A novel initialisation based on hospital-resident assignment for the k-modes algorithm

Henry Wilde, Vincent Knight and Jonathan Gillard

February 5, 2020

Abstract

This paper presents a new way of selecting an initial solution for the k-modes algorithm that allows for a notion of mathematical fairness and a leverage of the data that the common initialisations from literature do not. The method, which utilises the Hospital-Resident Assignment Problem to find the set of initial cluster centroids, is compared with the current initialisations on both benchmark datasets and a body of newly generated artificial datasets. Based on this analysis, the proposed method is shown to outperform the other initialisations in the majority of cases, especially when the number of clusters is optimised. In addition, we find that our method outperforms the leading established method specifically for low-density data.

1 Introduction

This work focusses on k-modes clustering — an extension to k-means that permits the sensible clustering of categorical (i.e. ordinal, nominal or otherwise discrete) data as set out in the seminal works by Huang [**?** **?** **?**]. In particular, the interest of this paper is in how the performance of the k-modes algorithm is affected by the quality of its initial solution. The initialisation method proposed in this work extends the method presented by Huang [**?**] by using results from game theory to ensure mathematical fairness and to lever the full learning opportunities presented by the data being clustered. In doing so, it is demonstrated that the proposed method is able to outperform both of the established initialisations for k-modes. The paper is structured as follows:

- Section 1 introduces the k-modes algorithm and its established initialisation methods.
- Section 2 provides a brief overview of matching games and their variants before a statement of the proposed initialisation.
- Section 3 presents analyses of the initialisations on benchmark and new, artificial datasets.
- Section 4 concludes the paper.

1.1 The k-modes algorithm

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

- Let $\mathcal{A} := A_1 \times \cdots \times A_m$ denote the *attribute space*. In this work, only categorical attributes are considered, 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 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 $\mathcal{X} \subset \mathcal{A}$ 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 *mode* (see Definition 1.2) which is denoted by $z^{(l)} = \left(z_1^{(l)}, \ldots, z_m^{(l)}\right) \in \mathcal{A}$. These points are also referred to as *representative points* or *centroids*. The set of all current cluster modes is denoted as $\overline{Z} = \left\{z^{(1)}, \ldots, z^{(k)}\right\}$.

Definition 1.1 describes a dissimilarity measure between categorical data points.

Definition 1.1. Let $\mathcal{X} \subset \mathcal{A}$ 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:

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

With this metric, the notion of a representative point of a cluster is addressed. With numeric data and k-means, such a point is taken to be the mean of the points within the cluster. With categorical data, however, the mode is used as the measure for central tendency. This follows from the concept of dissimilarity in that 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. The following definitions and theorem formalise this and a method to find such a point.

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 [**?**].

Theorem 1 defines the process by which cluster modes 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 [**?**].

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 $\mathcal{Z}$ 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

The standard selection method to initialise k-modes is to randomly sample k distinct points in the dataset. In all cases, the initial modes must be points in the dataset to ensure that there are no empty clusters in the first iteration of the algorithm. The remainder of this section describes two well-established initialisation methods that aim to preemptively lever the structure of the data at hand.

1.2.1 Huang's method

Amongst the original works by Huang, an alternative initialisation method was presented that selects modes by distributing frequently occurring values from the attribute space among k potential modes [**?**]. The process, denoted as Huang's method, is described in full in Algorithm 4. Huang's method considers a set of potential modes, $\widehat{Z} \subset \mathcal{A}$, that is then replaced by the actual set of initial modes, $\overline{Z} \subset \mathcal{X}$. The statement of how the set of potential modes are formed is ambiguous in the original paper — as is alluded to in [**?**]. Here, as is done in practical implementations of k-modes, this has been interpreted as being done via a weighted random sample (see Algorithm 5).

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 $j = 1, \ldots, m$ **do**
  for $s = 1, \ldots, d_j$ **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 $j = 1, \ldots, m$ **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

1.2.2 Cao's method

The second initialisation process that is widely used with k-modes is known as Cao's method [**?**]. This method selects the initial modes according to their density in the dataset whilst forcing dissimilarity between them. Definition 1.5 formalises the concept of density and its relationship to relative frequency. The method, which is described in Algorithm 6, is deterministic — unlike Huang's method which relies on random sampling.

Definition 1.5. Consider a dataset $\mathcal{X} \subset \mathcal{A} = \{A_1, \ldots, A_m\}$. Then the *average density* of any point $X_i \in \mathcal{X}$ with respect to $\mathcal{A}$ is defined [**?**] 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:

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

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

2 Matching games and the proposed method

Both of the initialisation methods described in Section 1.2 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, when the set of potential modes is replaced by a set of instances in the dataset. Specifically, this means that in any practical implementation of this method the order in which a set of potential modes is iterated over can affect the set of initial modes. Thus, there is no guarantee of consistency.

The initialisation proposed in this work extends Huang's method to be order-invariant in the final allocation — thereby eliminating its greedy component — and provides 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 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 'stable' mapping between the two sets of players such that no pair of players is (rationally) unhappy with their matching. Algorithms to 'solve' — i.e. 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 [**? ? ? ? ? ? ?**].

The particular constraints of this case — where the k potential modes must be allocated to a nearby unique data point — mirror those of the so-called Hospital-Resident Assignment Problem (HR). This problem gets its name from the real-world problem of fairly allocating medical students to hospital posts. A resident-optimal algorithm for solving HR was presented in [**?**] and was adapted in [**?**] to take advantage of the structure of the game. This adapted algorithm is given in Algorithm 7. A practical implementation of this algorithm has been implemented in Python as part of the `matching` library [**?**] and is used in the implementation of the proposed method for Section 3.

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 before a formal statement of the proposed method in Algorithm 11.

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
Potential modes	The set of residents
Data points closest to potential modes	The set of hospitals
Similarity between a potential mode and a point	Respective position in each other's preference lists
The data point to replace a potential mode	A pair in a matching

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

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 $\overline{Z}$
$\overline{Z} \leftarrow \emptyset$
$H \leftarrow \emptyset$
$R \leftarrow \textsc{SamplePotentialModes}\,(\mathcal{X})$
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

3 Experimental results

To give comparative results on the quality of the initialisation processes considered in this work, four well-known, categorical, labelled datasets — breast cancer, mushroom, nursery, and soybean (large) — will be clustered by the k-modes algorithm with each of the initialisation processes. These datasets have been chosen to fall in line with the established literature, and for their relative sizes and complexities. Each dataset is openly available under the UCI Machine Learning Repository [**?**], and their characteristics are summarised in Table 2. For the purposes of this analysis, incomplete instances (i.e. where data is missing) are excluded and the remaining dataset characteristics are reported as 'adjusted'.

