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

Henry Wilde, Vincent Knight and Jonathan Gillard

November 6, 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 Introduction

- What is clustering?
- What is the k-means paradigm?
- What is categorical data and how is it clustered?

1.1 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*.

- 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 $\overline{Z} = \left\{z^{(1)}, \ldots, z^{(k)}\right\}$.

As is discussed above, the notion of distance is lost in categorical space, and especially when that space is even partly nominal. Definition 1.1 describes a simple dissimilarity measure between categorical data points.

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:

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

In other words, the dissimilarity between two points is the number of attributes where their values are not the same. A proof that (1) is a valid distance metric is given as an appendix.

With this metric defined, the notion of a representative point within a cluster can be addressed. When clustering numeric data, a centroid of a cluster is taken to be the average of the points within the cluster so as to summarise the information contained within that cluster. With categorical data, however, a frequency approach is used. This follows from the concept of dissimilarity where the point that best represents (i.e. is closest to) those in a cluster is one with the most frequent attribute values of the points in the cluster. As such, a representative point of a cluster is often called a mode. The following definitions and theorem formally define such a representative point and a means of finding them.

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 $n\left(a_s^{(j)}\right)$ 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 $s = 1, \ldots, d_j$:

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

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

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 $n(u_j) \geq n\left(a_s^{(j)}\right)$ 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 [11].

Theorem 1 defines the process by which representatives are updated in k-modes (see Algorithm 3), and so the final component from the k-means paradigm to be configured is the objective (cost) function. This function is defined in Definition 1.4, and following that a practical statement of the k-modes algorithm is given in Algorithm 1 as set out in [11].

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:

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

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
    $Z_{l^*} \leftarrow$ SelectClosest $\left(X^{(i)}\right)$
    $Z_{l^*} \leftarrow Z_{l^*} \cup \left\{X^{(i)}\right\}$
    Update $\left(z^{(l^*)}\right)$
end
repeat
    for $X^{(i)} \in \boldsymbol{X}$ do
        Let $Z_l$ be the cluster $X^{(i)}$ currently belongs to
        $Z_{l^*} \leftarrow$ SelectClosest $\left(X^{(i)}\right)$
        if $l \neq l^*$ then
            $Z_l \leftarrow Z_l \setminus \left\{X^{(i)}\right\}$ and $Z_{l^*} \leftarrow Z_{l^*} \cup \left\{X^{(i)}\right\}$
            Update $\left(z^{(l)}\right)$ and Update $\left(z^{(l^*)}\right)$
        end
    end
until No point changes cluster
```

```
Algorithm 2: SelectClosest

Input: a data point $X^{(i)}$, a set of current clusters $mathcalZ$ and their modes $\overline{Z}$
Output: the cluster whose mode is closest to the data point $Z_{l^*}$
Select $z^{l^*} \in \overline{Z}$ that minimises: $d\left(X^{(i)}, z_{l^*}\right)$
Find their associated cluster $Z_{l^*}$
```

```
Algorithm 3: 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 $D(Z_l, z)$
$z^{(l)} \leftarrow z$
```

1.2 Initialisation processes

All of the methods within the k-means paradigm are heuristics and as such their performance is dependent on the quality of their initial solution. The quality of the initial centroids for a particular dataset is affected by two components: the metric attached to the attribute space and the process by which they are chosen.

Following the seminal k-modes papers [9, 10, 11], a number of alternative dissimilarity measures have been implemented to improve on the simple matching dissimilarity defined in (1). The main drawback of this measure is that it often produces clusters with low intra-cluster similarity [18] and does not take into account any relationships between attributes or their categories. Other measures have been designed to be used in a specific context where such relationships may be considered [4, 22, 23].

1.2.1 Huang's method

In the standard form of the k-modes algorithm, the k initial modes are chosen at random from $\mathcal{X}$. Below is an alternative method of selecting these modes that forces some diversity between them, as described in [11]. 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.

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 3, 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 4.

Algorithm 4: Huang's method

