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

Henry Wilde, Vincent Knight and Jonathan Gillard

January 19, 2020

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

Clustering is an unsupervised learning technique for discovering intrinsic structure within data. There exist many approaches to clustering but perhaps the most ubiquitous amongst them is centroid-based clustering. This approach aims to maximise summed within-cluster similarity by iterating over the points in a dataset and adjusting the current clusters according to some measure for central tendency until convergence. A popular algorithm for performing centroid-based clustering is the k-means algorithm. In k-means, a number of groups to identify in a dataset, k, is fixed *a priori* and each cluster has associated with it a centroid or representative point calculated as the mean of the data points within that cluster. Unfortunately, this is only valid for numeric data where the mean of a set is well-defined. Despite this, the paradigm in which k-means clustering exists is of interest as it is fast, scalable, easily parallelised, and simple in its design [29, 31].

In this work, the focus will be 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 [13, 14, 15]. An alternative, though largely equivalent, form for k-modes was presented in [6] but is not considered in this work. Under this framework, the central tendency measure used is the mode and Euclidean distance is replaced by a simple matching measure. All of the concepts used to define the k-modes algorithm are presented later in this section.

The interest of this paper is in how the performance of the k-modes algorithm may be affected. Since the k-modes algorithm is a heuristic, its performance is dependent on its initial solution. The

quality of the initial solution is affected by two components: the metric being used and the process by which the solution is chosen.

Strictly, introducing a new metric alters the space in which the data exists and its effect on the initial solution is not independent of the final solution. Having said that, following the seminal k-modes papers, a number of alternative dissimilarity measures have been implemented to improve on the simple matching dissimilarity used most regularly. The main drawback of the standard measure is that it often produces clusters with low intra-cluster similarity [23] 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 [5, 30, 32]. However, these measures sometimes are defined between a point in the dataset and a centroid rather than defining a metric for the entire space.

Instead of adjusting the overall space, this work considers the process by which an initial solution is found. The proposed method is an extension to that presented by Huang [15] that generates a game-theoretically fair and stable variant to that generated by Huang's method. The remainder of this 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 method.
- Section 3 presents analyses of the initialisation methods 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 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 [15].

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 [15].

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

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.

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

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 [15]. 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, $\overline{Z} \subset \mathcal{X}$.

In the original statement of Huang's method, it is stated that the most frequent categories should be assigned 'equally' to the set of potential modes. How the categories should be distributed 'equally' is not well-defined or easily seen from the example given in the paper. In software implementations, including the one used in Section 3, the term is taken to mean using a probability distribution to sample values from the attribute space. This probability distribution is formed by the relative frequencies of each attribute's categories.

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 [4]. This method selects representative points 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 often considered to be deterministic as there is no formally stochastic element. However, this is only true up to an arbitrary breaking of ties in the density-dissimilarity calculations and so many practical implementations cannot guarantee a unique solution across multiple runs using this method.

Definition 1.5. Consider a data set $\mathcal{X}$ with attribute set $\mathcal{A} = \{A_1, \ldots, A_m\}$. Then the *average density* of any point $X_i \in \mathcal{X}$ with respect to $\mathcal{A}$ is defined [4] 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:

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

With this alternative definition, it is clear — since m and N are fixed positive integers — that $\text{Dens}(X^{(i)})$ is maximised when $D(\mathcal{X},X^{(i)})$ is minimised. Then by Theorem 1, any data point with maximal average density is, in fact, a mode of $\mathcal{X}$. This observation indicates that there is a similarity between this method and Huang's in that they are attempting to achieve the same objective if only from opposite ends.

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 $\text{Dens}\left(X^{(i)}\right)$
end
Select $1\leq i_1\leq N$ which maximises $\text{Dens}\left(X^{(i)}\right)$
$\overline{Z}\leftarrow\overline{Z}\cup\left\{X^{(i_1)}\right\}$
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, 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 [9, 10, 19].

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 [10] 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 [17, 18]. 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 [8]. 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 [16, 20]. 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, 22] or estimating them *ad hoc* [25] 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 [3]. 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 [11, 12].

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 [10].

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 [26]. 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
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 $\overline{Z}$
$\overline{Z} \leftarrow \emptyset$
$H \leftarrow \emptyset$
$R \leftarrow \text{SamplePotentialModes}\,(\mathcal{X})$
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 [7], and their characteristics are summarised in Table 2.