	N	m	No. classes	Missing values	Adjusted N	Adjusted no. classes	No. clusters found
Breast cancer	699	10	2	True	683	2	8
Mushroom	8124	22	2	True	5644	2	17
Nursery	12960	8	5	False	12960	5	23
Soybean	307	35	19	True	266	15	8

Table 2: A summary of the benchmark datasets.

All of the source code used to produce the results and data in this analysis — including the datasets investigated in Section 3.3 — are archived at DOI. In addition to this, the implementation of the k-modes algorithm and its initialisations is available under DOI 10.5281/zenodo.3638035.

This analysis does not consider evaluative metrics related to classification such as accuracy, recall or precision as is commonly done [**?** **?** **?** **?** **?** **?** **?** **?**]. Instead, only internal measures are considered such as the cost function defined in (4). This metric is label-invariant and its values are comparable across the different initialisation methods. Furthermore, the effect of each initialisation method on the initial and final clusterings can be captured with the cost function. An additional, and often useful, metric is the silhouette coefficient. This measures the ratio between the intra-cluster cohesion and inter-cluster separation of a particular clustering. Therefore, it could be used in a similar way to reveal the effect of each initialisation method at the beginning and end of a run of k-modes. Unfortunately, this metric loses its intuition under the distance measure employed here and is omitted. The remaining performance measures used are the number of iterations for the k-modes algorithm to terminate and the time taken to terminate in seconds.

The final piece of information required in this analysis is a choice for k for each dataset. An immediate choice is the number of classes that are present in a dataset but this is not necessarily an appropriate choice since the classes may not be representative of true clusters [**?**]. However, this analysis will consider this case as there may be practical reasons to limit the value of k. The other strategy for choosing k considered in this work uses the knee point detection algorithm introduced in [**?**]. This strategy was chosen over other popular methods such as the 'elbow' method as its results are definitive.

The knee point detection algorithm was employed over values of k from 2 up to $\lfloor\sqrt{N}\rfloor$ for each dataset. The number of clusters determined by this strategy is reported in the final column of Table 2.

3.1 Using knee point detection algorithm for k

Tables 3—6 summarise the results of each initialisation method on the benchmark datasets where the number of clusters has been determined by the knee point detection algorithm. Each column shows the mean value of each metric and its standard 250 repetitions of the k-modes algorithm.

	Initial cost	Final cost	No. iterations	Time
Cao	3118.00 (0.000)	2774.00 (0.000)	4.00 (0.000)	0.30 (0.012)
Huang	2856.50 (104.245)	2748.83 (64.514)	2.68 (0.817)	0.22 (0.046)
Matching	2870.11 (101.869)	2752.59 (52.387)	2.72 (0.760)	0.16 (0.021)

Table 3: Summative metric results for the breast cancer dataset with $k = 8$.

	Initial cost	Final cost	No. iterations	Time
Cao	20381.00 (0.000)	20376.00 (0.000)	2.00 (0.000)	4.68 (0.205)
Huang	23027.24 (1209.753)	21869.06 (747.766)	2.90 (0.934)	5.11 (1.138)
Matching	23279.36 (1498.324)	21855.50 (751.641)	3.02 (0.936)	2.77 (0.325)

Table 4: Summative metric results for the mushroom dataset with $k = 17$.

	Initial cost	Final cost	No. iterations	Time
Cao	35544.00 (0.000)	35544.00 (0.000)	1.00 (0.000)	4.98 (0.152)
Huang	37535.06 (372.596)	37535.06 (372.596)	1.00 (0.000)	3.58 (0.121)
Matching	37484.29 (327.467)	37484.29 (327.467)	1.00 (0.000)	3.14 (0.141)

Table 5: Summative metric results for the nursery dataset with $k = 23$.

	Initial cost	Final cost	No. iterations	Time
Cao	1654.00 (0.000)	1585.00 (0.000)	4.00 (0.000)	0.28 (0.014)
Huang	1829.31 (92.308)	1708.55 (69.740)	3.58 (1.019)	0.28 (0.063)
Matching	1827.76 (86.852)	1711.49 (73.319)	3.42 (0.963)	0.17 (0.022)

Table 6: Summative metric results for the soybean dataset with $k = 8$.

By examining these tables it would seem that the proposed method and Huang's method are comparable across the board — although the proposed method is faster despite taking more iterations in general which may relate to a more intuitive initialisation. More importantly though, it appears that Cao's method performs the best out of the three initialisation methods: in terms of initial and final costs Cao's method improves, on average, by roughly 10 percent against the next best method for the three datasets that it succeeds with; the number of iterations is comparable; and the computation time is substantially less than the other two considering it is a deterministic

method and need only be run once to achieve this performance.

However, in the k-means paradigm, a particular clustering is selected based on it having the minimum final cost over a number of runs of the algorithm — not the mean — and while Cao's method is very reliable, in that there is no variation at all, it does not always produce the best clustering possible. There is a trade-off to be made between computational time and performance here. In order to gain more insight into the performance of each method, less granular analysis is required. Figures 1—4 display the cost function results for each dataset in the form of a scatter plot and two empirical cumulative density function (CDF) plots, highlighting the breadth and depth of the behaviours exhibited by each initialisation method.

Looking at Figure 1 it is clear that in terms of final cost Cao's method is middling when compared to the other methods. This was apparent from Table 3 and, indeed, Huang's and the proposed method are both very comparable when looking at the main body of the results. However, since the criterion for the best clustering (in practical terms) is having the minimum final cost, it is evident that the proposed method is superior; that the method produces clusterings with a larger cost range (indicated by the trailing right-hand side of each CDF plot) is irrelevant for the same reason.

This pattern of largely similar behaviour between Huang's and the proposed method is apparent in each of the figures here, and in each case the proposed method outperforms Huang's. In fact, in all cases except for the nursery dataset, the proposed method achieves the lowest final cost of all the methods and, as such, performs the best in practical terms on these particular datasets.