```
Input: a dataset $\mathcal{X} \subset \mathcal{A}$, a number of modes to find $k$
Output: a set of $k$ initial modes $\overline{Z}$
$\overline{Z} \leftarrow \emptyset$
$\widehat{Z} \leftarrow$ SamplePotentialModes $(\mathcal{X})$
for $\hat{z} \in \widehat{Z}$ do
    Select $X^{(i^*)} \in \mathcal{X} \setminus \overline{Z}$ that minimises $d\left(X^{(i)}, \hat{z}\right)$
    $\overline{Z} \leftarrow \overline{Z} \cup \left\{X^{(i^*)}\right\}$
end
```

Algorithm 5: SamplePotentialModes

```
Input: a dataset $\mathcal{X} \subset \mathcal{A}$, a number of modes to find $k$
Output: a set of $k$ potential modes $\widehat{Z}$
$\widehat{Z} \leftarrow \emptyset$
for $j = 1, \dots, m$ do
    for $s = 1, \dots, d_j$ do
        Calculate $\frac{n\left(a_s^{(j)}\right)}{N}$
    end
end
while $\left|\widehat{Z}\right| < k$ do
    Create an empty $m$-tuple $\hat{z}^{(l)}$
    for $j = 1, \dots, m$ do
        Sample $a_{s^*}^{(j)}$ from $A_j$ with respect to the relative frequencies of $A_j$
        $\hat{z}_j^{(l)} \leftarrow a_{s^*}^{(j)}$
    end
    $\widehat{Z} \leftarrow \widehat{Z} \cup \left\{\hat{z}^{(l)}\right\}$
end
```

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.

1.2.2 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 1.5. Consider a data set $\mathcal{X}$ with attribute set $\mathcal{A} = \{A_1, \dots, A_m\}$. Then the *average density* of any point $X_i \in \mathcal{X}$ with respect to $\mathcal{A}$ is defined [3] as:

$$\text{Dens}\left(X^{(i)}\right) = \frac{\sum_{j=1}^{m} \text{Dens}_j\left(X^{(i)}\right)}{m} \quad \text{where} \quad \text{Dens}_j\left(X^{(i)}\right) = \frac{\left|\left\{X^{(t)} \in \mathcal{X} : x_j^{(i)} = x_j^{(t)}\right\}\right|}{N} \tag{5}$$

Observe that:

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

And so, an alternative definition for (5) can be derived:

$$\begin{aligned}
\text{Dens}\left(X^{(i)}\right) &= \frac{1}{mN}\sum_{j=1}^{m}\sum_{t=1}^{N}\left(1 - \delta\left(x_j^{(i)}, x_j^{(t)}\right)\right) \\
&= \frac{1}{mN}\sum_{j=1}^{m}\sum_{t=1}^{N} 1 - \frac{1}{mN}\sum_{j=1}^{m}\sum_{t=1}^{N}\delta\left(x_j^{(i)}, x_j^{(t)}\right) \\
&= \frac{mN}{mN} - \frac{1}{mN}\sum_{t=1}^{N} d\left(X^{(i)}, X^{(t)}\right) \\
&= 1 - \frac{1}{mN} D\left(\mathcal{X}, X^{(i)}\right)
\end{aligned} \tag{6}$$

Remark. It is worth noting that for all $X^{(i)} \in \mathcal{X}$ it follows that $\frac{1}{N} \leq \text{Dens}\left(X^{(i)}\right) \leq 1$, since:

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

Remark. With this alternative definition, we see - since m and N are fixed positive integers - that Dens$(X^{(i)})$ is maximised when $D(\mathcal{X}, X^{(i)})$ is minimised. Then by Theorem **??** we have that such an $X^{(i)}$ with maximal average density in $\mathcal{X}$ with respect to $\mathcal{A}$ is, in fact, a mode of $\mathcal{X}$. This

Algorithm 6: Cao's method

