process. The hierarchical approach is a particularly popular choice for applications involving geographic data, and in other low-dimensional spatial settings where it is meaningful to unify two aggregations of data if a substantial portion of both are in close proximity to one another. In more general settings, however, commonly-used merge criteria can result in clusters of very poor quality, whose elements are related only by long chains of association.

Top-down approaches to clustering are also possible. Instead of composing groups from subgroups sharing common characteristics, the partitional clustering style can be viewed as one of splitting a heterogenous data set into smaller, more homogeneous subgroups. The clustering typically proceeds in an iterative fashion, with items continually being reassigned to groups in an attempt to optimize some global quality criterion. In general, only an approximate solution can be obtained, since the number of possible partitions of the data grows super-exponentially with the number of items. In essence, the membership reassignment process constitutes a local search heuristic over the space of all possible data partitions.

The most famous example of a partitional clustering algorithm, widely used in practice, is the k-MEANS heuristic (McQueen, 1967). k-MEANS produces a partition of the data set by assigning each item to a pro-

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

totype (representative point) within the data domain. For each cluster, prototypes are contructed by computing the *centroid* (also referred to as the *mean* or the *center of mass*) of its members. Items are then reassigned to centroids so as to minimize a global statistical homogeneity criterion over the set of clusters, typically the sum of squared Euclidean distances from each item to its representative. Each iteration (partition followed by reassignment) can be performed in time linear in the number of data items. k-MEANS can be viewed as an Expectation Maximization (EM) heuristic for the case where the data is assumed to follow a mixture of gaussian distributions (Dempster et al., 1977). The popularity of k-MEANS in practice is largely due to its ease of implementation, and its relative efficiency. The importance of k-MEANS heuristics has been shown through extensive experimentation conducted over the past three decades (Dubes & Jain, 1976; Chen et al., 2004). However, as we shall later see, k-MEANS has a number of serious deficiencies that limit its effectiveness, despite its advantages in speed.

A close variant of k-MEANS is the k-MEDOIDS heuristic, which adopts the restriction that all cluster representatives coincide with items of the data set (Kaufman & Rousseeuw, 1990). The use of medoids (items of the data set) as representives rather than centroids (points in the data domain) results in robustness in the statistical sense, in that medoid solutions are less sensitive to small changes in the composition of the data set — and by extension, less sensitive to noise and outliers. They also have the advantage of interprebility, in that each cluster representative is an instantiated example drawn from its membership. However, despite these advantages, medoid-based heuristics have not gained wide acceptance in practice due to their prohibitive cost, as a quadratic number of distance calculations are generally required for convergence.

The remainder of this paper is organized as follows. We first begin by formally describing two variant k-MEANS heuristics, and then prove that one of these, *local-search* k-MEANS, provides a constant-factor approximation of the cost of the optimal solution. We continue by introducing a hybrid centroid-medoid clustering heuristic that combines the best features of both k-MEANS and k-MEDOIDS. In Section 5, we present the results of experiments showing the effectiveness of our proposed method. Concluding remarks appear in Section 6.

2. Background

Given a set $X \subset \mathbb{R}^d$ consisting of n data items and integer $k \in \mathbb{N}_{\geq 2}$, we aim to find a partition $C = \{C_1, C_2, \ldots, C_k\}$ of k disjoint non-empty clusters minimizing the squared-error distortion cost function:

$$\Psi(C) = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mu(C_i)\|^2, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean distance and $\mu(C_i) = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$ is the center of mass of the subset C_i. The k-means problem as stated above is NP-hard — no known algorithm is guaranteed to find the optimum in polynomial time, and thus a heuristic approach is needed. However, it is known that the optimal partition is a centroidal Voronoi tessellation (CVT) on the set of sites $\{\mu(C_i) \mid 1 \leq i \leq k\}$, with the items of C_i all lying in the Voronoi cell of $\mu(C_i)$ (Du et al., 1999).

2.1. Batch k-MEANS heuristic

The batch k-MEANS method alternates between a *maximization* step in which items are assigned to their closest prototypes, and an *expectation* step where the prototypes are replaced by the centroids of the items that have been assigned to them. These two steps form what we call a single *run*, and are iterated until a convergence criterion is met — typically, until the run produces no significant change in the value of the cost function.