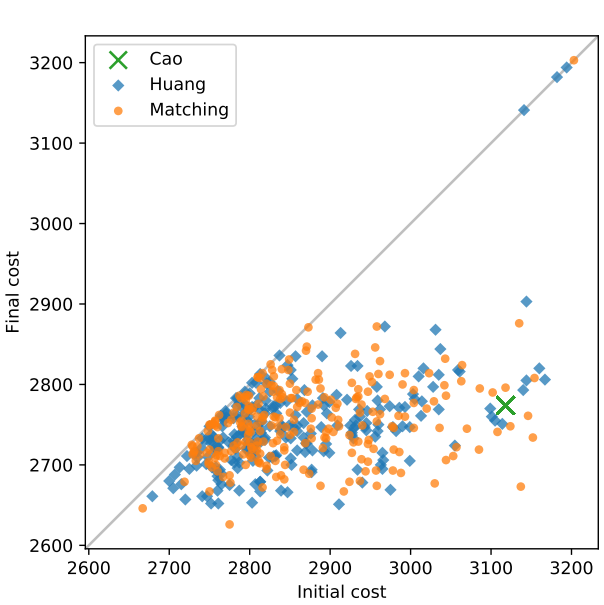
In the case of the nursery dataset, Cao's method is unquestionably the best performing initialisation method. It should be noted that none of the methods were able to find an initial clustering that could be improved on, and that this dataset exactly describes the entire attribute space in which it exists. This property could be why the other methods fall behind Cao's so decisively in that Cao's method is able to definitively choose the k most dense-whilst-separated points from the attribute space as the initial cluster centres whereas the other two methods are in essence randomly sampling from this space. That each initial solution in these repetitions is locally optimal remains a mystery.

3.2 Using number of classes for k

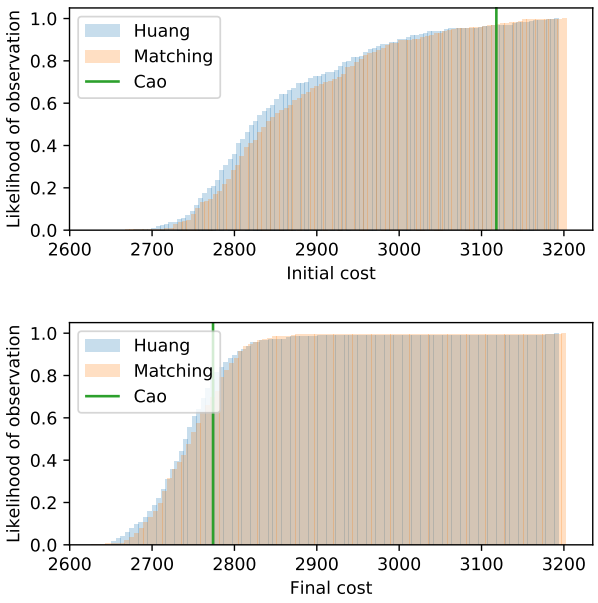
As is discussed above, the often automatic choice for k is the number of classes present in the data; this subsection repeats the analysis from the subsection above but with this traditional choice for k. Tables 7—10 contain the analogous summaries of each initialisation method's performance on the benchmark datasets over the same number of repetitions.

An immediate comparison to the previous tables is that for all datasets bar the soybean dataset, the mean costs are significantly higher and the computation times are lower. These effects come directly from the choice of k in that higher values of k will require more checks (and thus computational time) but will typically lead to more homogeneous clusters, reducing their within-cluster dissimilarity and therefore cost.

Looking at these tables on their own, Cao's method is the superior initialisation method on average: the means are substantially lower in terms of initial and final cost; there is no deviation in these results; again, the total computational time is a fraction of the other two methods. It is also apparent that Huang's method and the proposed extension are very comparable on average. As before, finer investigation will require finer visualisations. Figures 5—8 show the same plots as in

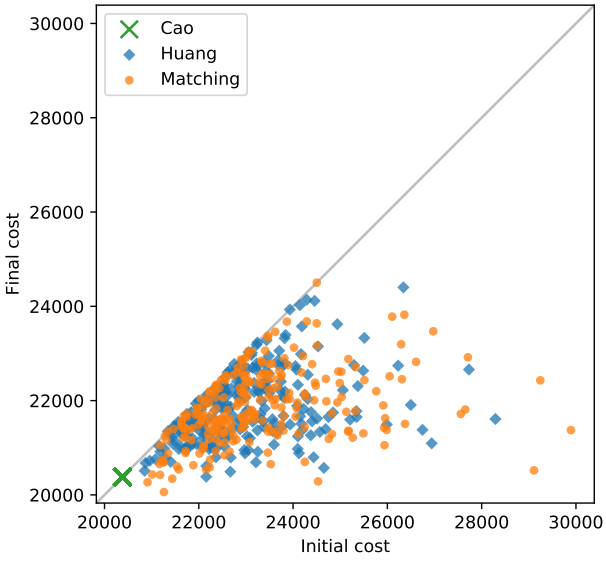


(a) Scatter plot of initial and final costs.

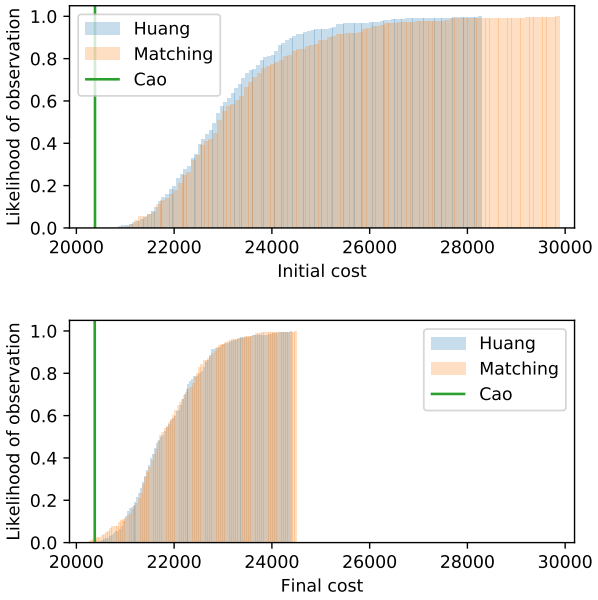


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 1: Summative plots for the breast cancer dataset with $k = 8$.

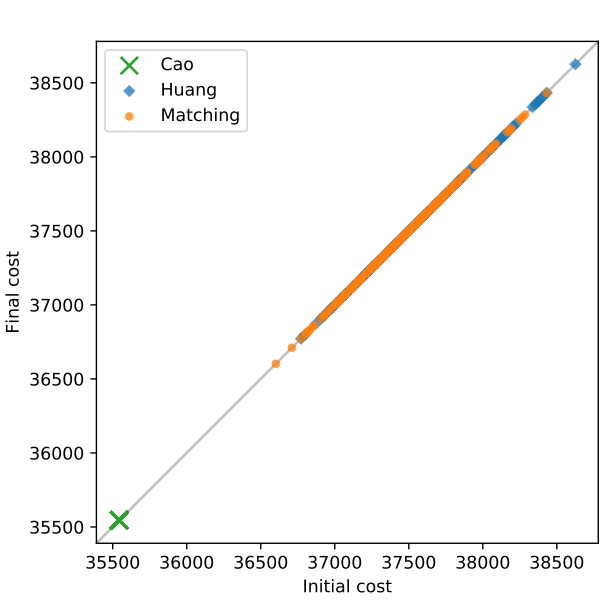


(a) Scatter plot of initial and final costs.