```
Input: a dataset $\mathcal{X}$, a number of modes to find $k$
Output: a set of $k$ initial modes $\overline{Z}$
$\overline{Z} \leftarrow \emptyset$
for $X^{(i)} \in \mathcal{X}$ do
    Calculate Dens $\left(X^{(i)}\right)$
end
Select $1 \leq i_1 \leq N$ which maximises Dens $\left(X^{(i)}\right)$
$\overline{Z} \leftarrow \overline{Z} \cup \left\{X^{(i_1)}\right\}$
while $\left|\overline{Z}\right| < k$ do
    Select $X^{(i^*)} \notin \overline{Z}$ which maximises $\min_{z^{(l)} \in \overline{Z}} \left\{\text{Dens}\left(X^{(i)}\right) \times d\left(X^i, z^{(l)}\right)\right\}$
    $\overline{Z} \leftarrow \overline{Z} \cup \left\{X^{(i^*)}\right\}$
end
```

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.

2 Matching games and the proposed method

Both of the initialisation methods described in Section **??** have a greedy component. Cao's method essentially chooses the densest point that has not already been chosen whilst forcing separation between the set of initial modes. In the case of Huang's, however, the greediness only comes at the end of the method, after the set of potential modes has been found by random sampling. In any practical implementation of this method, the order in which a set of potential modes is iterated over has no guarantee of consistency. The same is true for any arbitrary tie breaks. The result of this is that the initial set of modes that the method returns is altered since the next initial mode is chosen by the next locally optimal choice.

The initialisation method proposed in this work aims to extend Huang's method to be order-invariant in the final allocation - thereby eliminating its greedy component - and to provide a more intuitive starting point for the k-modes algorithm. This is done by constructing and solving a matching game between the set of potential modes and some subset of the data.

In general, matching games are defined by two sets (parties) of players (often termed suitors and reviewers) in which each player creates a preference list of at least some of the players in the other party. The objective then is to find a mapping between the two sets of players such that no pair of players is (rationally) unhappy with their matching. Such a mapping is considered stable. Algorithms to find stable matchings to instances of matching games are often structured to be party-oriented and aim to maximise some form of social or party-based optimality [6, 7, 13].

One of the simplest matching games models the Stable Marriage Problem (SM). In this game the sets of players must be of equal size and rank each other strictly (i.e. no ties allowed) and entirely. An algorithm was presented in the seminal work by D. Gale and L. Shapley [7] that 'solves' any instance of SM by finding for it a unique, suitor-optimal, stable matching. This kind of game is not considered in this work as it effectively reduces down to Huang's method. This can be seen as follows. Note that the concept of preference between points in an attribute space is synonymous with similarity. Thus, when constructing the game, each potential mode gets to pick the data point it is closest to but that has not already been picked. Then, up to an arbitrary breaking of any ties in the preference lists, each potential mode is assigned to its chosen data point.

A number of issues arise from this particular model other than it reducing to Huang's method. For instance:

- Ties are common when using the distance metric defined in (1).
- There is no intuitively concrete way of constructing sets of players or their preference lists.

Allowing for ties is a simple extension to SM but the notion of stability becomes tiered [15]. In each case of stability, if such a matching exists, then a polynomial-time algorithm will find one that is optimal for one set of players. However, there is no guarantee that such a level-of-stable matching exists and even in that case, the notion of party-optimality is lost [5]. Therefore this extension is not considered here where a stable solution to the game is required, and is preferably party-optimal.

Further to allowing ties, how preference lists are constructed is a point of interest in many applications of matching games [12]. Often this is a contextual problem and may be addressed in a number of ways. As in this case, where similarity and preference are considered equivalent, a bespoke distance metric may be used. Though not relevant to this work, if the game forms part of a larger, long-standing or otherwise complex model, introducing flexibility in preferences [1, 16] or estimating them *ad hoc* [20] may be helpful to obtain meaningful matchings.

Another method used to construct preference lists is to discount the preference lists presented by players. For instance, where acceptability of another player is the only criterion, binary preferences (i.e. incomplete preference lists with ties) can create games that are invulnerable to manipulative players' strategies [2]. This approach can be adapted to cater for larger games, such as student-school allocation. In this case, each student submits a set of acceptable schools and the schools form strict rankings of the students. The result of this is a simpler game (in the practical sense) and a reduction in the set of possible stable matchings [8].

As this particular case has no interactive element, and must guarantee a stable matching (ideally with optimality), none of these extensions are used in the proposed method. Instead, so as to keep the method as simple as possible within these constraints, the game used will model an instance of the Hospital-Resident Assignment Problem (HR) which was presented with SM as a practical solution to the problem that gives it its namesake [7].

Like SM, there exists an algorithm that can provide a unique, party-optimal, stable matching to any instance of HR. The resident-optimal algorithm is presented in Algorithm 7 and is adapted from the original to take advantage of the structure of the game [21]. The game used to model HR, its matchings, and its notion of stability are defined in Definitions 2.1 - 2.3. A summary of these definitions in the context of the proposed k-modes initialisation is given in Table 1.

Definition 2.1. Consider two distinct sets R, H and refer to them residents and hospitals. Each $h \in H$ has a capacity $c_h \in \mathbb{N}$ associated with them. Each player $r \in R$ and $h \in H$ has associated with it a strict preference list of the other set's elements such that:

- Each $r \in R$ ranks a non-empty subset of H, denoted by $f(r)$.

- Each $h \in H$ ranks all and only those residents that have ranked it, i.e. the preference list of h, denoted $g(h)$, is a permutation of the set $\{r \in R \mid h \in f(r)\}$. If no such residents exist, h is removed from H.

This construction of residents, hospitals, capacities and preference lists is called a *game* and is denoted by (R, H).

Definition 2.2. Consider a game (R, H). A *matching* M is any mapping between R and H. If a pair $(r, h) \in R \times H$ are matched in M then this relationship is denoted $M(r) = h$ and $r \in M^{-1}(h)$.

A matching is only considered *valid* if all of the following hold for all $r \in R, h \in H$:

- If r is matched then $M(r) \in f(r)$.
- If h has at least one match then $M^{-1}(h) \subseteq g(h)$.
- h is not over-subscribed, i.e. $\left|M^{-1}(h)\right| \leq c_h$.

A valid matching is considered *stable* if it does not contain any blocking pairs.

Definition 2.3. Consider a game (R, H). Then a pair $(r, h) \in R \times H$ is said to *block* a matching M if all of the following hold:

- There is mutual preference, i.e. $r \in g(h)$ and $h \in f(r)$.
- Either r is unmatched or they prefer h to $M(r)$.
- Either h is under-subscribed or h prefers r to at least one resident in $M^{-1}(h)$.

Object in k-modes initialisation	Object in a matching game
Similarity between two points $U, V \in \mathcal{A}$, $m - d(U, V)$	Respective position in preference lists f, g
Potential modes, $\widehat{Z}$	Residents, R
Data points closest to $\widehat{Z}$, $\mathcal{X}' \subset \mathcal{X}$	Hospitals, H
A mode $\hat{z} \in \widehat{Z}$ being replaced by a point $X \in \mathcal{X}'$	A pair in some matching M

Table 1: A summary of the relationships between the components of the initialisation for k-modes and those in a matching game (R, H).

3 Experimental results

To give comparative results on the quality of the initialisation processes defined in Sections **??**, **??** & **??**, 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

Algorithm 7: The hospital-resident algorithm (resident-optimal)