Algorithm 1 Batch k-MEANS heuristic

Initialization: Randomly generate a set $P = \{p_i \mid 1 \leq i \leq k\}$ of k initial prototype points.
repeat
  Step 1 (maximization): Assign the data items to their closest prototypes to form the clusters $C_i = \{x \in X \mid \forall j < i, \|x - p_i\| < \|x - p_j\|$ and $\forall j \geq i, \|x - p_i\| \leq \|x - p_j\|\}$.
  Step 2 (expectation): Replace the prototype p_i by the center of mass of its cluster, $\mu(C_i)$.
until convergence

The batch k-means algorithm has been widely adopted due to its simplicity and ease of implementation. Moreover, its time complexity of $\mathcal{O}(tknd)$, where t is the number of iterations required for convergence, is often taken to be linear in the size of the input when t, k, and d are all significantly smaller than n. However, this assumption is not always correct, especially in regard to the number of iterations t.

2.2. Local search k-MEANS

The local search technique employed by the k-MEDOIDS heuristic can also be applied to the case where the prototypes are not necessarily members of the data set, leading us to a local-search variant of k-MEANS.

Let $\mathcal{F} \subset \mathbb{R}^d$ be a set of prototype candidates, with $|\mathcal{F}| > k$. The local search k-MEANS heuristic attempts to find the set of prototypes $\mathcal{F}^* \subset \mathcal{F}$ of size $|\mathcal{F}^*| = k$ that collectively minimize the cost function. The algorithm proceeds by iteratively swapping candidates into a set of tentative prototypes. The procedure can be described in terms of:

- a set $\mathcal{FS}$ of all subsets consisting of exactly k prototypes of $\mathcal{F}$;
- a distortion cost function $\Pi : \mathcal{FS} \to \mathbb{R}$;
- a swap neighborhood structure $\mathcal{N} : \mathcal{FS} \to 2^{\mathcal{FS}}$, where for any $S \in \mathcal{FS}$, its swap neighborhood $\mathcal{N}(S)$ consists of all sets $S' \in \mathcal{FS}$ differing from S by a single prototype — that is, such that $|S' \cap S| = k - 1$.

We say that a solution S of $\mathcal{FS}$ is *locally optimal* if $\Pi(S) \leq \Pi(S')$ for any subset $S' \in \mathcal{N}(S)$.

Algorithm 2 Local search k-MEANS heuristic

```
Initialize S to hold k prototypes arbitrarily chosen from the set F.
while ∃ S' ∈ N(S) such that Π(S') < Π(S), do
    Arbitrarily choose some candidate set S* ∈ N(S) satisfying Π(S*) < Π(S).
    S ← S*.
end while
```

Clearly, the performance of local-search k-MEANS strongly depends on the choice of the set of candidate centers $\mathcal{F}$. Moreover, even though the local-search variant can immediately take advantage of an improving swap, the total computation time involved in evaluating swaps is still significant. Like k-MEDOIDS, the local-search variant of k-MEANS also suffers from a convergence time that is quadratic in the number of items of the data set. However, it turns out that the cost of the solution obtained never exceeds that of the optimal solution by more than a constant factor.

3. Analysis of local search k-MEANS

In this section, we prove the following theorem on the quality of the solutions found by local search k-MEANS. The initial stages of this proof resemble the analysis of the k-MEDOIDS cost due to (Arya et al., 2001).

Theorem 1. *Let $S \in \mathcal{FS}$ be a locally optimum solution for local search k-MEANS under the least squared error criterion of Equation (1). Let O denote the optimal solution. Then $\Psi(S) \leq 25\Psi(O)$.*

From the definition of local optimality, we immediately observe that

$$\Psi(S \cup \{o\} \setminus \{s\}) \geq \Psi(S) \quad \forall s \in S, o \in O \qquad (2)$$

Given a prototype $s \in S$, the set of data items contained in the Voronoi cell of s in the CVT of S will be denoted by $\mathcal{V}(s)$. Similarly, for $o \in O$, the items in the Voronoi cell of o in the CVT of O will be denoted by $\mathcal{V}_{\text{opt}}(o)$.

