A novel game-theoretic initialisation process for the k-modes algorithm using the hospital-resident assignment problem

Henry Wilde, Vincent Knight and Jonathan Gillard

October 24, 2019

Abstract

The k-modes algorithm is a centroid-based clustering algorithm, and is an extension of the k-means algorithm for categorical data. This work outlines a comparison of the established initialisation methods for the k-modes algorithm by use of examples and algebraic analysis of their cost functions. In doing so, the effect of the initial centroid selection on the overall efficiency and quality of the final clustering found by each method is exposed.

Following this, a novel initialisation process is described that utilises game-theoretic results to create a fair and robust initial selection for the algorithm. This process is modelled on the hospital-resident assignment problem and is solved using an adapted Gale-Shapley algorithm.

The paper concludes with a comparison between the established initialisation methods and the proposed method on a number of benchmark datasets, as well as an analysis using preferable artificial datasets. The analysis uses several label-invariant metrics to assess the quality of the clustering both at the beginning of the algorithm and at the end.

1 The k-modes algorithm

The following notation will be used throughout this work to describe the objects associated with clustering a dataset:

- Let $\mathcal{A} := A_1 \times \cdots \times A_m$ denote the *attribute space*. In this work, only categorical attributes are considered and so it is intuitive to describe each attribute as a set of its values, i.e. for each $j = 1, \ldots, m$ it follows that $A_j := \left\{a_1^{(j)}, \ldots, a_{d_j}^{(j)}\right\}$ where $d_j = |A_j|$ is considered the size of the j^{th} attribute.

- Let $\mathcal{X} := \left\{X^{(1)}, \ldots, X^{(N)}\right\} \subset \mathcal{A}$ denote a *dataset* where each $X^{(i)} \in \mathcal{X}$ is defined as an m-tuple $X^{(i)} := \left(x_1^{(i)}, \ldots, x_m^{(i)}\right)$ where $x_j^{(i)} \in A_j$ for each $j = 1, \ldots, m$. The elements of $\mathcal{X}$ are referred to as *data points* or *instances*. A dataset $\mathcal{X}$ can be represented as a table like so:

	A_1	A_2	$\ldots$	A_{m-1}	A_m
$X^{(1)}$	$x_1^{(1)}$	$x_2^{(1)}$	$\ldots$	$x_{m-1}^{(1)}$	$x_m^{(1)}$
$X^{(2)}$	$x_1^{(2)}$	$x_2^{(2)}$	$\ldots$	$x_{m-1}^{(2)}$	$x_m^{(2)}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$X^{(N)}$	$x_1^{(N)}$	$x_2^{(N)}$	$\ldots$	$x_{m-1}^{(N)}$	$x_m^{(N)}$

- Let $\mathcal{Z} := (Z_1, \ldots, Z_k)$ be a partition of a dataset $\mathcal{X}$ into $k \in \mathbb{Z}^+$ distinct, non-empty parts. Such a partition $\mathcal{Z}$ is called a *clustering* of $\mathcal{X}$.

- Each cluster Z_l has associated with it a *representative point* (see Definition 1.2) which is denoted by $z^{(l)} = \left(z_1^{(l)}, \ \ldots, \ z_m^{(l)}\right) \in \mathcal{A}$. These points may also be referred to as cluster modes. The set of all current representative points is denoted $\overline{Z} = \left\{z^{(1)}, \ldots, z^{(k)}\right\}$.

An immediate difference between the k-means and k-modes algorithms is that they deal with different types of data, and so the metric used to define the distance between two points

in our space must be different. With k-means, where the data has all-numeric attributes, Euclidean distance is often used. However, we do not have this sense of distance with categorical data. Instead, we utilise a dissimilarity measure - defined below - as our metric. It can be easily checked that this is indeed a distance measure.

Definition 1.1. Let $\mathcal{X}$ be a dataset and consider any $X^{(a)}, X^{(b)} \in \mathcal{X}$. The dissimilarity between $X^{(a)}$ and $X^{(b)}$, denoted by $d\left(X^{(a)}, X^{(b)}\right)$, is given by:

$$d(X^{(a)}, X^{(b)}) := \sum_{j=1}^{m} \delta(x_j^{(a)}, x_j^{(b)}) \quad \text{where} \quad \delta(x, y) = \begin{cases} 0, & \text{if } x = y \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

In other words, the dissimilarity between two points is the number of attributes where their values are not the same.

Now that we have defined a metric on our space, we can turn our attention to what we mean by the representative point $\mu^{(l)}$ of a cluster C_l. In k-means, these representative points are called centroids or cluster centers and are defined to be the average of all points $X^{(i)} \in C_l$. With categorical data, we use our revised distance measure from Definition 1.1 to specify a representative point. We typically call these points a mode of $\mathbf{X}$.

Definition 1.2. Let $\mathcal{X} \subset \mathcal{A}$ be a dataset and consider some point $z = (z_1, \ldots, z_m) \in \mathcal{A}$. Then z is called a *mode* of $\mathcal{X}$ if it minimises the following:

$$D(\mathcal{X}, z) = \sum_{i=1}^{N} d\left(X^{(i)}, z\right) \tag{2}$$

Definition 1.3. Let $\mathcal{X} \subset \mathcal{A}$ be a dataset. Then $n(a_s^{(j)})$ denotes the *frequency* of the s^{th} category $a_s^{(j)}$ of A_j in $\mathcal{X}$, i.e. for each $A_j \in \mathcal{A}$ and each $s = 1, \ldots, d_j$:

$$n(a_s^{(j)}) := \left| \left\{ X^{(i)} \in \mathcal{X} : x_j^{(i)} = a_s^{(j)} \right\} \right| \tag{3}$$

Furthermore, $\frac{n(a_s^{(j)})}{N}$ is called the *relative frequency* of category $a_s^{(j)}$ in $\mathbf{X}$.