```
Input: a set of residents $R$, a set of hospitals $H$, a set of hospital capacities $C$, two
       preference list functions $f, g$
Output: a stable, resident-optimal mapping $M$ between $R$ and $H$
for $h \in H$ do
    $M^{-1}(h) \leftarrow \emptyset$
end
while There exists any unmatched $r \in R$ with a non-empty preference list do
    Take any such resident $r$ and their most preferred hospital $h$
    MatchPair($s, h$)
    if $\left|M^{-1}(h)\right| > c_h$ then
        Find their worst match $r' \in M^{-1}(h)$
        UnmatchPair($r', h$)
    end
    if $\left|M^{-1}(h)\right| = c_h$ then
        Find their worst match $r' \in M^{-1}(h)$
        for each successor $s \in g(h)$ to $r'$ do
            DeletePair($s, h$)
        end
    end
end
```

Algorithm 8: MatchPair

```
Input: a resident $r$, a hospital $h$, a matching $M$
Output: an updated matching $M$
$M^{-1}(h) \leftarrow M^{-1}(h) \cup \{r\}$
```

Algorithm 9: UnmatchPair

```
Input: a resident $r$, a hospital $h$, a matching $M$
Output: an updated matching $M$
$M^{-1}(h) \leftarrow M^{-1}(h) \setminus \{r\}$
```

Algorithm 10: DeletePair

```
Input: a resident $r$, a hospital $h$
Output: updated preference lists
$f(r) \leftarrow f(r) \setminus \{h\}$
$g(h) \leftarrow g(h) \setminus \{r\}$
```

Algorithm 11: The proposed initialisation method