As in (Arya et al., 2001), we consider that a locally-optimal center $s \in S$ *captures* an optimal center $o \in O$ if $|\mathcal{V}(s) \cap \mathcal{V}_{\text{opt}}(o)| > \frac{1}{2}|\mathcal{V}_{\text{opt}}(o)|$; that is, if $\mathcal{V}(s)$ contains more than half the data items belonging to $\mathcal{V}_{\text{opt}}(o)$. It follows that a locally-optimal center can capture any number $\lambda \in \{1, \ldots, k\}$ of optimal centers, whereas an optimal center can be captured by at most one locally-optimal center.

For the analysis, we will consider the effect on the cost if a locally-optimal center $s \in S$ were to be swapped with an optimal center $o \in O$. Let us consider the set E consisting of all possible candidate swap pairs of the form (s, o), with $s \in S$ and $o \in O$. For the analysis, we will restrict our attention to any maximal subset $E' \subseteq E$ satisfying the following conditions:

- If s captures exactly one optimal center $o \in O$, then $(s, o) \in E'$.
- If s captures more than one optimal center, then s does not contribute to any pairs in E'.
- If s captures no optimal centers, then s contributes to at most two pairs in E'.

Since E' is assumed to be maximal subject to these conditions, it is not difficult to verify that the following also hold:

1. Each optimal center o must be paired with exactly one locally-optimal center s.
2. Each locally-optimal center s can be paired with at most two optimal centers.
3. For every pair $(s, o) \in E'$, s does not capture any other optimal center $o' \neq o$.

By simply rewriting Inequality (2), we can obtain an upper bound on the the cost change after the swap $S \leftarrow S \cup \{o\} \setminus \{s\}$, for any arbitrary choice of $s \in S$ and $o \in O$:

$$\Psi(S \cup \{o\} \setminus \{s\}) - \Psi(S) \geq 0.$$

For any center $c \in S \cup \{o\} \setminus \{s\}$, let $\mathcal{V}_{s\to o}(c)$ denote set of data items contained in the Voronoi cell of c with respect to the CVT of $S \cup \{o\} \setminus \{s\}$. After the swap, the points assigned to o will simply be those that are contained in the new cell $\mathcal{V}_{s\to o}(o)$. The cost will change by the amount

$$\sum_{x \in \mathcal{V}_{s\to o}(o)} (\|x - o\|^2 - \|x - s_x\|^2),$$

where s_x denotes the closest locally-optimal center to x in S. The items lying in $\mathcal{V}(s) \setminus \mathcal{V}_{s\to o}(o)$ will be reassigned to other centers as a result of the swap.

Let o_x denote the closest optimal center to $x \in \mathcal{V}(s) \setminus \mathcal{V}_{s\to o}(o)$.

- If $o_x = o$, then x will be reassigned to some other locally-optimal center $s'_x \in S$. Since $x \notin \mathcal{V}_{s\to o}(o)$, we have $\|x - s'_x\| \leq \|x - o_x\|$.
- Otherwise, if $o_x \neq o$, then due to the third condition stated above, we know that s does not capture o_x. Therefore, x will be assigned to another locally-optimal center $s'_x \in S$.

We now analyze the impact of swaps on the cost. From this point onward, the proof strategy diverges from that of the k-MEDOIDS analysis of (Arya et al., 2001).

If we denote by s_{o_x} the closest locally-optimal center to o_x in $S \cup \{o\} \setminus \{s\}$, then the cost change relative to the reassignment of point $x \in \mathcal{V}(s) \setminus \mathcal{V}_{s\to o}(o)$ is bounded by $\|x - s_{o_x}\|$. That is,

$$\|x - s'_x\| \leq \|x - s_{o_x}\| \ \ \forall\ x \in \mathcal{V}(s) \setminus \mathcal{V}_{s\to o}(o).$$

The cost change is thus

$$\begin{aligned}\Lambda &= \sum_{x \in \mathcal{V}(s) \setminus \mathcal{V}_{s\to o}(o)} (\|x - s'_x\|^2 - \|x - s\|^2) \\ &\leq \sum_{x \in \mathcal{V}(s) \setminus \mathcal{V}_{s\to o}(o)} (\|x - s_{o_x}\|^2 - \|x - s\|^2). \quad (3)\end{aligned}$$