Theorem 1. Consider a dataset $\mathcal{X} \subset \mathcal{A}$ and some $U = (u_1, \ldots, u_m) \in \mathcal{A}$. Then $D(\mathcal{X}, U)$ is minimised if and only if $n(u_j) \geq n(a_s^{(j)})$ for all $s = 1, \ldots, d_j$ for each $j = 1, \ldots, m$.

A proof of this theorem can be found in the Appendix of [2].

Definition 1.4. Let $\mathcal{Z} = \{Z_1, \ldots, Z_k\}$ be a clustering of a dataset $\mathcal{X}$, and let $\overline{Z} = \left\{z^{(1)}, \ldots, z^{(k)}\right\}$ be the corresponding cluster modes. Then $W = (w_{i,l})$ is an $N \times k$ *partition matrix* of $\mathcal{X}$ such that:

$$w_{i,l} = \begin{cases} 1, & \text{if } X^{(i)} \in Z_l \\ 0, & \text{otherwise.} \end{cases}$$

With this, the *cost function* is defined to be the summed within-cluster dissimilarity:

$$C\left(W, \overline{Z}\right) = \sum_{l=1}^{k} \sum_{i=1}^{N} \sum_{j=1}^{m} w_{i,l}\ \delta\left(x_j^{(i)}, z_j^{(l)}\right) \tag{4}$$

Below is a practical implementation of the k-modes algorithm [2]:

Remark. The processes by which the k initial modes are selected are detailed in Sections 2 & 4.

2 Initialisation processes

From the literature surrounding this topic, it has been established that the initial choice of clusters impacts the final solution of the k-modes algorithm [1, 2]. While some works attempt to improve the quality of k-modes and similar algorithms by considering an alternative dissimilarity measure [5], this work will examine the way in which these k initial representative points are chosen. Two established methods of selecting these initial points are described in Sections 2.1 & 2.2.

Algorithm 1: The k-modes algorithm

Input: a dataset $\mathcal{X}$, a number of clusters to form k
Output: a clustering $\mathcal{Z}$ of $\mathcal{X}$
Select k initial modes $z^{(1)}, \ldots, z^{(k)} \in \mathcal{X}$
$\overline{Z} \leftarrow \left\{z^{(1)}, \ldots, z^{(k)}\right\}$
$\mathcal{Z} \leftarrow \left(\left\{z^{(1)}\right\}, \ldots, \left\{z^{(k)}\right\}\right)$
for $X^{(i)} \in \mathcal{X}$ **do**
    Select l^* that minimises $d(X^{(i)}, z^{(l)})$
    $Z_{l^*} \leftarrow Z_{l^*} \cup \left\{X^{(i)}\right\}$
    UPDATE($z^{(l^*)}$)
end
repeat
    for $X^{(i)} \in \boldsymbol{X}$ **do**
        for $\mu^{(l)} \in \bar{\mu}$ **do**
            Calculate $d(X^{(i)}, \mu^{(l)})$
        end
        if $d(X^{(i)}, \mu^{(l^*)}) > d(X^{(i)}, \mu^{(l')})$ *for some* $l' \neq l^*$ **then**
            $C_{l^*} \leftarrow C_{l^*} \setminus \left\{X^{(i)}\right\}$
            $C_{l'} \leftarrow C_{l'} \cup \left\{X^{(i)}\right\}$
            UPDATE($\mu^{(l^*)}$) and UPDATE($\mu^{(l')}$).
        end
    end
until *No point changes cluster after a full cycle through* $\mathcal{X}$

Algorithm 2: UPDATE

Input: an attribute space $\mathcal{A}$, a mode to update $z^{(l)}$ and its cluster Z_l
Output: an updated mode
Find $z \in \mathcal{A}$ that minimises $D(Z_l, z)$
$z^{(l)} \leftarrow z$

2.1 Huang's method

In the standard form of the k-modes algorithm, the k initial modes are chosen at random from $\mathbf{X}$. Below is an alternative method of selecting these modes that forces some diversity between them, as described in [2]. Here, we consider two sets of modes, $\tilde{\mu}$ and $\bar{\mu}$. The former acts as a placeholder set of modes, whereas the latter is the set of modes to go on to be used by the k-modes algorithm.

Algorithm 3: Huang's method

[0] **Input:** a dataset $\mathbf{X}$, with attribute sets $A_1, \ldots, A_m$, and a number of modes to find k **Output:** a set of k initial modes $\bar{\mu}$
Initialisation step $\tilde{\mu} \leftarrow \emptyset$ $\bar{\mu} \leftarrow \emptyset$ **for** $j = 1, \ldots, m$ **do**
end
$s = 1, \ldots, d_j$ Calculate the relative frequency of each attribute value: $\frac{n(a_s^{(j)})}{N}$.
Distribute most common attribute values **for** $l = 1, \ldots, k$ **do**
end
$j = 1, \ldots, m$ Sample $a_{s^*}^{(j)}$ from A_j by considering the relative frequencies of the elements of A_j as a probability distribution. $\mu_j^{(l)} \leftarrow a_{s^*}^{(j)}$ $\tilde{\mu} \leftarrow \tilde{\mu} \cup \left\{\mu^{(l)}\right\}$
Replace $\tilde{\mu}$ with points in $\mathbf{X}$ to avoid empty clusters **for** $\mu \in \tilde{\mu}$ **do**
end
Select $X^{(i^*)} \in \mathbf{X}$ such that:

$$X^{(i^*)} = \underset{1 \leq i \leq N}{\arg\min} \left\{ d(X^{(i)}, \mu) : \ X^{(i^*)} \neq \mu' \ \forall \mu' \in \bar{\mu} \right\}$$

$\bar{\mu} \leftarrow \bar{\mu} \cup \left\{X^{(i^*)}\right\}$

In the original statement of Huang's method, the algorithm states that the most frequent categories should be assigned 'equally' to the k initial modes. How the categories should be distributed 'equally' is not well-defined or easily seen from the example given. This ambiguity in the definition of Huang's method means that a probabilistic element must be introduced, and unless seeded pseudo-random numbers are used, computer-generated results are not necessarily reproducible.