```
Input: a dataset $\mathcal{X} \subset \mathcal{A}$, a number of modes to find $k$
Output: a set of $k$ initial modes $\overline{Z}$
$\overline{Z} \leftarrow \emptyset$
$H \leftarrow \emptyset$
$R \leftarrow \textsc{SamplePotentialModes}\left(\mathcal{X}\right)$
for $r \in R$ do
    Find the set of $k$ data points $H_r \subset \mathcal{X}$ that are the least dissimilar to $r$
    Arrange $H_r$ into descending order of similarity with respect to $r$, denoted by $H_r^*$
    $H \leftarrow H \cup H_r$
    $f(r) \leftarrow H_r^*$
end
for $h \in H$ do
    $c_h \leftarrow 1$
    Sort $R$ into descending order of similarity with respect to $h$, denoted by $R^*$
    $g(h) \leftarrow R^*$
end
Solve the matching game defined by $(R, H)$ to obtain a matching $M$
for $r \in R$ do
    $\overline{Z} \leftarrow \overline{Z} \cup \{M(r)\}$
end
```

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 [3, 11, 19]. 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 3.1. Let $\mathbf{X}$ be a dataset and consider a clustering of $\mathbf{X}$ into k parts, denoted by $C = \{C_1, \dots, C_k\}$. 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.

3.1 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 [17]. 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 [14]. 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`.

3.2 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.

3.2.1 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 sum-

	Initial cost	Final cost	Silhouette	No. iterations	Time
Cao	2220.48 (41.755)	1423.28 (67.357)	-0.01 (0.001)	4.36 (0.898)	0.69 (0.068)
Huang	1592.88 (74.713)	1448.66 (62.399)	-0.00 (0.001)	4.20 (1.325)	0.43 (0.099)
Matching	1586.38 (57.090)	1327.74 (35.361)	-0.01 (0.002)	4.28 (1.031)	0.23 (0.016)

Table 2: Summative metric results for the soybean dataset with $k = 15$.

	Initial cost	Final cost	Silhouette	No. iterations	Time
Cao	45585.18 (1337.353)	43100.26 (1437.688)	-0.00 (0.000)	1.94 (0.314)	1.41 (0.104)
Huang	41720.20 (2519.647)	38612.64 (2086.246)	-0.00 (0.000)	3.14 (1.471)	1.86 (0.692)
Matching	41471.80 (2374.049)	39000.18 (2320.722)	-0.00 (0.000)	3.28 (1.230)	1.31 (0.411)