Hence, the total cost change for a single swap is:

$$\begin{aligned}&\Psi(S \cup \{o\} \setminus \{s\}) - \Psi(S) \\ &= \sum_{x \in \mathcal{V}_{s\to o}(o)} (\|x - o\|^2 - \|x - s_x\|^2) \\ &\quad + \sum_{x \in \mathcal{V}(s) \setminus \mathcal{V}_{s\to o}(o)} (\|x - s'_x\|^2 - \|x - s\|^2) \geq 0\end{aligned}$$

Using Inequality (3), we obtain:

$$\begin{aligned}0 \ \leq\ & \sum_{x \in \mathcal{V}_{s\to o}(o)} (\|x - o\|^2 - \|x - s_x\|^2) \quad (4) \\ &+ \sum_{x \in \mathcal{V}(s) \setminus \mathcal{V}_{s\to o}(o)} (\|x - s_{o_x}\|^2 - \|x - s\|^2)\end{aligned}$$

Since s is the closest locally-optimal center to x in S, we can further expand the second term on the right of the inequality by adding the positive quantities $\|x - s_{o_x}\|^2 - \|x - s\|^2$ for all $x \in \mathcal{V}(s) \cap \mathcal{V}_{s\to o}(o)$. Thus, Inequality (4) implies that

$$\begin{aligned}0 \ \leq\ & \sum_{x \in \mathcal{V}_{s\to o}(o)} (\|x - o\|^2 - \|x - s_x\|^2) \\ &+ \sum_{x \in \mathcal{V}(s)} (\|x - s_{o_x}\|^2 - \|x - s\|^2)\end{aligned}$$

Next, we shall derive a bound on the total cost change over all pairs in E'. Recall the conditions that apply to these pairs — namely, that each locally-optimal center contributes to at most two pairs, and that each optimal center appears in exactly one pair. We therefore obtain:

$$\begin{aligned}0 \ \leq\ & \sum_{(s,o) \in E'} \sum_{x \in \mathcal{V}_{s\to o}(o)} (\|x - o\|^2 - \|x - s_x\|^2) \\ &+ \sum_{x \in \mathcal{V}(s)} (\|x - s_{o_x}\|^2 - \|x - s\|^2) \\ \leq\ & \sum_{x \in X} (\|x - o_x\|^2 - \|x - s_x\|^2) \\ &+ 2 \sum_{x \in X} (\|x - s_{o_x}\|^2 - \|x - s_x\|^2) \\ \leq\ & \Psi(O) - \ 3\ \Psi(S)\ + 2 \sum_{x \in X} \|x - s_{o_x}\|^2. \quad (5)\end{aligned}$$

This leaves us with a bound independent of the actual choice of E'. To bound $\sum_{x \in X} \|x - s_{o_x}\|^2$, we require the following technical lemma:

Lemma 1. *Let I be any set of items, and let μ be its centroid. Then the sum of squared Euclidean distances from the items of I to any point μ' is at most $\sum_{x \in I} \|x - \mu\|^2 + |I| \cdot \|\mu' - \mu\|^2$.*

Proof. Omitted in this version. □

Using this lemma, we obtain:

$$\begin{aligned}&\sum_{x \in X} \|x - s_{o_x)}\|^2 \\ &\leq \sum_{o \in O} \left(\sum_{x \in \mathcal{V}_{\text{opt}}(o)} \|x - o\|^2 + |\mathcal{V}_{\text{opt}}(o)| \cdot \|o - s_{o_x}\|^2 \right) \\ &= \sum_{o \in O} \sum_{x \in \mathcal{V}_{\text{opt}}(o)} \left(\|x - o\|^2 + \|o - s_{o_x}\|^2 \right).\end{aligned}$$

Observing that $\forall x \in X$, $\|o - s_{o_x}\| \leq \|o - s_x\|$, it follows that:

$$\begin{aligned}
&\sum_{x \in X} \|x - s_{o_x}\|^2 \\
&\quad \leq \sum_{x \in X} \left(\|x - o_x\|^2 + \|o_x - s_x\|^2 \right) \\
&\quad \leq \Psi(O) + \sum_{x \in X} (\|x - o_x\| + \|x - s_x\|)^2 \\
&\quad = 2\Psi(O) + \Psi(S) + 2 \sum_{x \in X} \|x - o_x\| \cdot \|x - s_x\|.
\end{aligned}$$