In this work, as is done in the implementation used to apply the k-modes algorithm in Section 6, the term 'equally' is considered to mean taking a sample from a probability

distribution. This distribution is formed by the relative frequencies of the attributes' values (defined in Definition 1.3), as is described in Algorithm 3.

In practice, taking a random sample according to some probability distribution will lead to variation between runs of this method. As such, when Huang's method is used to initialise the k-modes algorithm it is typically run multiple times and the result with lowest final cost is used.

2.2 Cao's method

Cao's method selects representative points by the average density of a point in the dataset. As will be seen in the following definition, this average density is in fact the average relative frequency of all the attribute values of that point. This method is considered deterministic as there is no probabilistic element unlike Huang's method or a random initialisation. So, we can consider the results to be largely reproducible, except in the case where a tie must be broken (see Example **??**).

Definition 2.1. Consider a data set $\mathbf{X}$ with attribute set $\mathbf{A} = \{A_1, \ldots, A_m\}$. Then the *average density* of any point $X_i \in \mathbf{X}$ with respect to $\mathbf{A}$ is defined [1] as:

$$\text{Dens}(X^{(i)}) = \frac{\sum_{j=1}^{m} \text{Dens}_j(X^{(i)})}{m}, \quad \text{where} \quad \text{Dens}_j(X^{(i)}) = \frac{|\{X^{(t)} \in \mathbf{X} : x_j^{(i)} = x_j^{(t)}\}|}{N} = \frac{n(x_j^{(i)})}{N}$$

Observe that:

$$|\{X^{(t)} \in \mathbf{X} : x_j^{(i)} = x_j^{(t)}\}| = n(x_j^{(i)}) = \sum_{t=1}^{N} (1 - \delta(x_j^{(i)}, x_j^{(t)}))$$

And so, we can find an alternative definition for $\text{Dens}(X^{(i)})$:

$$\begin{aligned}
\text{Dens}(X^{(i)}) &= \frac{1}{mN}\sum_{j=1}^{m}\sum_{t=1}^{N}(1-\delta(x_j^{(i)},x_j^{(t)})) \\
&= \frac{1}{mN}\sum_{j=1}^{m}\sum_{t=1}^{N}1 - \frac{1}{mN}\sum_{j=1}^{m}\sum_{t=1}^{N}\delta(x_j^{(i)},x_j^{(t)}) \\
&= \frac{mN}{mN} - \frac{1}{mN}\sum_{t=1}^{N}d(X^{(i)},X^{(t)}) \\
&= 1 - \frac{1}{mN}D(\mathbf{X},X^{(i)})
\end{aligned} \tag{5}$$

Remark. It is worth noting that we have $\frac{1}{N} \leq \text{Dens}(X^{(i)}) \leq 1$, since:

- If $n(x_j^{(i)}) = 1$ for all $j = 1,\ldots,m$, then $\text{Dens}(X^{(i)}) = \frac{\sum_{j=1}^{m}\frac{1}{N}}{m} = \frac{m}{mN} = \frac{1}{N}$
- If $n(x_j^{(i)}) = N$ for all $j = 1,\ldots,m$, then $\text{Dens}(X^{(i)}) = \frac{\sum_{j=1}^{m}1}{m} = \frac{m}{m} = 1$

Algorithm 4: Cao's method

[0] **Input:** a dataset $\mathbf{X}$, with attribute sets $A_1, \ldots, A_m$, and a number of modes to find k **Output:** a set of k initial modes $\bar{\mu}$

Initialisation step $\bar{\mu} \leftarrow \emptyset$ **for** $X^{(i)} \in \boldsymbol{X}$ **do**

end

Calculate $\text{Dens}(X^{(i)})$.

Select the point with maximal density Select $X^{(i_1)} \in \mathbf{X}$ which satisfies:

$$X^{(i_1)} = \underset{1 \leq i \leq N}{\arg\max} \left\{ \text{Dens}(X^{(i)}) \right\}$$

$\bar{\mu} \leftarrow \bar{\mu} \cup \left\{ X^{(i_1)} \right\}$

Second point maximises both density and distance from the first mode Select $X^{(i_2)} \in \mathbf{X}$ which satisfies:

$$X^{(i_2)} = \underset{1 \leq i \leq N}{\arg\max} \left\{ \text{Dens}(X^{(i)}) \times d(X^{(i)}, X^{(i_1)}) \right\}$$

$\bar{\mu} \leftarrow \bar{\mu} \cup \left\{ X^{(i_2)} \right\}$

Continue to choose points in this fashion until k are chosen **while** $|\bar{\mu}| < k$ **do**

end

Select $X^{(i_3)} \in \mathbf{X}$ which satisfies:

$$X^{(i_3)} = \underset{1 \leq i \leq N}{\arg\max} \left\{ \min_{\mu^{(l)} \in \bar{\mu}} \left\{ \text{Dens}(X^{(i)}) \times d(X^i, \mu^{(l)}) \right\} \right\}$$

$\bar{\mu} \leftarrow \bar{\mu} \cup \left\{ X^{(i_3)} \right\}$

Remark. With this alternative definition, we see - since m and N are fixed positive integers - that $\text{Dens}(X^{(i)})$ is maximised when $D(\mathbf{X}, X^{(i)})$ is minimised. Then by Theorem 1 we have that such an $X^{(i)}$ with maximal average density in $\mathbf{X}$ with respect to $\mathbf{A}$ is, in fact, a mode of $\mathbf{X}$. This observation allows us to consider some sense of similarity between Huang and Cao's methods, as they seem to be trying to achieve the same objective - if only from opposite ends.