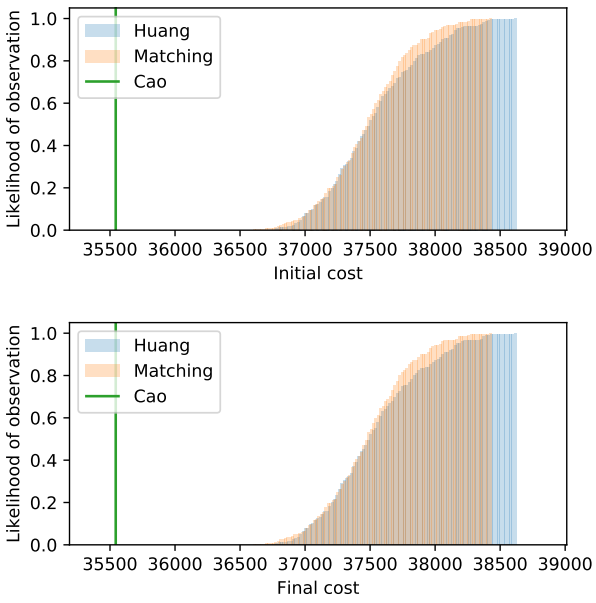


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 2: Summative plots for the mushroom dataset with $k = 17$.

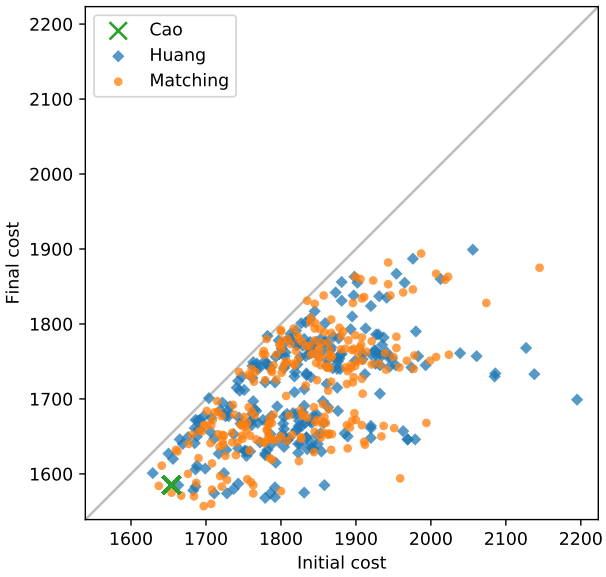


(a) Scatter plot of initial and final costs.

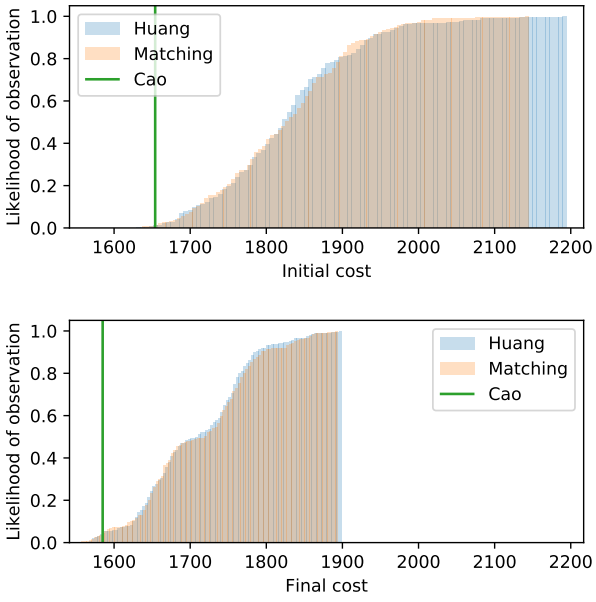


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 3: Summative plots for the nursery dataset with $k = 23$.



(a) Scatter plot of initial and final costs.



(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 4: Summative plots for the soybean dataset with $k = 8$.

	Initial cost	Final cost	No. iterations	Time
Cao	3315.00 (0.000)	3172.00 (0.000)	2.00 (0.000)	0.13 (0.005)
Huang	3393.80 (120.772)	3348.51 (144.849)	1.54 (0.653)	0.10 (0.024)
Matching	3406.73 (111.686)	3355.56 (144.621)	1.61 (0.638)	0.09 (0.018)

Table 7: Summative metric results for the breast cancer dataset with $k = 2$.

	Initial cost	Final cost	No. iterations	Time
Cao	37662.00 (0.000)	37662.00 (0.000)	1.00 (0.000)	0.94 (0.035)
Huang	41974.07 (2393.889)	39226.25 (2483.933)	3.11 (1.430)	1.92 (0.679)
Matching	42175.54 (2520.163)	39617.53 (2637.574)	3.03 (1.439)	1.38 (0.491)

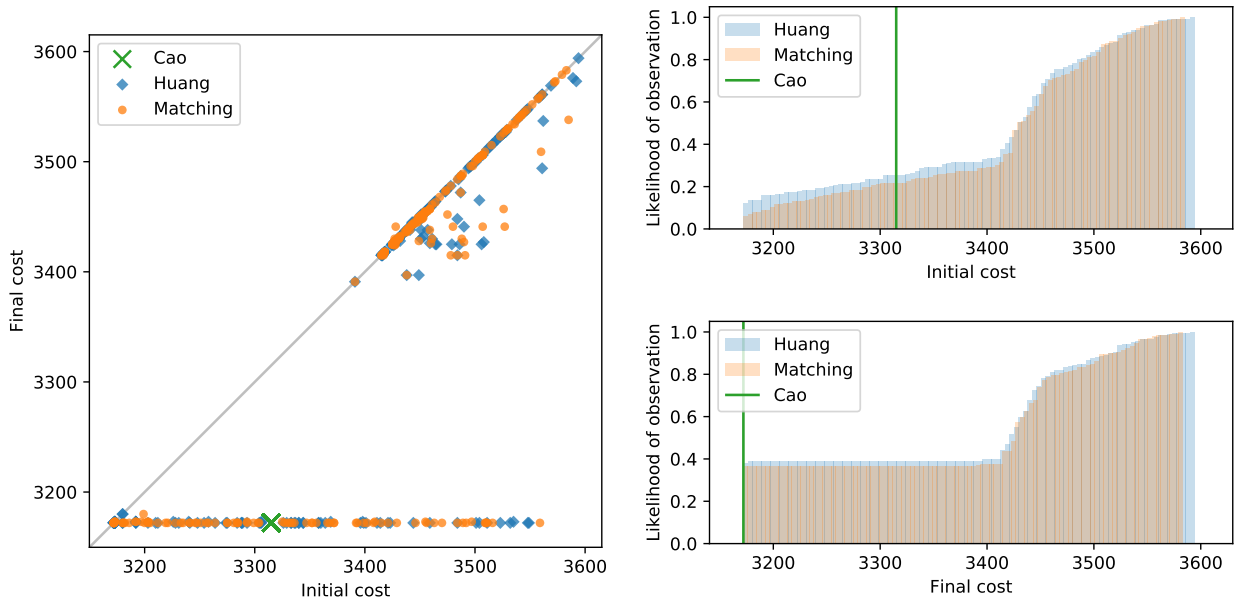
Table 8: Summative metric results for the mushroom dataset with $k = 2$.