Using the Cauchy-Schwartz Inequality, we then obtain:

$$\begin{aligned}
&\sum_{x \in X} \|x - o_x\| \cdot \|x - s_x\| \\
&\quad \leq \left(\sum_{x \in X} \|x - o_x\|^2 \right)^{\frac{1}{2}} \left(\sum_{x \in X} \|x - s_x\|^2 \right)^{\frac{1}{2}}
\end{aligned}$$

from which it follows that

$$\sum_{x \in X} \|x - s_{o_x}\|^2 \leq 2\Psi(O) + \Psi(S) + 2\Psi(O)^{\frac{1}{2}} \Psi(S)^{\frac{1}{2}}.$$

Combining the above with Inequality (5):

$$\begin{aligned}
0 \quad &\leq \quad \Psi(O) - 3\,\Psi(S) \\
&\qquad +2(2\Psi(O) + \Psi(S) + 2\Psi(O)^{\frac{1}{2}} \Psi(S)^{\frac{1}{2}}) \\
&\leq \quad \left(5\,\Psi(O)^{\frac{1}{2}} - \Psi(S)^{\frac{1}{2}}\right) \left(\Psi(O)^{\frac{1}{2}} + \Psi(S)^{\frac{1}{2}}\right).
\end{aligned}$$

Both factors must be non-negative, leading to the following bound, as required:

$$\Psi(S) \leq 25\,\Psi(O).$$

The theorem implies that the local search k-Means heuristic is a $(25 + \epsilon)$-approximation of the optimal k-Means solution, the $\epsilon > 0$ term arising from the choice of centers from among a discrete set of candidates $\mathcal{F}$ rather than from the entire space. For a sufficiently dense choice of $\mathcal{F}$, the value of ϵ can be driven arbitrarily close to zero.

4. Hybrid centroid-medoid algorithm

In this section, we present the details of our proposed hybrid centroid-medoid clustering heuristic. Intuitively speaking, it seeks to boost the performance of batch k-Means by occasionally considering local-search swap operations, so as to avoid being trapped at solutions in which some clusters are overrepresented by prototypes while others are unrepresented. The hybrid algorithm, shown as Algorithm 3, starts by running batch k-Means with a randomly-chosen initial

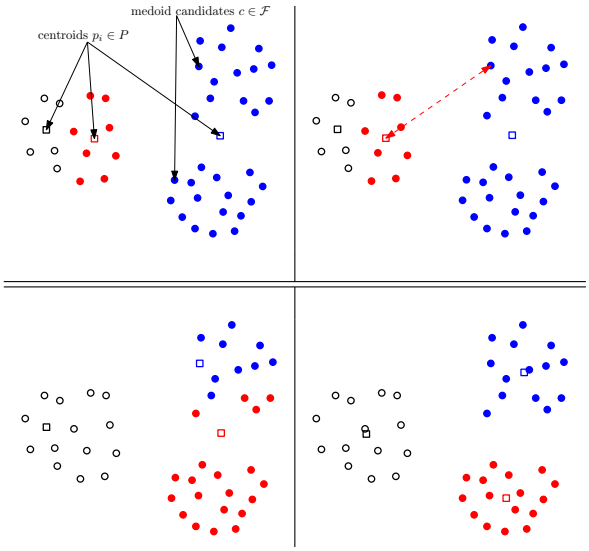

Figure 1. Illustration of the hybrid centroid-medoid method. Upper left: after convergence of the centroid phase; upper right: an improving medoid swap; lower left: the medoid is accepted as a new prototype; lower right: the result of the next centroid phase.

set of k prototype centers. After a predetermined number of runs $r \geq 1$, we obtain a set of k centroids P. Next, we attempt to swap a single candidate centroid of P with medoids selected from a set of candidates $\mathcal{M} \subseteq X$ of size sk, for some integer $s \geq 1$ — the precise strategy for generating $\mathcal{M}$ will be described later. Then, if a swap is discovered that would lead to an improvement in the cost value, the swap is immediately performed. The algorithm would then reiterate the process, alternating between r runs of batch k-Means (the 'centroid' phase) and the search for a centroid-medoid swap (the 'medoid' phase).