	No. rows	No. columns	No. classes	Missing values	Adjusted no. rows	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.

Clustering algorithms are often evaluated based on their performance as a classifier [2, 4, 5, 15, 23, 24, 28]. This is a fundamentally flawed approach — especially given that classification belongs to an entirely different branch of learning. Moreover, doing so requires a number of assumptions about the topology of the data within the metric space that is being considered [21]. One such assumption is that the classes recorded in the data are indeed separable objects like clusters. The soybean dataset acts as an example where the labels attached to the dataset may be too fine with respect to the space in which the data exists. That is, in this case, only 8 clusters were identified from a possible 15 classes. This indicates that some of the classes are not easily distinguished from one another. Likewise with the remaining datasets, an increase in the number of clusters on the number of classes could suggest that the classes are too vague or at the very least that more structure exists in the data than is imposed. In any case, the choice of k when clustering is of great importance and should strike a balance between explaining structure in the data and interpretability of a model.

This analysis does not consider evaluative metrics related to classification such as accuracy, recall or precision. 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, as stated above, this is not necessarily a fair or wise choice since the classes may not be representative of true clusters. A popular strategy for choosing an optimal number of clusters is known as the 'elbow' method. The aim of this method is to identify a kink (elbow) in a plot of number of clusters against cost for a dataset. Figure 1 is an example of such a plot for the nursery dataset. This kink suggests that an

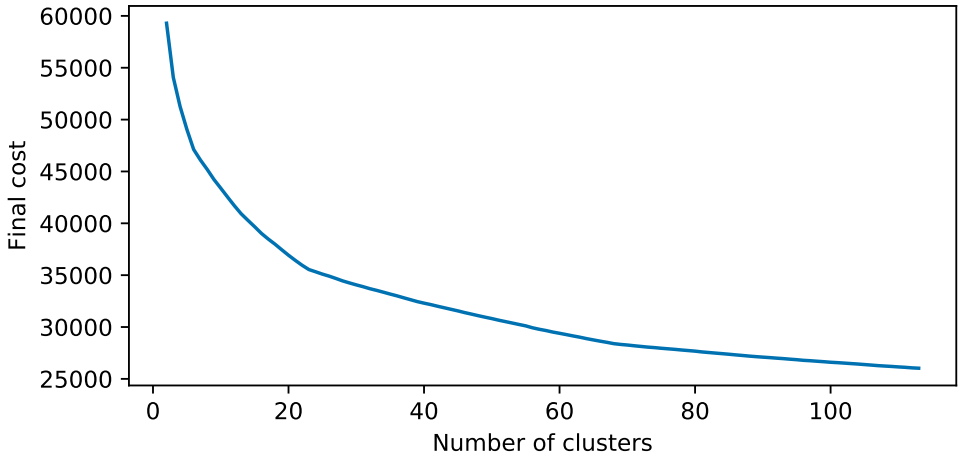


Figure 1: An elbow plot for the nursery dataset using Cao's initialisation method.

increase in k from there would not sufficiently improve the performance of the model. On its own, this method is vague and somewhat unreliable which raises a number of questions:

- What constitutes a kink?
- How does one discern between multiple kinks?
- Is choosing a particular kink subjective to the observer?

An alternative 'elbow' may be identified objectively by using the knee point detection algorithm [27] where the maximal value of k is taken to be $\lfloor\sqrt{N}\rfloor$. This algorithm identifies the value of k with the maximum curvature in the plot described above by computation rather than inspection — eliminating the concerns raised.

3.1 Elbow method

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 deviation in parentheses over 250 independent repetitions of the k-modes algorithm.

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. 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

	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 $k = 8$.

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 2—5 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 2 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.