	Initial cost	Final cost	No. iterations	Time
Cao	49060.00 (0.000)	49060.00 (0.000)	1.00 (0.000)	1.80 (0.090)
Huang	51229.45 (902.503)	51229.45 (902.503)	1.00 (0.000)	1.72 (0.116)
Matching	51107.52 (910.258)	51101.95 (903.525)	1.00 (0.063)	1.37 (0.128)

Table 9: Summative metric results for the nursery dataset with $k = 5$.

	Initial cost	Final cost	No. iterations	Time
Cao	1364.00 (0.000)	1314.00 (0.000)	2.00 (0.000)	0.33 (0.009)
Huang	1588.89 (83.682)	1446.22 (59.844)	4.02 (1.081)	0.45 (0.085)
Matching	1582.56 (87.418)	1447.08 (60.154)	4.01 (1.128)	0.24 (0.025)

Table 10: Summative metric results for the soybean dataset with $k = 15$.



(a) Scatter plot of initial and final costs.

(b) Empirical CDF plots for initial (top) and final (bottom) costs.

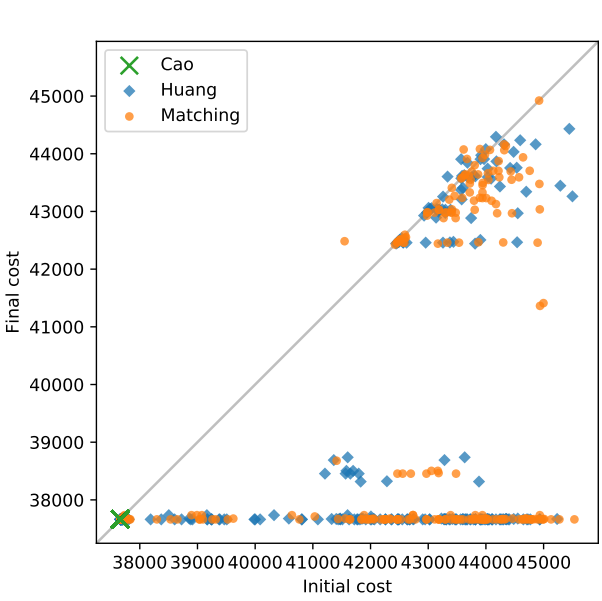
Figure 5: Summative plots for the breast cancer dataset with $k = 2$.

the previous subsection except the number of clusters has been taken to be the number of classes present in each dataset.

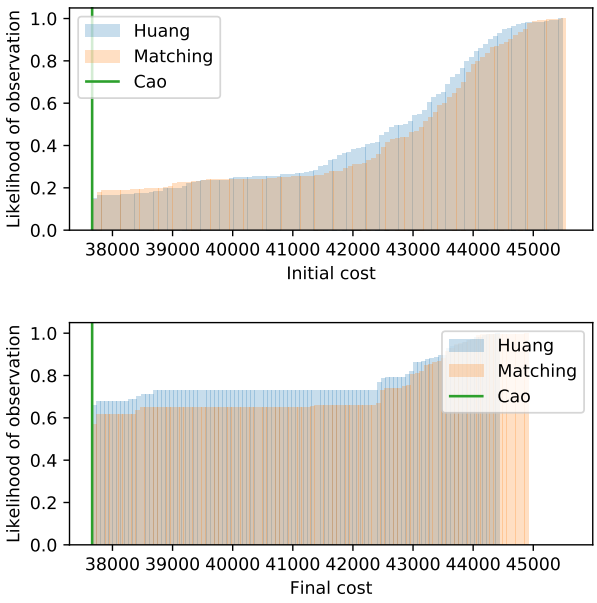
Figures 5 & 6 indicate that a particular behaviour emerged during the runs of the k-modes algorithm. Specifically, each solution falls into one of (predominantly) two types: effectively no improvement on the initial clustering, or terminating at some clustering with a cost that is bounded below across all such solutions. Invariably, Cao's method achieves or approaches this lower bound and unless Cao's method is used, these particular choices for k mean that the performance of the k-modes algorithm is exceptionally sensitive to its initial clustering. Moreover, the other two methods are effectively indistinguishable in these cases and so if a robust solution is required, Cao's method is the only viable option.

Figure 7 corresponds to the nursery dataset results with $k = 5$. In this set of runs, the same pattern emerges as in Figure 3 where sampling the initial centres from amongst the most dense points (via Huang's method and the proposed) is an inferior strategy to one considering the entire attribute space such as with Cao's method. Again, no method is able to improve on the initial solution except for one repetition with the matching initialisation method.

The primary conclusion from this analysis is that while Huang's method is largely comparable to the proposed extension, there is no substantial evidence from these use cases to use Huang's method over the one proposed in this work. In fact, Figure 8 is the only instance where Huang's method was able to outperform the proposed method. Other than this, the proposed method consistently performing better (or as well as) Huang's method in terms of minimal final costs and computational time over a number of runs in both the cases where an external framework is imposed on the data (by choosing k to be the number of classes) and not. Furthermore, though not discussed in this work, the matching initialisation method has the scope to allow for expert or prior knowledge to

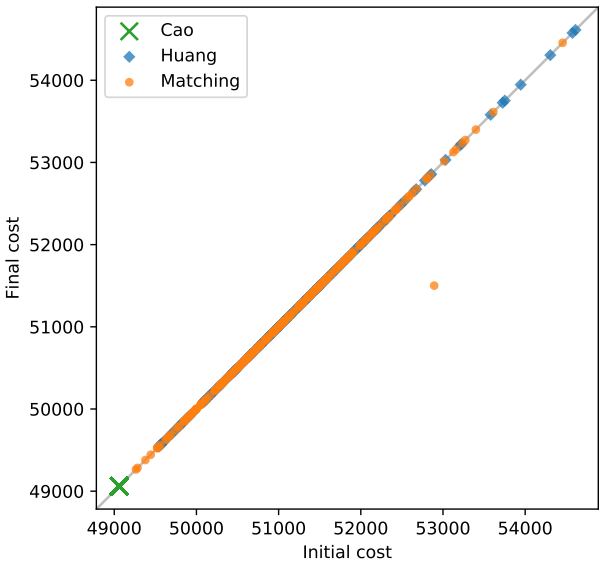


(a) Scatter plot of initial and final costs.