3 Matching games

The primary motivation for this work is to move away from the greedy approaches defined above by incorporating some techniques from game theory, namely: matching games. The purpose of solving a matching game is to link the elements of two sets in a 'fair' way so that no element could feasibly be better off. By considering the virtual modes found during Huang's method with some suitable subset of our dataset as a matching game to be solved, we hope to find a closer-to-optimal set of initial modes for the k-modes algorithm.

Definition 3.1. Consider two sets S, R each of size N, and let us call them 'suitors' and 'reviewers'. Each element of S and R has associated with it a preference list of the other set's elements. These preference lists are ranked in descending order. We consider the preference lists as functions which produce tuples, f and g respectively:

$$f : S \to R^n,\ g : R \to S^n$$

This construction of sets and preference lists is called a *matching game* of size N and we denote the game itself by (S, R).

A matching, M, is defined to be any bijection between S and R. If $s \in S$ and $r \in R$ are matched by M, then we write $M(s) = r$. A matching M is considered to be either stable or unstable based on the preference lists of its suitors and reviewers, and the presence of blocking pairs.

Definition 3.2. Let (S, R) be a matching game of size N with some matching M. A pair $(s, r) \in S \times R$ is said to *block* M if $M(s) \neq r$ but s prefers r to $M(s) = r'$ and r prefers s to $M^{-1}(r) = s'$. That is, r appears before r' in $f(s)$ and s appears before s' in $g(r)$.

Definition 3.3. Let (S, R) be a matching game of size N with some matching M. Then we say M is a *stable matching* if there are no blocking pairs, and *unstable* otherwise.

3.1 The Gale-Shapley algorithm

The Gale-Shapley algorithm is known to find a unique stable matching for any matching game of size N. This matching is also considered to be suitor-optimal. That is, each suitor is matched with the best possible reviewer that ensures a stable matching, but is in fact the worst possible matching for the reviewers. [cite or have these theorems stated/proven]

As was discussed at the start of Section 3, the outline of the method proposed in this paper is to extend Huang's method by considering the virtual modes with some subset of the data as a matching game to be solved. It should be noted, however, that in this method we may not have equally sized sets of suitors and reviewers. As a result of this, the Gale-Shapley algorithm becomes inapplicable as the matching produced M would not be a bijection of our suitors and reviewers.

3.2 The capacitated Gale-Shapley algorithm for the hospital-resident problem

The situation where a large set of suitors are to be matched with a number reviewers is not limited to abstraction. A practical example of this problem is how to best assign a cohort of medical students to a group of hospitals. Here, we have all of the requisite components of a matching game:

- A set of reviewers (the hospitals) and a set of suitors (the potential residents)
- A ranking of the students/residents by the hospitals, and vice versa

The only obstacle which stops us from using the Gale-Shapley algorithm is the disparity in the sizes of our sets. In reality, hospitals need not always take at most one resident on from a cohort of medical students. So each hospital has a capacity associated with it and we can consider our matching game to be 'capacitated'. By this we mean that each reviewer

(hospital) may be matched with any number of suitors (students) up to their capacity, making our matching $M : S \rightarrow R$ surjective.

Research surrounding the hospital-resident assignment problem is well-documented [cite] and an extension of the Gale-Shapley algorithm was developed to solve it, awarding the authors the 2012 Nobel Prize in Economic Sciences. This algorithm is currently used by the National Resident Matching Program (http://www.nrmp.org) to assign medical students to hospitals in the United States of America.

As before, we consider a set of suitors and reviewers denoted by S and R. These sets are no longer (necessarily) the same size. We also have our preference lists f, g, and a set $C = \{c_1, \ldots, c_{|R|}\}$ of reviewer capacities. Finally, let $S_u \subset S$ denote the set of suitors that are currently unmatched. The capacitated Gale-Shapley algorithm is given below.

Algorithm 5: Capacitated Gale-Shapley

[0] **Input:** two sets of objects S, R, a set of capacities for the elements of R, denoted by $C = \{c_1, \ldots, c_{|R|}\}$, and two preference list functions, f, g **Output:** a surjective matching M between S and R
Initialise all suitors and reviewers to be unmatched **for** $s \in S$ **do**
end
$M(s) \leftarrow \emptyset$ **for** $r \in R$ **do**
end
$M^{-1}(r) \leftarrow \emptyset$ $S_u \leftarrow S$
Find a tentative match for some currently unmatched suitor **while** $|S_u| > 0$ **do**
end
Select any $s \in S_u$. **if** $|f(s)| = 0$ **then**
end
$S_u \leftarrow S_u \setminus \{s\}$ **else**
end
Select s's most preferred reviewer $r \in R$.
If r has room, match to s **if** $|M(r)| < c_r$ **then**
end
$M(r) \leftarrow M(r) \cup \{s\}$ $S_u \leftarrow S_u \setminus \{s\}$ **else**
end
Otherwise determine whether their matching would be preferable to r over its worst current matching Find $s' \in M(r)$ such that r prefers s' least. **if** r *prefers* s *to* s' **then**
end
$M(r) \leftarrow M(r) \cup \{s\}$ $S_u \leftarrow S_u \cup \{s\}$ $M(r) \leftarrow M(r) \setminus \{s'\}$ $S_u \leftarrow S_u \cup \{s'\}$ **else**
end
$f(s) \leftarrow f(s) \setminus \{r\}$

4 The proposed method

Now that we have defined what we mean by a matching game, with the algorithm described above, we can construct an alternative initialisation process for the k-modes algorithm.