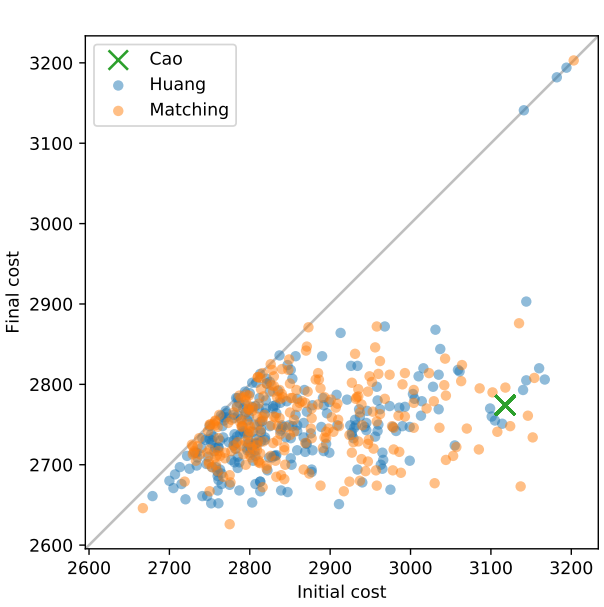
In the case of the nursery dataset, Cao's method is unquestionably the best performing initialisation method. It should be noted that none of the methods were able to find an initial clustering that could be improved on, and that this dataset exactly describes the entire attribute space in which it exists. 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 is locally optimal remains a mystery potential for further work?.

3.2 Number of classes

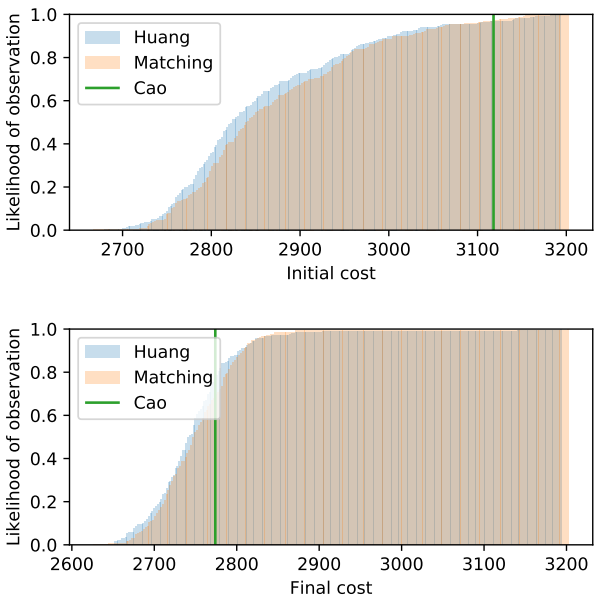
As is discussed above, the often automatic choice for k is the number of classes present in the data; this subsection repeats the analysis as with the elbow method but with this traditional choice for k. Tables 7—10 contain the analogous summaries of each initialisation method's performance on the benchmark 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 6—9 show the same plots as in the previous subsection except the number of clusters has been taken to be the number of classes present in each dataset.

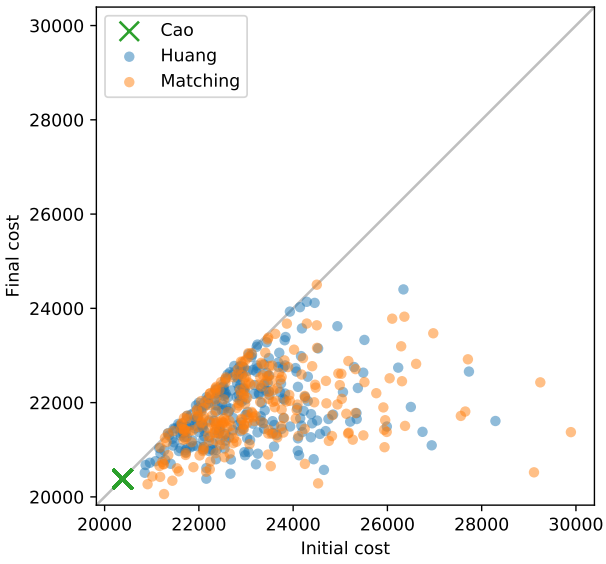


(a) Scatter plot of initial and final costs.

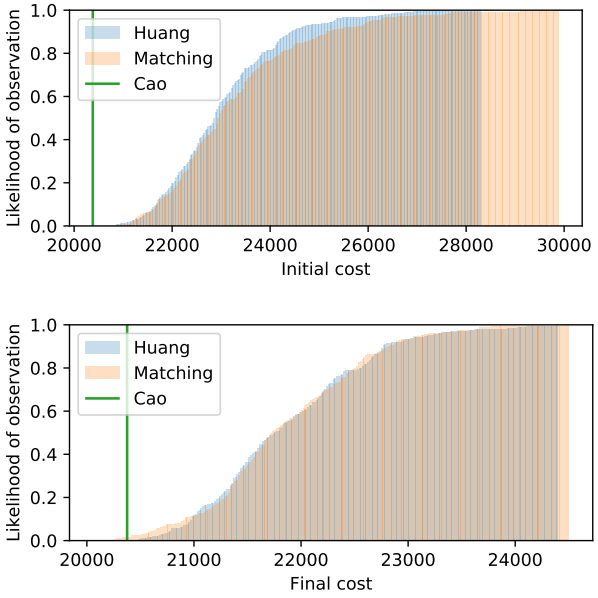


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 2: Summative plots for the breast cancer dataset with $k = 8$.