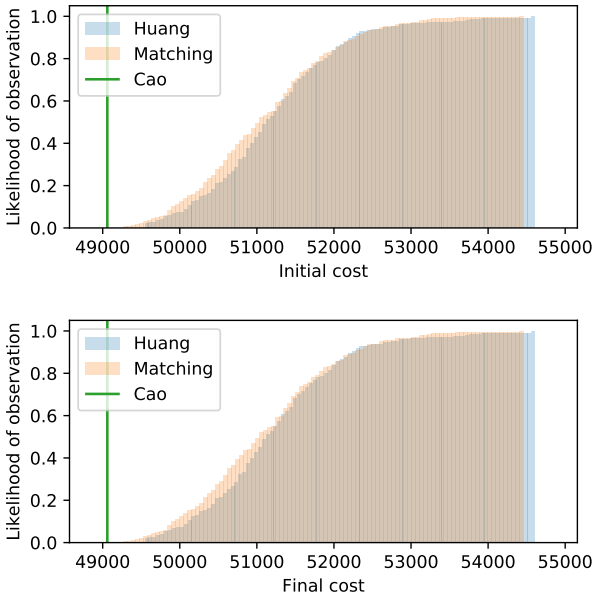


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 6: Summative plots for the mushroom dataset with $k = 2$.

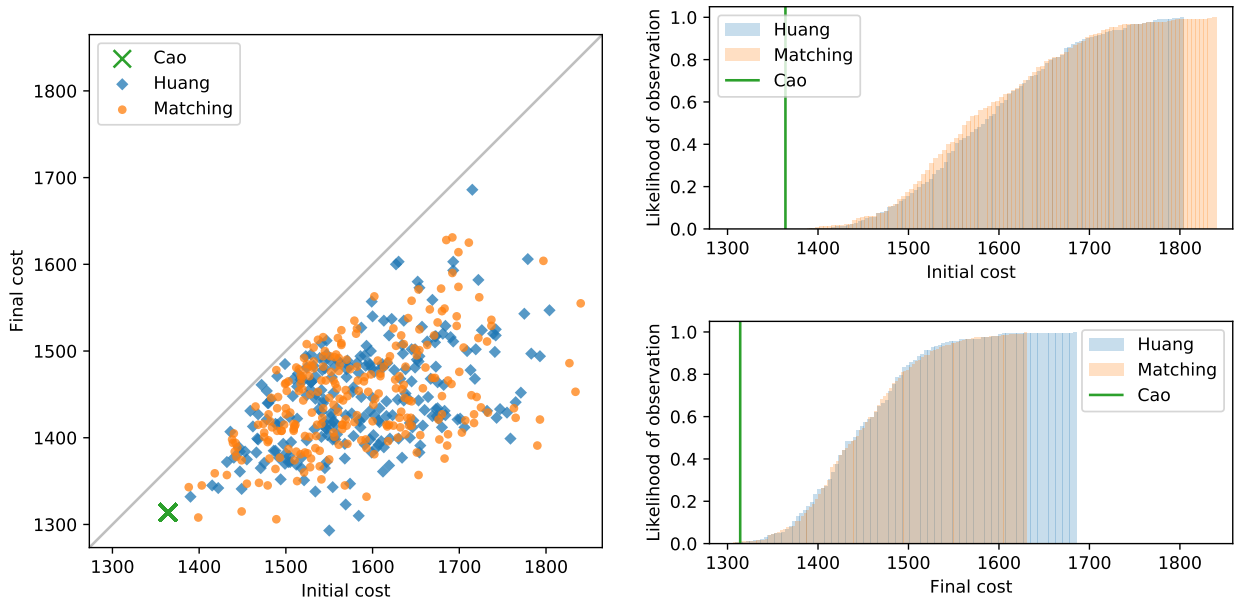


(a) Scatter plot of initial and final costs.



(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 7: Summative plots for the nursery dataset with $k = 5$.



(a) Scatter plot of initial and final costs.

(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 8: Summative plots for the soybean dataset with $k = 15$.

be included in an initial clustering by using some *ad hoc* preference list mechanism.

3.3 Artificial datasets

Following on from the conclusions of the analysis thus far, the competition between Cao's method and the proposed matching method may be studied more deeply. All of the results leading up to this point were conducted using benchmark datasets and while there are certainly benefits to comparing methods in this way, it does not afford a rich understanding of how any of them perform more generally. This stage of the analysis relies on a method for generating artificial datasets introduced in [**?**]. In essence, this method is an evolutionary algorithm which acts on entire datasets to explore the space in which potentially all possible datasets exist. The key component of this method is an objective function that takes a dataset and returns a value that is to be minimised; this function is referred to as the fitness function.

In order to reveal the nuances in the performance of Cao's method and the proposed initialisation on a particular dataset, two cases are considered: where Cao's method outperforms the proposed, and vice versa. Both cases use the same fitness function — with the latter using its negative — which is defined as follows:

$$f(\mathcal{X}) = C_{\text{cao}} - C_{\text{match}} \tag{7}$$

where C_{cao} and C_{match} are the final costs when a dataset $\mathcal{X}$ is clustered using Cao's method and the proposed matching method respectively with $k = 3$. For the sake of computational time, the proposed initialisation was given 25 repetitions as opposed to the 250 repetitions in the remainder of this section. Apart from the sign of f, the dataset generation processes used identical parameters

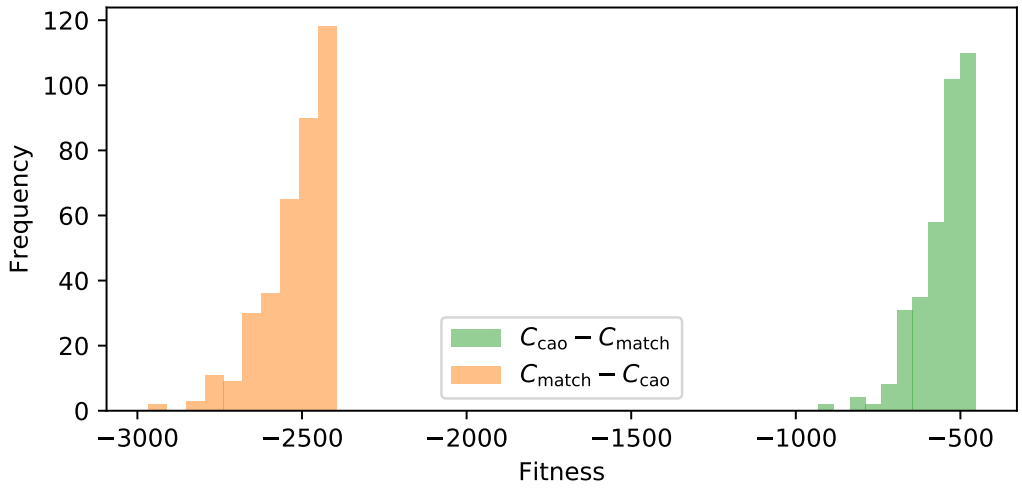


Figure 9: Histograms of fitness for the top performing percentile in each case.

in each case and the datasets considered here are all of comparable shape.