Let $\mathbf{X}$ be a dataset with attribute set $\mathbf{A}$, and let $\tilde{\mu}$ be the set of virtual modes found by Huang's method (i.e. the set of virtual centroids to be assigned to points in $\mathbf{X}$). We then take this set of virtual modes $\tilde{\mu}$ and construct a capacitated matching game to be solved by the capacitated Gale-Shapley algorithm in the following way.

Algorithm 6: The proposed initialisation method

[0] **Input:** a dataset $\mathbf{X}$, with attribute sets $A_1, \ldots, A_m$, and a number of modes to find k **Output:** a set of k initial modes $\bar{\mu}$
Initialisation step $\tilde{\mu} \leftarrow \emptyset$ $\bar{\mu} \leftarrow \emptyset$ $R \leftarrow \emptyset$ $S \leftarrow \emptyset$ $C \leftarrow \{c_1, \ldots, c_k\}$
for $j = 1, \ldots, m$ **do**
end
$s = 1, \ldots, d_j$ Calculate the relative frequency of each attribute value: $\frac{n(a_s^{(j)})}{N}$.
Find the set of virtual modes, $\tilde{\mu}$, according to Huang's method. **for** $l = 1, \ldots, k$ **do**
end
$j = 1, \ldots, m$ Consider the probability distribution given by:
$\mathbb{P}(A_j) := \left(\frac{n(a_s^{(j)})}{N} : a_s^{(j)} \in A_j\right)$ Sample $a_{s^*}^{(j)}$ from A_j with respect to $\mathbb{P}(A_j)$. $\mu_j^{(l)} \leftarrow a_{s^*}^{(j)}$
$\tilde{\mu} \leftarrow \tilde{\mu} \cup \left\{\mu^{(l)}\right\}$
Construct and solve a capacitated matching game. $R \leftarrow \tilde{\mu}$ **for** $r \in R$ **do**
end
$c_r \leftarrow 1$ Find the set of k vectors, S_r, in $\mathbf{X}$ that are the least dissimilar to r. Arrange S_r into descending order of similarity. $S \leftarrow S \cup S_r$ **for** $r \in R$ **do**
end
$g(r) \leftarrow S_r$ Select a method for suitor preference lists and construct $f(s)$ accordingly for each $s \in S$. Solve the capacitated matching game defined by (S, R, C) to obtain a matching $M : R \rightarrow S$. **for** $r \in R$ **do**
end
$\bar{\mu} \leftarrow \bar{\mu} \cup \{M(r)\}$

Remark. The method for constructing the preference lists of our suitors can affect the outcome and performance of this method. Please refer to Sections 5 & 6.

5 Resident preference lists

For the purposes of this piece of work and to demonstrate how preference lists can be generated, a small number will be defined and used. While these preference lists do not necessarily hold any mathematical justification for why they could be appropriate or useful at all, they are the simplest methods available. In fact, research into this area could prove to be promising as a means of incorporating prior or expert knowledge into the clustering algorithm.

Definition 5.1. The three preference list methods to be used will be referred to as 'Best', 'Worst' and 'Random' from this point, most notably in Section 6. Their definitions are somewhat self-explanatory but are given below.

Let (S, R, C) be a capacitated matching game where $R = \tilde{\mu}$ and $S = \{S_r : r \in R\}$ as in the proposed method. Then the reviewer-suitor preference function g is well-defined by the proposed method. Consider some $s \in S$. Then their preference function $f(s)$ could be defined in the following way:

- **Best:** Rank the elements of R in *ascending* order of dissimilarity with respect to s and set this to be $f(s)$.
- **Worst:** Rank the elements of R in *descending* order of dissimilarity with respect to s and set this to be $f(s)$.
- **Random:** Take a random permutation of the elements of R and set this to be $f(s)$.

Remark. It should be noted that 'Best' could be considered the greediest approach to take since it involves choosing the most preferred option for each of the slots available in the list. 'Random' should also be expected to perform badly on average since it creates an element of stupidity in a method that is intended to be an intelligent cluster selection. 'Worst' is included to observe the effects of deliberately choosing a preference list that goes against common sense approaches.

Note also that each of these preference list operators requires a suitor to rank all of the reviewers. A slight modification to some (such as the inclusion of a threshold on dissimilarity) could drastically improve their performance. Also, it is possible to generate a preference list such that the proposed method produces an initial clustering identical to that found by Huang's method, though that is not discussed here as it has been considered trivial.

6 Experimental results

To give comparative results on the quality of the initialisation processes defined in Sections 2, 4 & 5, four well-known, categorical, labelled datasets – soybean, mushroom, breast cancer, and zoo animal – will be clustered by the k-modes algorithm with each of the initialisation processes using their respective number of classes as the number of clusters. These datasets have been chosen to fall in line with the established literature, and for their relative sizes and complexities.

Typically, the quality of a clustering algorithm is measured by its performance at classifying datasets [1, 2, 6]. In this work, however, we will not follow this approach since our motivation is to compare the quality of the clustering produced when using these initialisation methods. So, for the purposes of measuring the performance of our various initialisation methods as parts of a clustering algorithm, we will make use of internal metrics that are independent of any external information such as a class labelling. This family of metrics are built up from two characteristics of the clusters found: cohesion and separation. Cluster cohesion is effectively the summed, within-cluster variation or dissimilarity of its points, whereas a cluster's separation is a sum of the distances between all points in the cluster and every other point not in the cluster. In this analysis, we will make use of two internal measures for cluster validity: our cost function from Definition 1.4 and the average silhouette coefficient, or silhouette score, of our clustering, defined below.