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

	Initial cost	Final cost	Silhouette	No. iterations	Time
Cao	3519.44 (52.233)	3179.76 (50.833)	-0.00 (0.000)	2.46 (0.542)	0.12 (0.014)
Huang	3369.68 (126.396)	3327.34 (141.856)	-0.00 (0.000)	1.50 (0.580)	0.09 (0.017)
Matching	3367.64 (112.420)	3275.72 (134.650)	-0.00 (0.000)	1.68 (0.621)	0.08 (0.012)

Table 4: Summative metric results for the breast cancer dataset with $k = 2$.

	Initial cost	Final cost	Silhouette	No. iterations	Time
Cao	500.28 (25.513)	257.76 (14.623)	-0.01 (0.002)	3.08 (0.778)	0.07 (0.009)
Huang	274.32 (28.838)	251.96 (17.393)	-0.01 (0.002)	2.44 (0.884)	0.04 (0.010)
Matching	280.44 (27.014)	247.42 (17.012)	-0.01 (0.002)	2.58 (0.731)	0.03 (0.004)

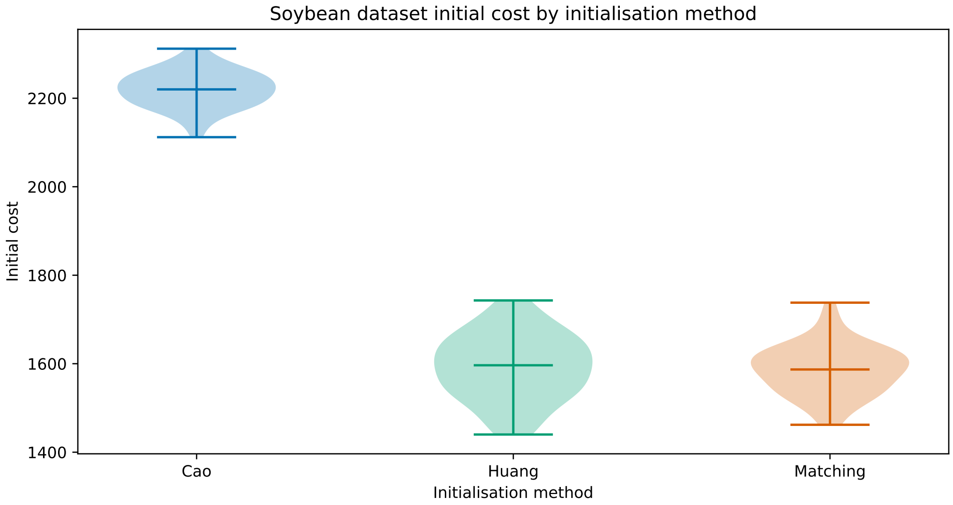
Table 5: Summative metric results for the zoo animal dataset with $k = 7$.

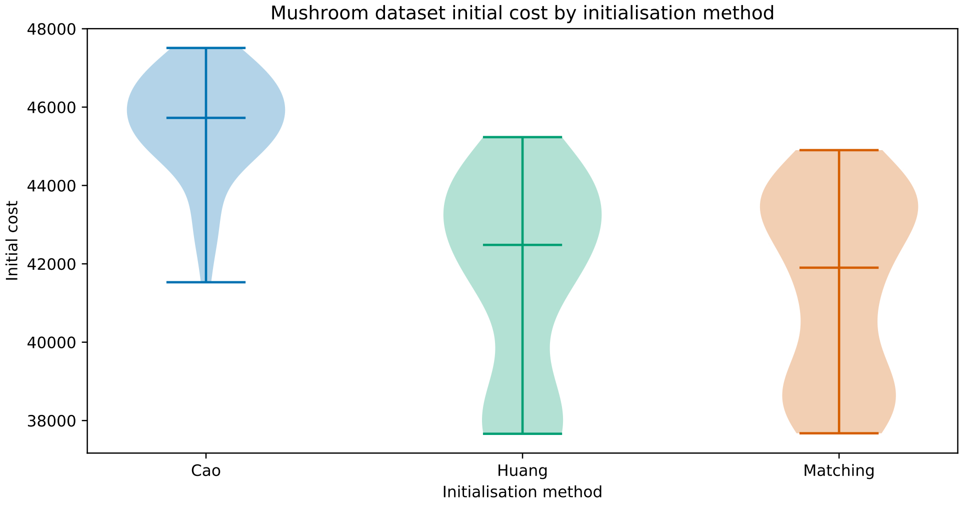
marised below in Tables **??** - **??** by their mean and median values, and their standard deviation over the 25 runs.

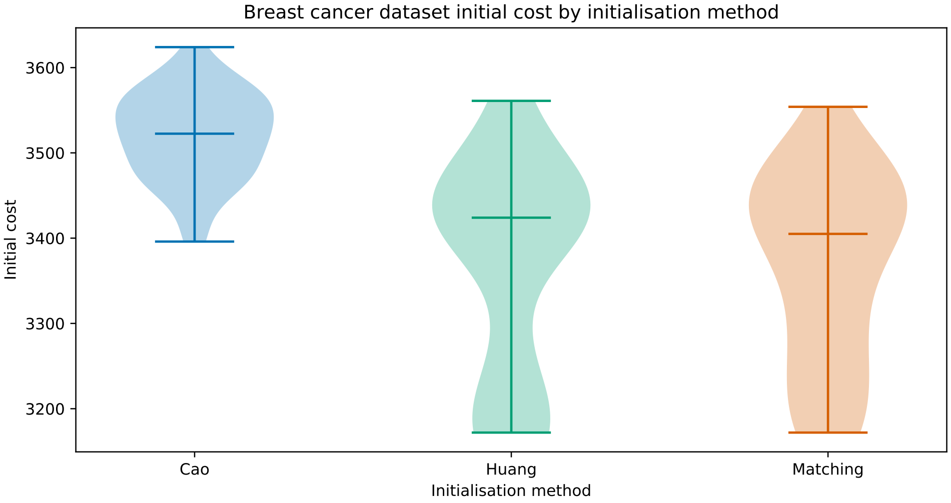
3.2.2 Epoch costs

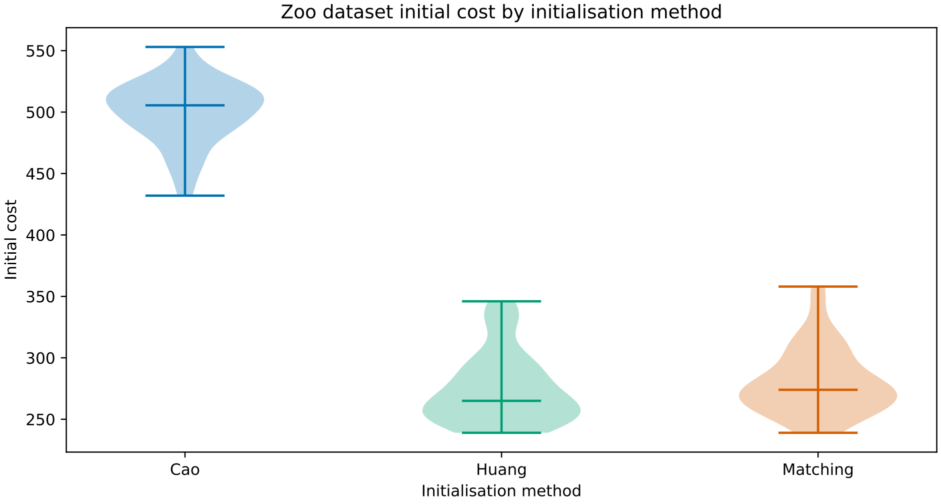
The epoch-cost plots in this section were created by setting an initial seed for each initialisation method and then running the k-modes algorithm 25 times. Of these runs, the best set of costs is then chosen by their final cost and plotted.

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


Soybean dataset initial cost by initialisation method
Initial cost
2200
2000
1800
1600
1400
Cao
Huang
Matching
Initialisation method


Mushroom dataset initial cost by initialisation method
Initial cost
48000
46000
44000
42000
40000
38000
Cao
Huang
Matching
Initialisation method


Breast cancer dataset initial cost by initialisation method
Initial cost
3600
3500
3400
3300
3200
Cao
Huang
Matching
Initialisation method


Zoo dataset initial cost by initialisation method
Initial cost
550
500
450
400
350
300
250
Cao
Huang
Matching
Initialisation method

References

[1] Nikhil Agarwal. Policy analysis in matching markets. *American Economic Review*, 107(5):246–50, May 2017.

[2] Anna Bogomolnaia and Herve Moulin. Random matching under dichotomous preferences. *Econometrica*, 72(1):257–279, 2004.