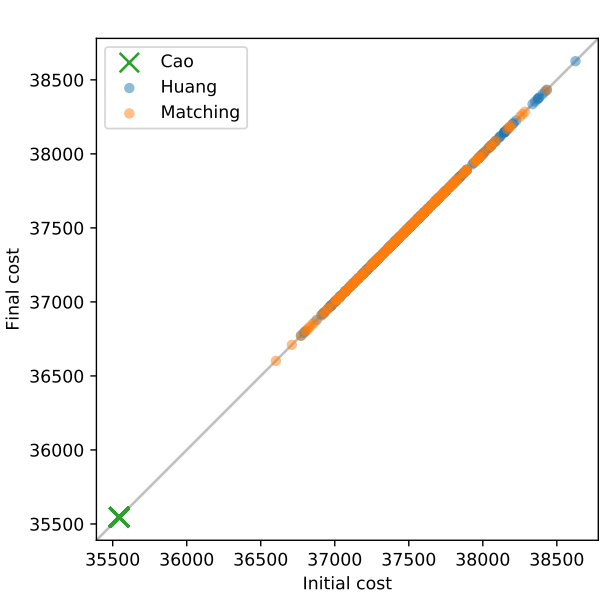


(a) Scatter plot of initial and final costs.

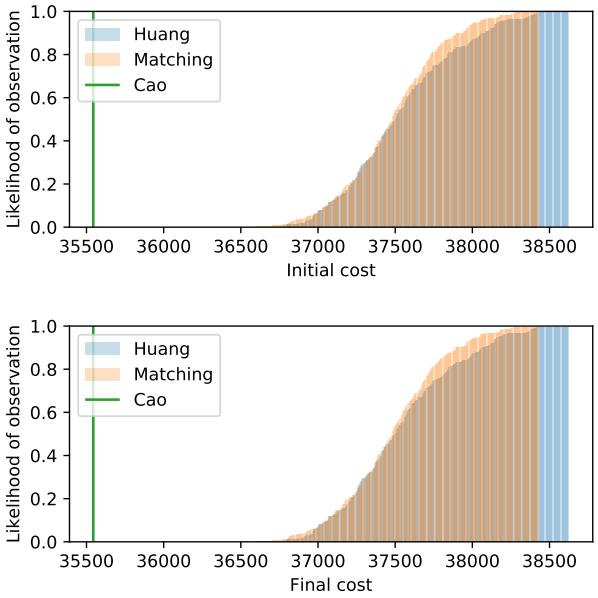


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 3: Summative plots for the mushroom dataset with $k = 17$.

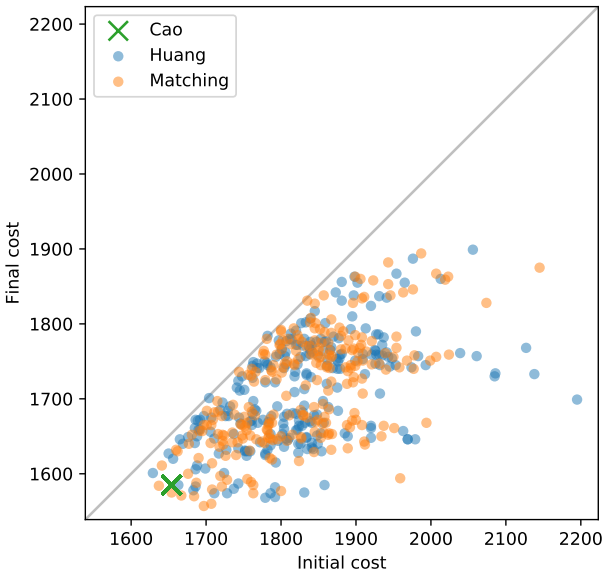


(a) Scatter plot of initial and final costs.