Definition 6.1. Let $\mathbf{X}$ be a dataset and consider a clustering of $\mathbf{X}$ into k parts, denoted by $C = \{C_1, \ldots, C_k\}$. For each $X^{(i)} \in \mathbf{X}$, we define the following two quantities:

- Let $a\left(X^{(i)}\right)$ denote the average dissimilarity between $X^{(i)}$ and every other point in its cluster. Without loss of generality, let $X^{(i)} \in C_l$. Then:

$$a\left(X^{(i)}\right) := \frac{1}{|C_l|} D\left(C_l, X^{(i)}\right)$$

- Let $b\left(X^{(i)}\right)$ denote the lowest average dissimilarity between $X^{(i)}$ and all other points in each cluster other than C_l. That is:

$$b\left(X^{(i)}\right) := \min_{l' \neq l} \left\{ \frac{1}{|C_{l'}|} D\left(C_{l'}, X^{(i)}\right) \right\}$$

With these quantities we define, for each point in our datset, their *silhouette coefficient*, denoted by $s(X^{(i)})$:

$$s(X^{(i)}) := \frac{b\left(X^{(i)}\right) - a\left(X^{(i)}\right)}{\max\left\{a\left(X^{(i)}\right), b\left(X^{(i)}\right)\right\}}$$

The *silhouette score* of a clustering C is simply the average of all the silhouette coefficients. Silhouette scores take value in the range $[-1, 1]$. Negative scores generally suggest that elements in the data have been mis-clustered since there exists a closer cluster centre than its own. Values around 0 indicate overlapping clusters, whereas silhouette scores close to 1 suggest well-separated and effective clusters.

6.1 The datasets

As stated above, the datasets being used for this work are well-known and openly available. Below is a summary of their properties and access links for each.

Soybean

The soybean dataset describes 35 characteristics of 307 soybean instances to classify which disease is present. The attributes are encoded numerically as integers but will be considered as strings for this analysis. The diseases form .8 classes, though the first 15 are the only ones used since they contain a considerable number of instances each [4]. Available at: https://archive.ics.uci.edu/ml/datasets/Soybean+(Large).

Mushroom

The mushroom dataset was constructed to classify 8124 mushroom instances forming 23 species found in North America into two classes: edible and poisonous. The attributes describe the physical characteristics and habitat of the mushrooms, and are encoded as strings [3]. Available at: https://archive.ics.uci.edu/ml/datasets/mushroom.

Breast cancer

Wisconsin University constructed the breast cancer dataset using a decision tree with linear programming as a diagnostic tool. The features were created using digital images of a fine needle aspirate of a breast mass to describe the structure of cell nuclei. There are 699 instances and 32 attributes in total. Available upon request to members of the academic community at: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic).

Zoo animal

The zoo animal dataset is an entirely artificial dataset used to classify 101 animals into 7 classes, those being mammal, reptile, amphibian, bird, fish, insect, and crustacean. The 17 attributes include the name of the animal and a series of Boolean variables describing characteristics and the habitat of the animals. Available at: http://archive.ics.uci.edu/ml/datasets/zoo.

6.2 Results

In this section, two sets of results will be considered. The first are the more classically seen tables of metrics defined above, and the latter are a collection of plots showing the descent in the cost function of the k-modes algorithm over time. In either case, results are generated using the Python library `kmodes` to which the proposed method has been added as

another initialisation method. The number of clusters to be determined, k, is chosen as the number of classes associated with each dataset. Note that this value may not be optimal (as suggested by the relatively low silhouette scores in most cases), and that the class variable is not considered in the running of the algorithm.

6.2.1 Metric results

Each of the tables of results given below were obtained by running the k-modes algorithm 25 times with each initialisation method on the dataset in question. For each of these 25 runs, the simulation is seeded to make the results reproducible.

At each run of the experiment the number of epochs to termination, the initial and final costs, and the average silhouette score were recorded for the clustering found. These metrics are summarised below in Tables 1 - 4 by their mean and median values, and their standard deviation over the 25 runs.

Initialisation		No. of iterations	Initial cost	Final cost	Silhouette score
Cao	mean	2.0000	1574.0000	1314.0000	0.2349
	median	2.0000	1574.0000	1314.0000	0.2349
	std	0.0000	0.0000	0.0000	0.0000
Huang	mean	3.7600	1795.4400	1455.3200	0.1212
	median	4.0000	1830.0000	1428.0000	0.1234
	std	1.0520	131.9621	72.7637	0.0357
Matching Best	mean	3.6800	1773.2400	1449.6400	0.1205
	median	3.0000	1781.0000	1443.0000	0.1193
	std	1.1075	118.8992	63.0971	0.0247
Matching Random	mean	3.8800	1774.3600	1438.8000	0.1237
	median	4.0000	1780.0000	1431.0000	0.1291
	std	1.0924	121.9911	54.0301	0.0234
Matching Worst	mean	4.0400	1774.9600	1431.8400	0.1290
	median	4.0000	1777.0000	1430.0000	0.1291
	std	1.0985	123.0411	53.9859	0.0248
Random	mean	3.6800	1638.8000	1351.2400	0.1608
	median	3.0000	1626.0000	1345.0000	0.1620
	std	1.1804	98.2480	36.3126	0.0312

Table 1: Summative metric results for the soybean dataset with $k = 15$.

Initialisation		No. of iterations	Initial cost	Final cost	Silhouette score
Cao	mean	2.0000	63242.0000	62644.0000	0.2517
	median	2.0000	63242.0000	62644.0000	0.2517
	std	0.0000	0.0000	0.0000	0.0000
Huang	mean	2.8400	75107.8400	63231.8400	0.2274
	median	3.0000	76026.0000	63002.0000	0.2179
	std	0.9866	5864.1758	1059.4049	0.0332
Matching Best	mean	2.8400	74822.2800	63319.5600	0.2209
	median	3.0000	75192.0000	63015.0000	0.2179
	std	1.0279	4914.2731	1186.2087	0.0345
Matching Random	mean	2.8400	74822.2800	63319.5600	0.2209
	median	3.0000	75192.0000	63015.0000	0.2179
	std	1.0279	4914.2731	1186.2087	0.0345
Matching Worst	mean	2.8400	74822.2800	63319.5600	0.2209
	median	3.0000	75192.0000	63015.0000	0.2179
	std	1.0279	4914.2731	1186.2087	0.0345
Random	mean	2.6400	78345.8400	63824.8000	0.2243
	median	2.0000	78603.0000	63155.0000	0.2281
	std	0.8103	6146.4742	1684.9847	0.0391

Table 2: Summative metric results for the mushroom dataset with $k = 2$.