This process yielded approximately 35,000 unique datasets for each case, and the ensuing analysis only considers the top-performing percentile of datasets from each. Figure 9 shows the fitness distribution of the top percentile in each case. It should be clear from (7) that large negative values are preferable here. With that, and bearing in mind that the generation of these datasets was parameterised in a consistent manner, it appears that the attempt to outperform Cao's method proved somewhat easier. This is indicated by the substantial difference in the locations of the fitness distributions.

Given the quantity of data available, to understand the patterns that have emerged, they must be summarised; in this case, univariate statistics are used. Despite the datasets all being of similar shapes, there are some discrepancies. With the number of rows this is less of an issue but any comparison of statistics across datasets of different widths is difficult without prior knowledge of the datasets. Moreover, there is no guarantee of contingency amongst the attributes, and the comparison of more than a handful of variables becomes complicated even when the attributes are identifiable. To combat this and bring uniformity to the datasets, each dataset is represented as their first principal component obtained via centred Principal Component Analysis (PCA) [**?**]. While some subtleties may be lost, this representation captures the most important characteristics of each dataset in a single variable meaning they can be compared directly.

Since the transformation by PCA is centred, all measures for central tendency are moot. In fact, the mean and median are not interpretable here given that the original data is categorical. As such, the univariate statistics used here describe the spread and shape of the principal components, and are split into two groups:

- Central moments: variance, skewness and kurtosis.
- Empirical quantiles: interquartile range, lower decile and upper decile.

Figures 10 & 11 show the distributions of the six univariate statistics across all of the principal components in each case. In addition to this, they show a fitted Gaussian kernel density estimate [**?**

] to accentuate the general shape of the histograms. What becomes immediately clear from each of these plots is that for datasets where Cao's method succeeds, the general spread of their first principal component is much tighter than in the case where the proposed initialisation method succeeds. This is particularly evident in Figure 10a where relatively low variance in the first case indicates a higher level of density in the original categorical data.

The patterns in the quantiles further this. Although Figure 11a suggests that the components of Cao-preferable datasets can have higher interquartile ranges than in the second case, the lower and upper deciles tend to be closer together as is seen in Figures 11b & 11c. This suggests that despite the body of the component being spread, its extremities are not.

In Figures 10b & 10c, the most notable contrast between the two cases is the range in values for both skewness and kurtosis. This supports the evidence thus far that individual datasets have higher densities and lower variety (i.e. tighter extremities) when Cao's method succeeds over the proposed initialisation. In particular, larger values of skewness and kurtosis translate to high similarity between the instances in a categorical dataset which is equivalent to having high density.

Overall, this analysis has revealed that if a dataset shows clear evidence of high-density points, then Cao's method should be used over the proposed method. However, if there is no such evidence, the proposed method is able to find a substantially better clustering than Cao's method.

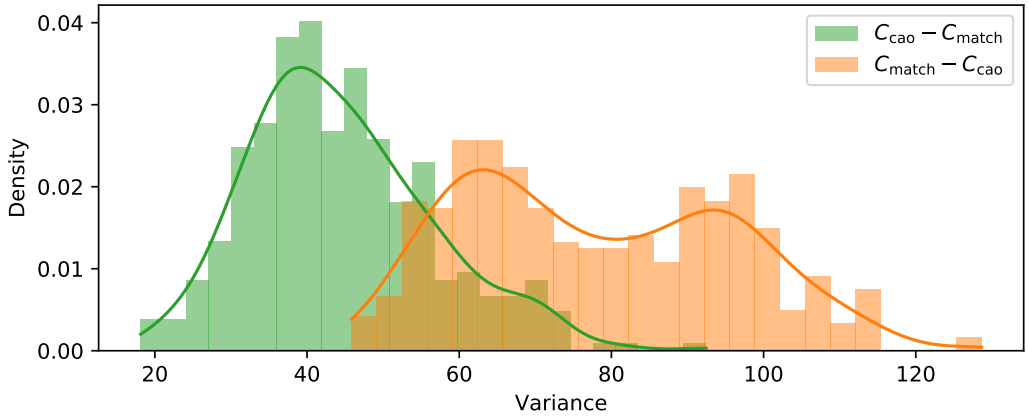
4 Conclusion

In this paper a novel initialisation method for the k-modes was introduced that built on the method set out in the seminal paper [**?**]. The new method models the final 'replacement' process in the original as an instance of the Hospital-Resident Assignment Problem that may be solved to be mathematically fair and stable.

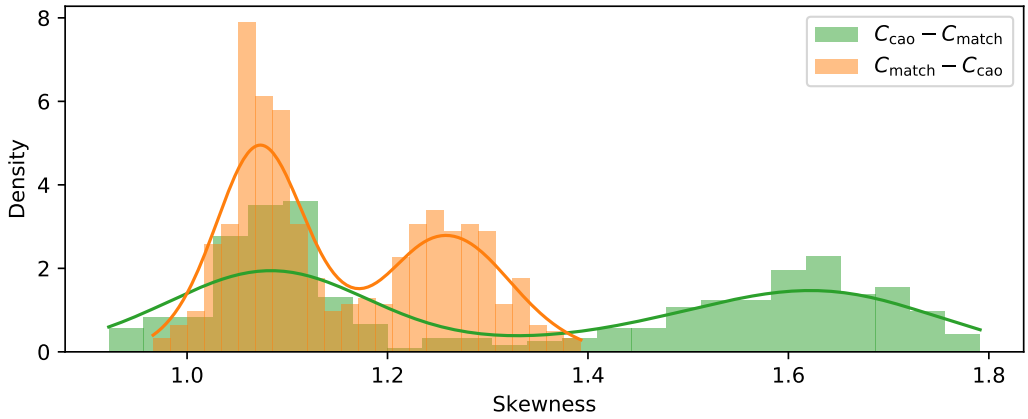
Following a thorough description of the k-modes algorithm and the established initialisation methods, a comparative analysis was conducted amongst the three initialisations using both benchmark and artificial datasets. This analysis revealed that the proposed initialisation was able to outperform both of the other methods when the choice of k was optimised according to a mathematically rigorous elbow method. However, the proposed method was unable to beat Cao's method (established in [**?**]) when an external framework was imposed on each dataset by choosing k to be the number of classes present.