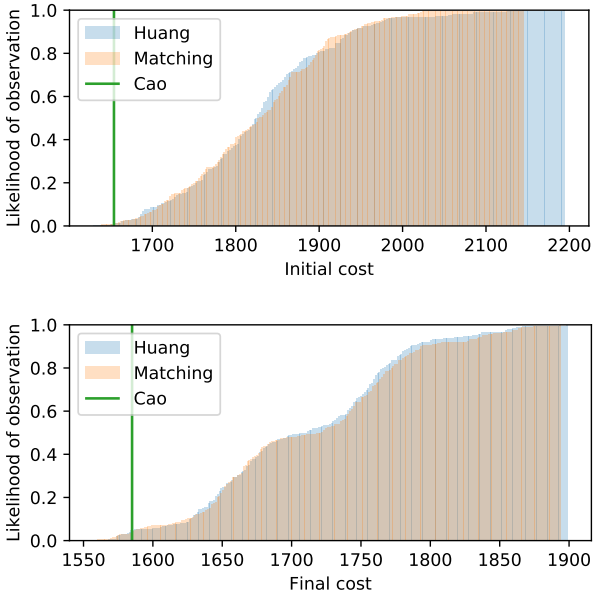


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 4: 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 5: 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 (0.006)
Huang	3393.80 (120.772)	3348.51 (144.849)	1.54 (0.653)	0.11 (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)	0.98 (0.052)
Huang	41974.07 (2393.889)	39226.25 (2483.933)	3.11 (1.430)	1.92 (0.682)
Matching	42175.54 (2520.163)	39617.53 (2637.574)	3.03 (1.439)	1.41 (0.504)

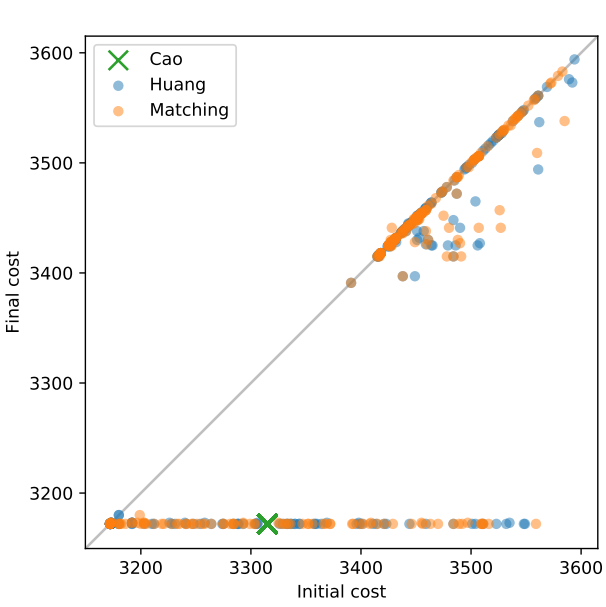
Table 8: Summative metric results for the mushroom dataset with $k = 2$.

	Initial cost	Final cost	No. iterations	Time
Cao	49060.00 (0.000)	49060.00 (0.000)	1.00 (0.000)	1.77 (0.052)
Huang	51229.45 (902.503)	51229.45 (902.503)	1.00 (0.000)	1.70 (0.064)
Matching	51107.52 (910.258)	51101.95 (903.525)	1.00 (0.063)	1.36 (0.050)

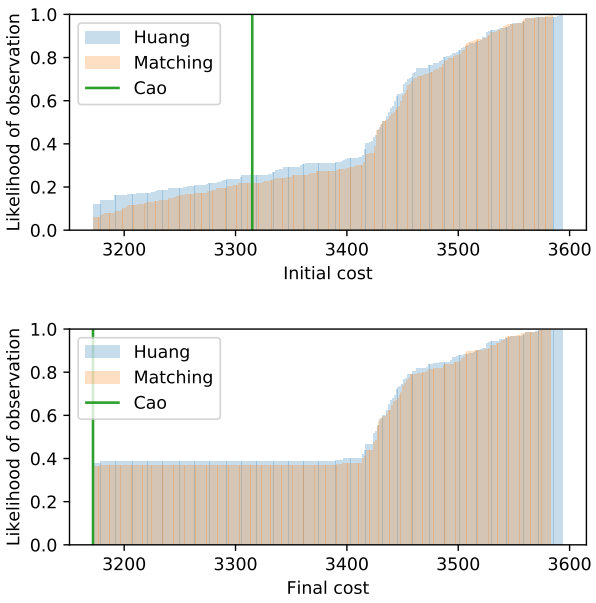
Table 9: Summative metric results for the nursery dataset with $k = 5$.