Initialisation		No. of iterations	Initial cost	Final cost	Silhouette score
Cao	mean	3.0000	3546.0000	3250.0000	0.3453
	median	3.0000	3546.0000	3250.0000	0.3453
	std	0.0000	0.0000	0.0000	0.0000
Huang	mean	2.0400	3809.3200	3428.3200	0.1655
	median	2.0000	3846.0000	3492.0000	0.0933
	std	0.4546	206.2981	151.9832	0.1595
Matching Best	mean	2.0400	3804.7200	3386.4000	0.1943
	median	2.0000	3803.0000	3257.0000	0.3439
	std	0.4546	177.0794	146.2635	0.1681
Matching Random	mean	2.0400	3804.7200	3386.4400	0.1940
	median	2.0000	3803.0000	3257.0000	0.3439
	std	0.4546	177.0794	146.2247	0.1677
Matching Worst	mean	2.0400	3804.7200	3386.4400	0.1940
	median	2.0000	3803.0000	3257.0000	0.3439
	std	0.4546	177.0794	146.2247	0.1677
Random	mean	2.0400	4150.8000	3349.9200	0.2338
	median	2.0000	3682.0000	3250.0000	0.3453
	std	0.3512	828.8702	136.8819	0.1631

Table 3: Summative metric results for the breast cancer dataset with $k = 2$.

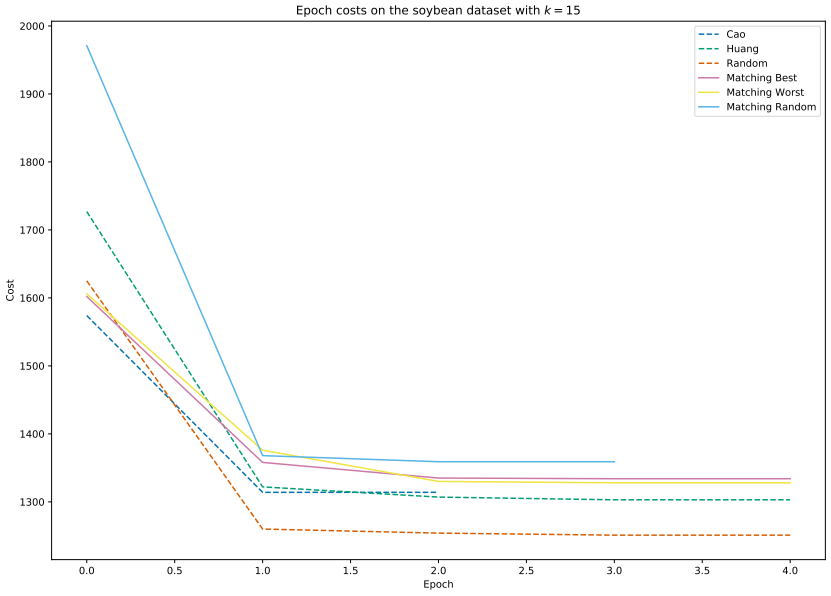
Initialisation		No. of iterations	Initial cost	Final cost	Silhouette score
Cao	mean	2.0000	377.0000	232.0000	0.3860
	median	2.0000	377.0000	232.0000	0.3860
	std	0.0000	0.0000	0.0000	0.0000
Huang	mean	2.4800	360.2800	250.3200	0.3047
	median	2.0000	358.0000	244.0000	0.3271
	std	0.6532	41.1213	15.2226	0.1038
Matching Best	mean	2.3600	359.6800	251.8400	0.3045
	median	2.0000	352.0000	252.0000	0.3184
	std	0.4899	42.6602	12.5322	0.0926
Matching Random	mean	2.4000	360.5200	252.6000	0.3083
	median	2.0000	352.0000	253.0000	0.3300
	std	0.5000	42.5853	12.1587	0.0938
Matching Worst	mean	2.4800	367.0000	249.8800	0.3047
	median	2.0000	356.0000	252.0000	0.3138
	std	0.6532	42.7444	9.8375	0.0729
Random	mean	2.4000	340.0800	265.2000	0.2658
	median	2.0000	335.0000	253.0000	0.3122
	std	0.7638	58.5227	27.1953	0.1204

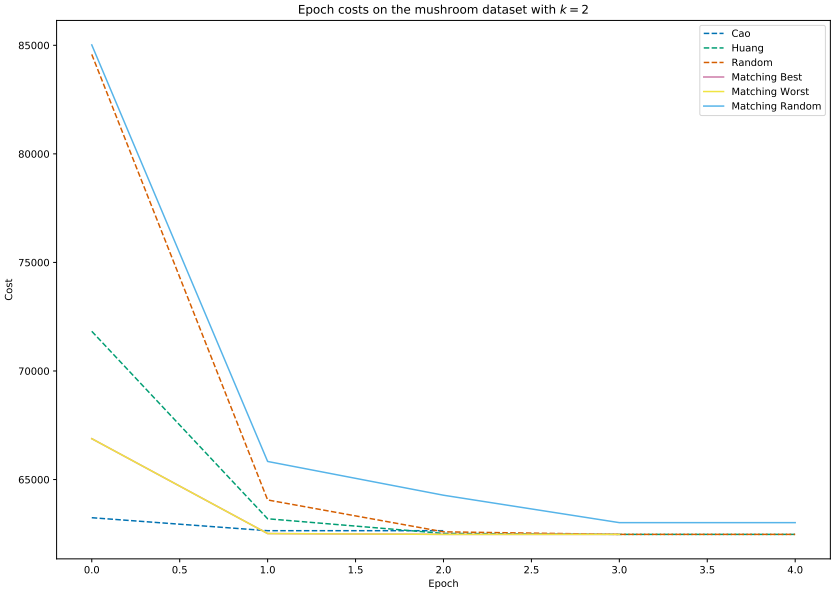
Table 4: Summative metric results for the zoo animal dataset with $k = 7$.