The trade-off between the improved convergence rate and iteration time efficiency is governed by the choice of parameters r and s. Choosing large values of r causes the method to emulate batch k-Means, whereas choosing a large value of s increases the time cost of the medoid phase. Although it would seem better to consider many candidate centroids for swapping in the medoid phase, and to consider all items of X as candidate medoids, this would lead to a prohibitively high cost for this phase. Instead, we limit the cost of a medoid phase to be of the same order as that of a centroid phase, by carefully targeting a single centroid most likely to lead to an improving swap, and using sampling techniques to determine a small yet representative collection of candidate medoids.

The candidate centroid for swapping is determined by computing a closest pair among the current set of cen-

troids, and then selecting one of the pair arbitrarily. The rationale for this choice is illustrated in Figure 1. Provided that k, the number of clusters sought, is at least as large as the 'true' number of clusters of data set X, any failure to discover a particular cluster would imply that some other cluster has more than one representative assigned to it. Especially in high-dimensional contexts, this implies that the distance between these two centroids is likely to be substantially lower than the average inter-prototype distance. Hence, if the candidate medoid set $\mathcal{M}$ contains at least one item of an undiscovered cluster, relocating one of the centroids to a medoid from the undiscovered cluster should result in a decrease in the cost. Further iterations of batch k-MEANS would then be required so as to readjust the positions of the representatives within the 'donor' cluster and the 'recipient' cluster.

Algorithm 3 Hybrid centroid-medoid heuristic

```
Initialization: randomly generate a set P = {p_i | 1 ≤ i ≤ k} of k initial prototype points.
repeat
  Update P by performing r runs of the batch k-MEANS heuristic.
  Determine a closest pair (p, q) of prototypes; that is, prototypes p, q ∈ P such that p ≠ q and ‖p − q‖ ≤ ‖p′ − q′‖ for any choice of p′, q′ ∈ P, where p′ ≠ q′.
  Select a set of candidate medoids M ⊂ X of size |M| = sk.
  for each c ∈ M do
    if swapping p and c would decrease the distortion cost then
      Perform the swap P ← P ∪ {c} \ {p}.
      Break from the inner loop.
    end if
    if swapping q and c would decrease the distortion cost then
      Perform the swap P ← P ∪ {c} \ {q}.
      Break from the inner loop.
    end if
  end for
until convergence
```

The simplest way of selecting the candidate medoid set $\mathcal{M}$ is via uniform random selection. However, this has the effect of over-representing larger clusters in the data set and missing smaller ones. Ideally, we would prefer each cluster to provide roughly s items of $\mathcal{M}$ regardless of its size. An appropriate alternative would be to bias the sampling in order to promote selection from smaller clusters, as we shall now describe.

4.1. Weighted random sampling

As a preprocessing step to the algorithm, we label each data point $x \in X$ using a coin-flipping procedure, as follows. With equal probability $\frac{1}{2}$ to each outcome — 'heads' or 'tails' — a sequence of coin flips is performed. The label $\omega(x)$ assigned to x is simply the number of 'heads' obtained until the first outcome of 'tails'. The labeling of X can be computed in linear expected time, as a preprocessing step. It is easy to see that the probability of x being labeled with label l or greater is simply $\mathbf{Pr}(\omega(x) \geq l) = \frac{1}{2^l}$.

The selection of $\mathcal{M}$ is performed as follows. Let $C = \{C_i \mid 1 \leq i \leq k\}$ be the clusters associated with the current set of centroids P. For each C_i, we select those members of $x \in C_i$ satisfying $\omega(x) \geq \lfloor \log_2 \frac{|C_i|}{s} \rfloor$. From the precomputed labeling, this selection can be performed in time proportional to $|C_i|$.

For each cluster $|C_i|$, the expected number of items selected can be seen to lie in the range $s \leq \mu_i < 2s$. Although we can expect s or more items of C_i to be included in $\mathcal{M}$, the random selection process may result in fewer than s candidates being generated, especially when $|C_i|$ is small. The probability of this occurring can be estimated using standard Chernoff bound techniques (Motwani & Raghavan, 1995). For simplicity and efficiency, in our experimentation, we shall consider the effect of setting s to be small constant values, without explicitly bounding the probability of error.

5. Experimental Evaluation

In this section, we evaluate the performance of the centroid-medoid algorithm against that of batch k-MEANS and k-MEDOIDS, as well as against two recent methods not based on k-MEANS: the constant-approximation heirarchical clustering heuristic of DasGupta and Long (DasGupta & Long, 2005), and the affinity propagation method of Frey and Dueck (Frey & Dueck, 2007).