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

[4] F. Cao, J. Liang, D. Li, L. Bai, and C. Dang. A dissimilarity measure for the k-modes clustering algorithm. *Knowledge-Based Systems*, 26:120–127, 2012.

[5] Aytek Erdil and Haluk Ergin. Two-sided matching with indifferences. *Journal of Economic Theory*, 171:268 – 292, 2017.

[6] Tomoko Fuku, Akira Namatame, and Taisei Kaizoji. *Collective Efficiency in Two-Sided Matching*, pages 115–126. 1 2006.

[7] D. Gale and L. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[8] Guillaume Haeringer and Vincent Iehlé. Two-sided matching with one-sided preferences. 06 2014.

[9] Z. Huang. Clustering large data sets with mixed numeric and categorical values. In *The First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 21–34, 1997.

[10] Z. Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 1–8, 1997.

[11] 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.

[12] K. Iwama and S. Miyazaki. A survey of the stable marriage problem and its variants. In *International Conference on Informatics Education and Research for Knowledge-Circulating Society*, pages 131–136, 1 2008.

[13] Augustine Kwanashie, Robert W. Irving, David F. Manlove, and Colin T. S. Sng. Profile-based optimal matchings in the student/project allocation problem. In *Combinatorial Algorithms*, pages 213–225, 2015.

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

[15] David Manlove. Stable marriage with ties and unacceptable partners. 2 1999.

[16] Konrad Menzel. Large matching markets as two-sided demand systems. *Econometrica*, 83(3):897–941, 2015.

[17] 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.

[18] 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, 3 2007.

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

[20] Baharak Rastegari, Paul Goldberg, and David Manlove. Preference elicitation in matching markets via interviews: A study of offline benchmarks. In *International Conference on Autonomous Agenys & Multiagent Systems*, pages 1393–1394, 2 2016.

[21] A. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy*, 92(6):991–1016, 1984.