	Initial cost	Final cost	No. iterations	Time
Cao	1364.00 (0.000)	1314.00 (0.000)	2.00 (0.000)	0.33 (0.010)
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 (0.023)

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

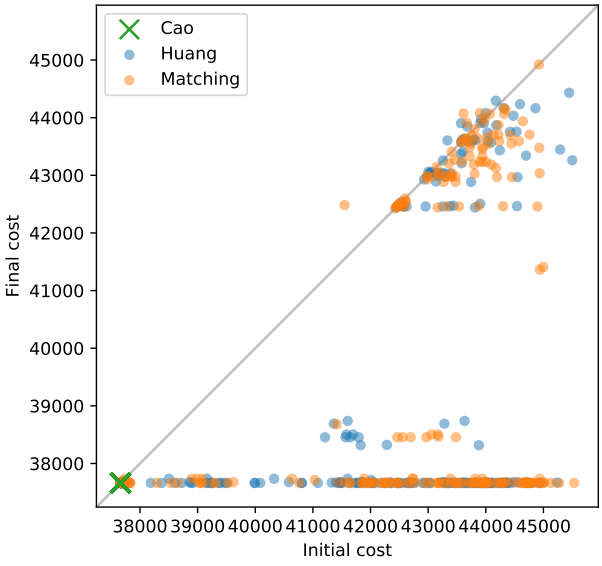


(a) Scatter plot of initial and final costs.

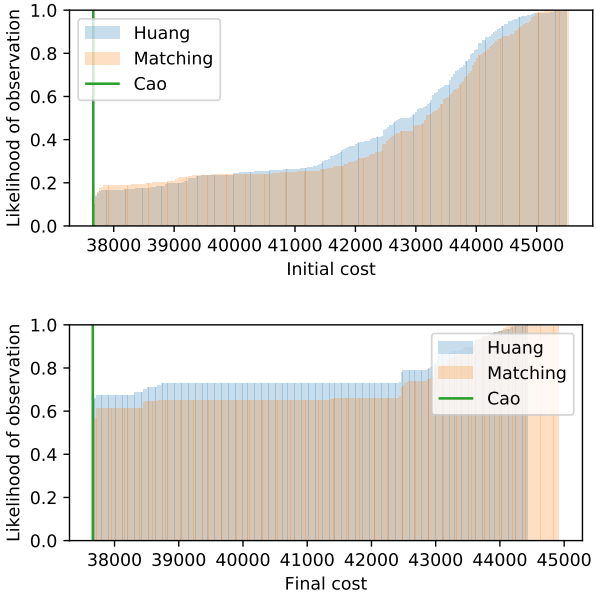


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

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

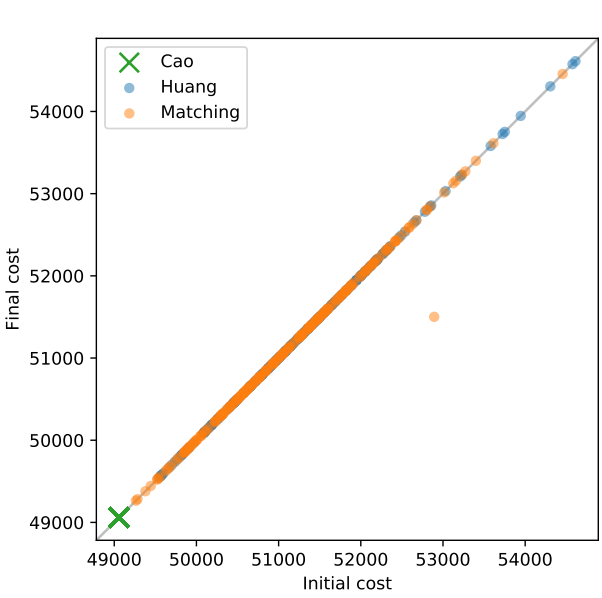


(a) Scatter plot of initial and final costs.