DasGupta and Long's method initially constructs a spanning tree of the data set by means of a farthest-first traversal, in quadratic time. It then groups the items into 'levels of granularity' ordered according to the distances by which the items were connected into the spanning tree. A cluster hierarchy (cluster tree) is then formed by reconnecting each item of the spanning tree to the closest node taken from levels of higher granularity (shorter distances), thereby reducing overall connection distances and improving the compactness of the resulting clusters. Theoretically, if the cost of a clustering is taken to be the largest radius of its clusters, DasGupta and Long's method generates

a clustering with cost at most 8 times greater than the optimal. It should be noted that their 'minimax' cost function is more sensitive to outlier points than the sum-of-squares function used by k-MEANS, and that it is based on (unsquared) Euclidean distance rather than squares of distances. In practice, their method relies on parameters that control the numbers and sizes of the levels of granularity. In our experimentation, we report only the results using the default values recommended in their paper — varying them had little effect on the execution time or clustering quality.

Frey and Dueck's affinity propagation method for k-clustering takes as input a collection of real-valued similarities between data points, and iteratively propagates this information between neighboring items in an attempt to identify high-quality representatives in every local vicinity. The negotiation is mediated by real-valued messages between data points and potential representatives that are used to promote or inhibit the choice of representatives, and the assignment of points to representatives, so as to minimize an appropriately-chosen energy function. The affinity propogation method does not accept a desired number of clusters k. Instead, the number of clusters produced depends on a set of 'preference values', supplied by the user for each item, that expresses the relative suitability of the items for selection as a representative. Also, when the similarity is based on distance values, all pairwise distances must be precomputed, a prohibitive cost when the data set size is large.

The experimental comparison was performed on a dataset consisting of 1440 images grouped in 20 clusters, selected from the COIL-100 database (Nene et al., 1996). Each cluster contains 72 different views of the same physical object. The global image features used for the image database are described in (Boujemaa et al., 2001). The signature vector corresponding to each image has 120 dimensions. The clusterings were performed using the Euclidean distance metric, except in the case of affinity propagation, for which the negative squared Euclidean distance was used.

Figure 2. A sample of the COIL image database.

One way to measure the partition quality with regards to an underlying image database ground-truth is to measure the normalized mutual information (NMI) between true and predicted labels. If $\hat{L}$ is the random variable denoting the labels assigned by the algorithm, and L the random variable corresponding to the ground-truth labels, then the normalized mutual information NMI is the quantity

$$\mathrm{NMI} = 2\frac{\mathrm{H}(L) - \mathrm{H}(L|\hat{L})}{\mathrm{H}(L) + \mathrm{H}(\hat{L})}$$

where $\mathrm{H}(L)$, $\mathrm{H}(\hat{L})$ are the marginal entropies of L and $\hat{L}$, and $\mathrm{H}(L|\hat{L})$ is the conditional entropy. Simply stated, the NMI corresponds to the amount of information that knowing either variable provides about the other.

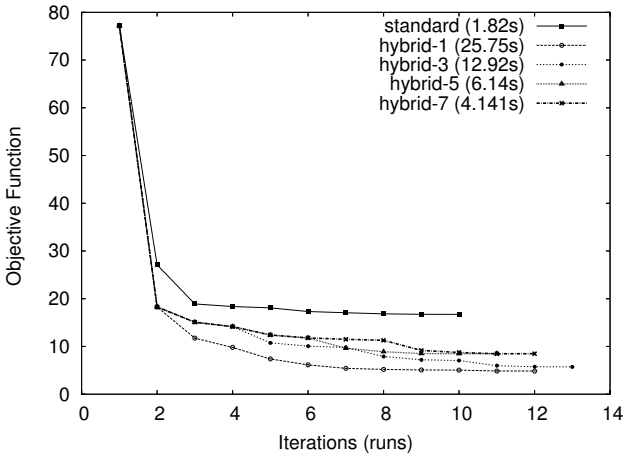

Figure 3. The hybrid algorithm achieves lower costs than batch k-MEANS, for $s = 2$ and $r = 1, 3, 5, 7$.