[22] H. Yu, Z. Zhang, Z. Zhu, W. Xiong, and G. Zhang. Nominal data similarity: A hierarchical measure. *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2018.

[23] Hongfang Zhou, Yihui Zhang, and Yibin Liu. An improved k-modes clustering algorithm based on intra-cluster and inter-cluster dissimilarity measure. In *2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017)*, 2016.

A Appendix

A.1 Proof that simple matching dissimilarity is a metric

Let $\mathcal{A}$ be a categorical attribute space and let the dissimilarity function, $d : \mathcal{A}^2 \to \mathbb{R}$, be defined as in (1). Then d is a metric such that for all $x, y, z \in \mathcal{A}$:

(i) $d(x, y) \geq 0$

(ii) $d(x, y) = 0 \iff x = y$

(iii) $d(x, y) = d(y, x)$

(iv) $d(x, y) + d(y, z) \geq d(x, z)$

Proof. Let $\mathcal{A}$ be a categorical attribute space and consider any $x, y, z \in \mathcal{A}$.

(i) If $x = y$, then $x_j = y_j$ for all $j = 1, \ldots, m$. Then it immediately follows that $d(x, y) = 0$. Otherwise, $\delta(x_j, y_j) = 1$ for at least one $j = 1, \ldots, m$. Therefore, $d(x, y) \geq 1$, as required.

(ii) As above, if $x = y$ then $d(x, y) = 0$. Now consider any $x, y \in \mathcal{A}$ such that $d(x, y) = 0$. Then $d(x, y) = 0 \iff \delta(x_j, y_j) = 0$ for all $j = 1, \ldots, m \iff x_j = y_j$ for all $j = 1, \ldots, m \iff x = y$, as required.

(iii) This follows from the commutativity of equality in the definition of δ in (1):

$$d(x, y) = \sum_{j=1}^{m} \delta\left(x_j,\ y_j\right) = \sum_{j=1}^{m} \delta\left(y_j,\ x_j\right) = d(y, x)$$

(iv) Without ambiguity, let x, y, z each be represented as a set of its elements. Then an alternative form for d may be considered where $d(x, y) = m - |x \cap y|$. With this, it is sufficient to show:

$$(m - |x \cap y|) + (m - |y \cap z|) \geq m - |x \cap z|$$

That is, $|x \cap y| - |x \cap z| + |y \cap z| \leq m$. So,

$$\begin{aligned} |x \cap y| - |x \cap z| + |y \cap z| &\leq |x \cap (y \setminus z)| + |y \cap z| \\ &\leq |y \setminus z| + |y \cap z| \\ &= |y| = m \end{aligned}$$

Therefore, d satisfies conditions (i) - (iv) and is a metric. □