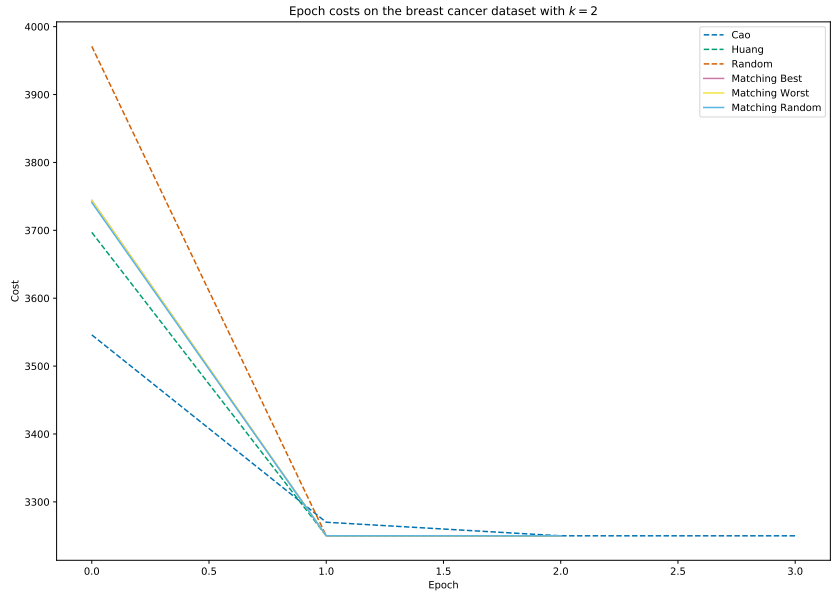
6.2.2 Epoch costs

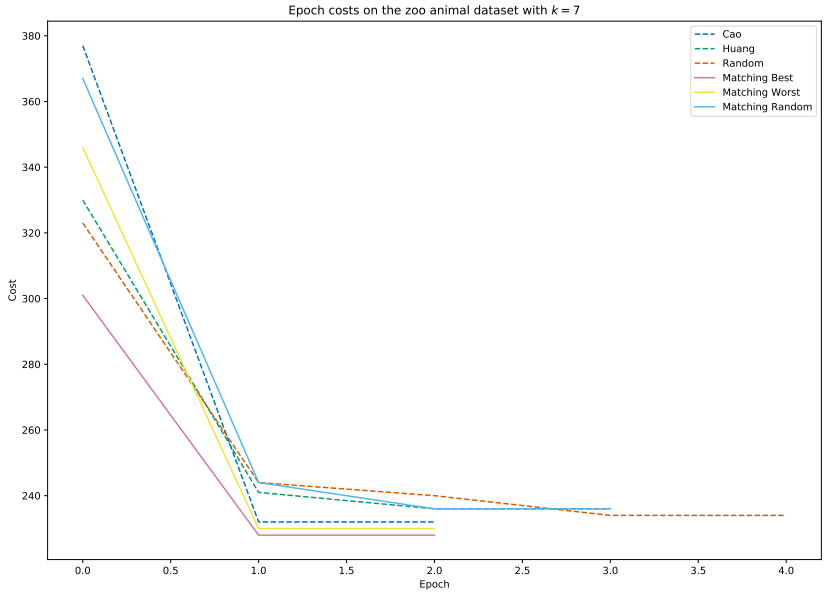
The epoch-cost plots in this section were created by setting an initial seed for each initialisation method and then running the k-modes algorithm 25 times. Of these runs, the best set of costs is then chosen by their final cost and plotted.

Note that in each figure, dotted lines indicate the established initialisation methods whilst solid lines are used for the proposed method.




Epoch costs on the mushroom dataset with $k = 2$
Cao
Huang
Random
Matching Best
Matching Worst
Matching Random
Cost
Epoch
85000
80000
75000
70000
65000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Epoch costs on the breast cancer dataset with $k = 2$
Cao
Huang
Random
Matching Best
Matching Worst
Matching Random
Cost
Epoch
4000
3900
3800
3700
3600
3500
3400
3300
0.0
0.5
1.0
1.5
2.0
2.5
3.0


Epoch costs on the zoo animal dataset with $k = 7$
Cao
Huang
Random
Matching Best
Matching Worst
Matching Random
Cost
Epoch
380
360
340
320
300
280
260
240
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

References

[1] F. Cao, J. Liang, and L. Bai. A new initialization method for categorical data clustering. *Expert Systems with Applications*, 36:10223–10228, 2009.

[2] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, September 1998.

[3] G. H. Lincoff. The Audubon Society field guide to North American mushrooms. R, 1981.

[4] R. S. Michalski and R. L. Chilausky. Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, 4(2), 1980.

[5] M. K. Ng, M. J. Li, Z. Huang, and Z. He. On the impact of dissimilarity measure in k-modes clustering algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):503–507, March 2007.

[6] Abass Olaode, Golshah Naghdy, and Catherine Todd. Unsupervised image classification by Probabilistic Latent Semantic Analysis for the annotation of images. In *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications*, 11 2014.