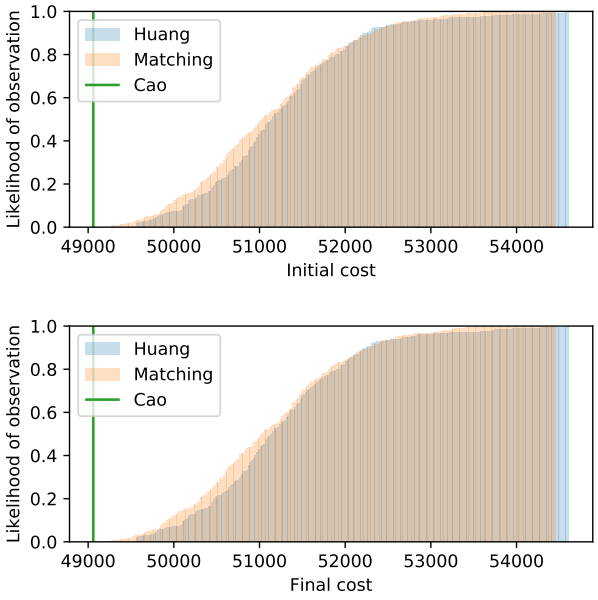


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 7: Summative plots for the mushroom dataset with $k = 2$.

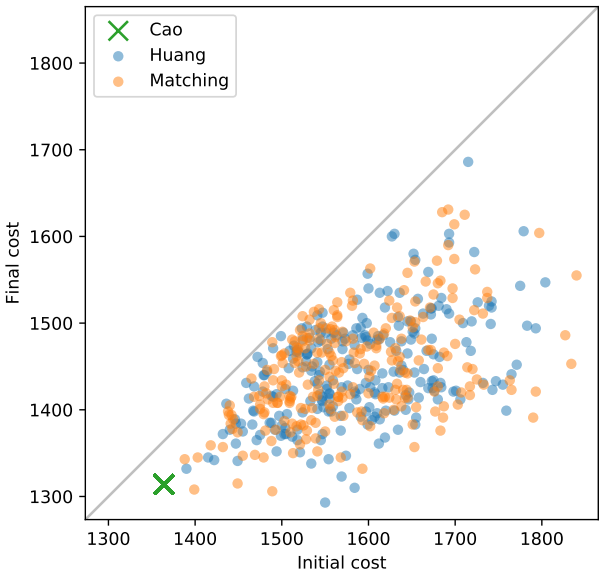


(a) Scatter plot of initial and final costs.

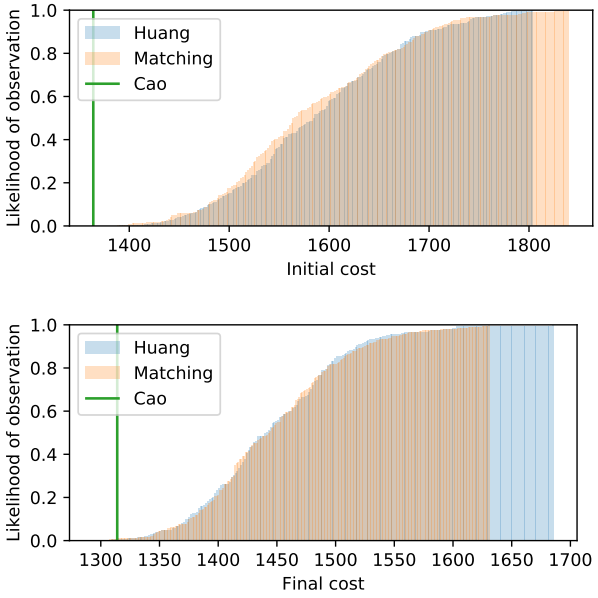


(b) Empirical CDF plots for initial (top) and final (bottom) costs.

Figure 8: 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 9: Summative plots for the soybean dataset with $k = 15$.

4 Conclusion

References

[1] N. Agarwal. Policy analysis in matching markets. *American Economic Review*, 107(5):246–50, 2017. doi: 10.1257/aer.p20171112.

[2] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, 2007. ISBN 978-0-898716-24-5. URL `http://dl.acm.org/citation.cfm?id=1283383.1283494`.

[3] A. Bogomolnaia and H. Moulin. Random matching under dichotomous preferences. *Econometrica*, 72(1):257–279, 2004. doi: 10.1111/j.1468-0262.2004.00483.x.