	NMI	Time (s)	#C
k-MEANS	0.759	1.28	20
Hybrid ($r = 3$, $s = 2$)	0.899	12.92	20
Hybrid ($r = 3$, $s = 10$)	0.893	52.02	20
k-MEDOIDS	0.907	352.08	20
DasGupta-Long	0.850	23.56	20
AP (prefs $= -0.5$)	0.866	59.53	55
AP (prefs $= -3.9$)	0.923	55.39	20

Table 1. The normalized mutual information for clusterings produced by batch k-MEANS, the hybrid algorithm with $r = 3$, k-MEDOIDS, DasGupta & Long's, and affinity propagation, with the corresponding execution times in seconds, and number of clusters.

The results of the experiments are shown in Table 5. For affinity propagation, we show the results for two choices of the preference values, one where values are set uniformly to the median similarity score (-0.5), and the other (-3.9) determined — after much trial and error — so as to produce the desired number of clusters, 20. The best NMI scores were achieved with

affinity propagation, although the execution times were higher than all but k-MEDOIDS due to the explicit computation of all pairwise distances. DasGupta and Long's farthest-first traversal method was outperformed in both execution time and clustering quality by the hybrid algorithm (with $s = 2$).

When comparing k-MEANS, k-MEDOIDS, and the hybrid method, we can see that evaluating even a very small number s of candidate medoids per cluster can lead to a significant improvement over batch k-MEANS, at speeds substantially faster than k-MEDOIDS. Figure 3 presents the dependence between the cost of solutions and the number of iterations for both batch k-means and the proposed hybrid algorithm, when s is fixed to be 2 and r is chosen to be one of 1, 3, 5 or 7. Once again, the hybrid algorithm outperforms batch k-MEANS by improving further upon the final solution cost of k-MEANS, and thus obtaining a better optimum approximation. Figure 3 also shows the associate times for convergence. Decreasing the number of runs in the centroid phase k-means iterations improves the convergence rate at the expense of execution time — 4.14 seconds to convergence when the centroid phase consists of 7 iterations, compared with 25.75 seconds when the centroid phase is limited to 1 iteration. Bearing this in mind, one should adjust the two parameters s and r depending on the application requirements.

6. Conclusion

In this paper we proposed a hybrid centroid-medoid local-search method for improving the performance of k-MEANS. We used a hierarchy of random samples to take into account variable cluster sizes while selecting good candidate medoids, and showed experimentally that the number of medoid swaps needed to reach a significant improvement of the convergence point is low. The simple hybridization allows for a competitively low computational complexity, making our approach suitable for the same applications as standard k-MEANS.

References

Arya, V., Garg, N., Khandekar, R., Munagala, K., & Pandit, V. (2001). Local search heuristic for k-median and facility location problems. *STOC '01: Proc. 33rd ACM Symposium on Theory of Computing* (pp. 21–29). New York, NY, USA: ACM Press.

Boujemaa, N., Fauqueur, J., Ferecatu, M., Fleuret, F., Gouet, V., Saux, B. L., & Sahbi, H. (2001). Ikona: Interactive generic and specific image retrieval. *Proc. International Workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR'2001)*. Rocquencourt, France.

Chen, J.-S., Ching, R. K. H., & Lin, Y.-S. (2004). An extended study of the k-means algorithm for data clustering and its applications. *J. Operational Research Society*, *55*, 976–987.

DasGupta, S., & Long, P. M. (2005). Performance guarantees for hierarchical clustering. *J. Computer and System Sciences*, *70*, 555–569.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *39*, 1–38.

Du, Q., Faber, V., & Gunzburger, M. (1999). Centroidal voronoi tessellations: Applications and algorithms. *SIAM Rev.*, *41*, 637–676.

Dubes, R. C., & Jain, A. K. (1976). Clustering techniques: The user's dilemma. *Pattern Recognition*, *8*, 247–260.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification*. John Wiley & Sons.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*, 972–976.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. Academic Press.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. New York, USA: John Wiley & Sons.

McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297).

Motwani, R., & Raghavan, P. (1995). *Randomized algorithms*. New York, USA: Cambridge University Press.

Nene, S. A., Nayar, S. K., & Murase, H. (1996). *Columbia object image library* (Technical Report). Department of Computer Science, Columbia University, http://www.cs.columbia.edu/CAVE/.