The proposed method should be employed over Cao's when there are no hard restrictions on what k may be, or if there is no immediate evidence that the dataset at hand has some notion of high density. Otherwise, Cao's method remains the most reliable initialisation in terms of computational time and final cost.

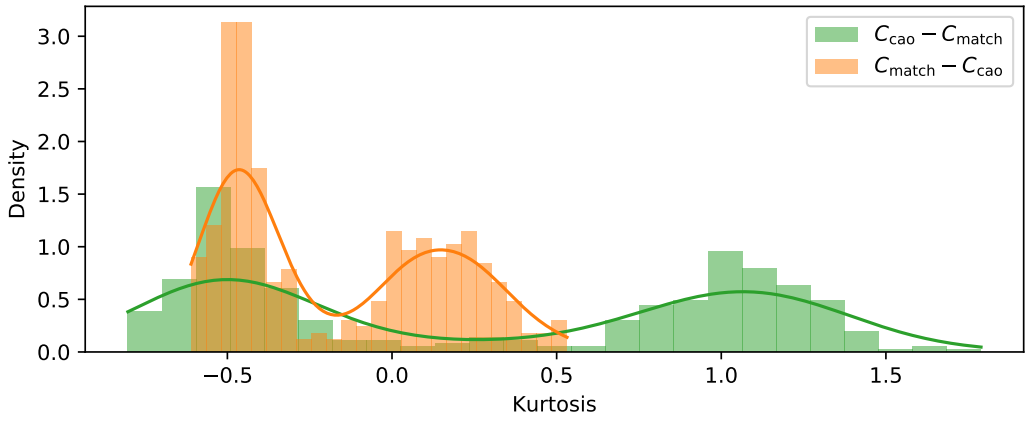
References



(a)

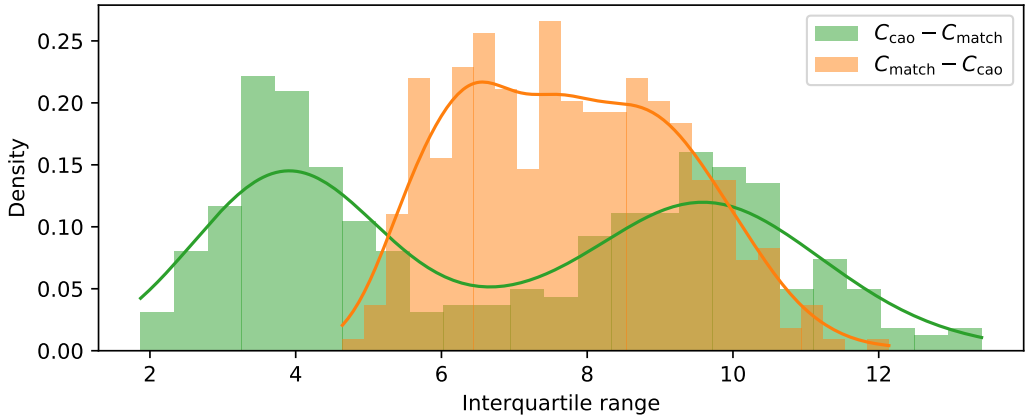


(b)

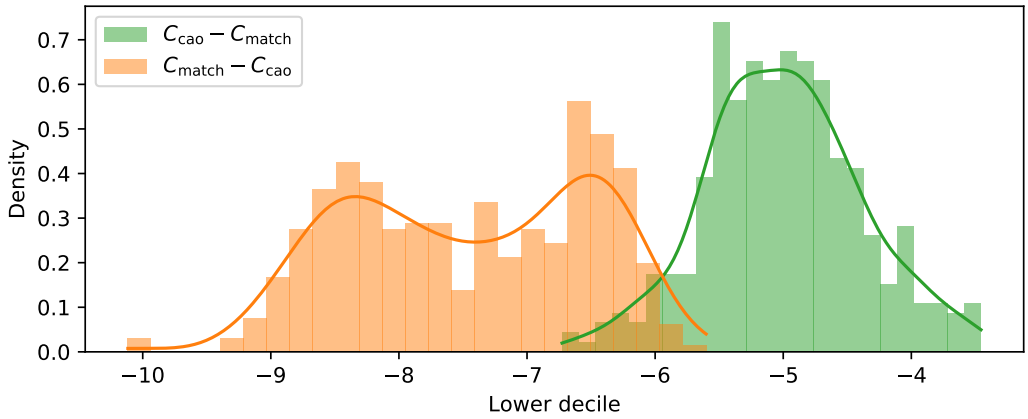


(c)

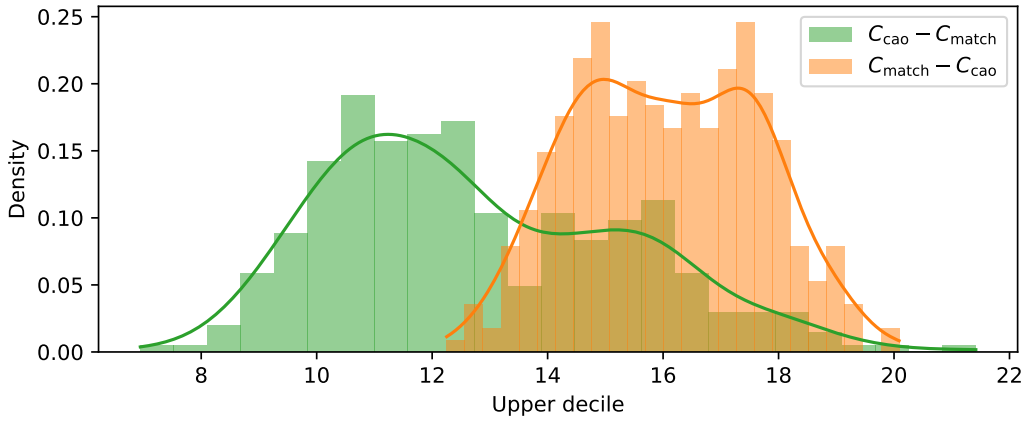
Figure 10: Distribution plots for the (a) variance, (b) skewness and (c) kurtosis of the first principal components in each case.



(a)



(b)



(c)

Figure 11: Distribution plots for the (a) interquartile range, (b) lower decile and (c) upper decile of the first principal components in each case.