[4] F. Cao, J. Liang, and L. Bai. A new initialization method for categorical data clustering. *Expert Systems with Applications*, 36:10223–10228, 2009. URL `https://pdfs.semanticscholar.org/1955/c6801bca5e95a44e70ce14180f00fd3e55b8.pdf`.

[5] 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. doi: 10.1016/j.knosys.2011.07.011.

[6] A. Chaturvedi, P. E. Green, and J. D. Caroll. K-modes clustering. *Journal of Classification*, 18(1):35–55, 2001. doi: 10.1007/s00357-001-0004-3.

[7] D. Dua and C. Graff. UCI Machine Learning Repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[8] A. Erdil and H. Ergin. Two-sided matching with indifferences. *Journal of Economic Theory*, 171:268 – 292, 2017. doi: 10.1016/j.jet.2017.07.002.

[9] T. Fuku, A. Namatame, and T. Kaizoji. *Collective Efficiency in Two-Sided Matching*, pages 115–126. 2006. doi: 10.1007/3-540-28547-4_10.

[10] D. Gale and L. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. doi: 10.2307/2312726.

[11] G. Haeringer and V. Iehlé. Two-sided matching with one-sided preferences. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, pages 353–353. ACM, 2014. doi: 10.1145/2600057.2602853.

[12] G. Haeringer and V. Iehlé. Two-sided matching with (almost) one-sided preferences. *American Economic Jounral: Microeconomics*, 11(3):155–190, 2019. doi: 10.1257/mic.20170115.

[13] 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.

[14] 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.

[15] 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, 1998. doi: 10.1023/A:1009769707641.

[16] 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. doi: 10.1109/ICKS.2008.7.

[17] K. Iwama and S. Miyazaki. *Stable Marriage with Ties and Incomplete Lists*, pages 2071–2075. Springer New York, 2016. doi: 10.1007/978-1-4939-2864-4_805.

[18] K. Iwama, S. Miyazaki, Y. Morita, and D. Manlove. Stable marriage with incomplete lists and ties. In *Automata, Languages and Programming*, pages 443–452. Springer Berlin Heidelberg, 1999. doi: 10.1007/3-540-48523-6_41.

[19] A. Kwanashie, R. W. Irving, D. F. Manlove, and C. T. S. Sng. Profile-based optimal matchings in the student/project allocation problem. In *Combinatorial Algorithms*, pages 213–225, 2015. doi: 10.1007/978-3-319-19315-1_19.

[20] D. F. Manlove, R. W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1):261–279, 2002. doi: 10.1016/S0304-3975(01)00206-7.

[21] F. Mémoli. Metric structures on datasets: Stability and classification of algorithms. In *Computer Analysis of Images and Patterns*, pages 1–33. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-23678-5_1.

[22] K. Menzel. Large matching markets as two-sided demand systems. *Econometrica*, 83(3):897–941, 2015. doi: 10.2307/43616957.

[23] M. K. Ng, M. J. Li, J. 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, 2007. doi: 10.1109/TPAMI.2007.53.

[24] A. Olaode, G. Naghdy, and C. Todd. Unsupervised image classification by Probabilistic Latent Semantic Analysis for the annotation of images. In *International Conference on Digital Image Computing: Techniques and Applications*, 2014. doi: 10.13140/2.1.1909.4086.

[25] B. Rastegari, P. Goldberg, and D. Manlove. Preference elicitation in matching markets via interviews: A study of offline benchmarks. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, AAMAS '16, pages 1393–1394, 2016. ISBN 978-1-4503-4239-1.

[26] 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. doi: 10.1086/261272.

[27] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan. Finding a 'kneedle' in a haystack: Detecting knee points in system behavior. In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166 – 171, 07 2011. doi: 10.1109/ICDCSW.2011.20.

[28] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007. ISSN 1574-0137. doi: 10.1016/j.cosrev.2007.05.001.

[29] X. Wu and V. Kumar. *The Top Ten Algorithms in Data Mining.* CRC, 2009. ISBN 9781420089646.

[30] 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. doi: 10.1109/ijcnn.2018.8488994.

[31] W. Zhao, H. Ma, and Q. He. Parallel k-means clustering based on MapReduce. In *Cloud Computing*, pages 674–679. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-10665-1_71.

[32] H. Zhou, Y. Zhang, and Y. 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. doi: 10.2991/iccia-17.2017.67.

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):

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

(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. □