Comparing established initialisation processes for the k-modes algorithm, and an alternative process utilising the hospital-resident assignment problem

Henry Wilde

December 11, 2017

1 The k-modes algorithm

The k-modes algorithm is a part of the family of clustering algorithms known as 'prototype-based clustering', and is an extension of the k-means algorithm for categorical data as set out in [3]. This work will outline the key differences between the two algorithms and then aim to examine how the initial cluster selection process has an impact on the efficiency and quality of the k-modes algorithm.

1.1 Notation

We will use the following notation throughout this work to describe our data set, points, clusters and representative points:

- All data is drawn from a universe $\mathbf{U}$ described by a set of m attributes $\mathbf{A} = A_1, \dots, A_m$.
- Our data set is a subset of our universe of size N, denoted by $\mathbf{X} = \{X_1, \dots, X_N\} \subseteq \mathbf{U}$. Each data point X_i can be represented as a vector:

$$X_i = [A_1 = x_{i,1}, A_2 = x_{i,2}, \dots, A_m = x_{i,m}], \quad i = 1, \dots, N$$

  where $x_{i,j}$ is the value of the j^{th} attribute of the i^{th} data point, X_i.
- We are aiming to partition $\mathbf{X}$ into a set of k distinct clusters $\mathbf{C} = \{C_1, \dots, C_k\}$. Each cluster C_l has a representative point associated with it $\bar{\mu}_l = [\mu_{l,1}, \dots, \mu_{l,m}]$.

1.2 Dissimilarity measure

The most immediate difference between k-means and k-modes is that they deal with different types of data, and so the metric used to define the distance between two points in our space will likely be different. With k-means, where our data set has all-numeric attributes, the Euclidean distance is often used. However, we do not have this sense of distance with categorical data. Instead, we utilise a dissimilarity measure - defined below - as our metric. It can be easily checked that this is indeed a distance measure.

Definition 1.1. Let $\mathbf{X}$ be a categorical data set and consider $X_1, X_2 \in \mathbf{X}$. We define the dissimilarity between X_1 and X_2 to be

$$d(X_1, X_2) = \sum_{j=1}^{j=m} \delta(x_{1,j}, x_{2,j}) \quad \text{where} \quad \delta(x, y) = \begin{cases} 0, & x = y \\ 1, & \text{else} \end{cases}$$

1.3 Representative points

Now that we have defined a metric on our space, we can turn our attention to what we mean by the representative point $\bar{\mu}_l$ of a cluster C_l. In k-means, we call μ_l a 'centroid' and define it to be the average of all points $X_i \in C_l$ by Euclidean distance. With categorical data, we use our revised distance measure defined in 1.1 to specify a representative point. We call such a point a mode of $\mathbf{X}$.

Definition 1.2. We define a mode of our set $\mathbf{X}$ to be any vector $\bar{\mu} = [\mu_1, \ldots, \mu_m] \in \mathbf{U}$ that minimises

$$D(\mathbf{X}, \bar{\mu}) = \sum_{i=1}^{i=n} d(X_i, \bar{\mu}) \tag{1}$$

NB: $\bar{\mu}$ is not necessarily in $\mathbf{X}$.

Theorem 1. Let $f_r(A_j = c_{s,j}) = \frac{n_{c_{s,j}}}{N}$ denote the relative frequency of category $c_{s,j}$ in $\mathbf{X}$ where $n_{c_{s,j}}$ is the number of points in $\mathbf{X}$ which take the s^{th} category $c_{s,j}$ of the j^{th} attribute A_j. Then

The function $D(\mathbf{X}, \bar{\mu})$ is minimised $\iff$ $f_r(A_j = \mu_j|\mathbf{X}) \geq f_r(A_j = c_{s,j}|\mathbf{X})$ for $\mu_j \neq c_{s,j}$ $\forall j = 1, \ldots, m$.

A proof of this theorem can be found in the Appendix of [3].

1.4 The cost function

We can use these two definitions to determine a cost function for our algorithm. Let $\mathbf{M} = \{\bar{\mu_1}, \ldots, \bar{\mu_k}\}$ be a set of k modes of $\mathbf{X}$, and let $W = (w_{i,l})$ be an $n \times k$ matrix such that:

$$w_{i,l} = \begin{cases} 1, & X_i \in C_l \\ 0, & otherwise \end{cases}$$

Then we define our cost function to be the summed within-cluster dissimilarity:

$$C(W, \mathbf{M}) = \sum_{l=1}^{l=k} \sum_{i=1}^{i=n} \sum_{j=1}^{j=m} w_{i,l} \delta(x_{i,j}, \mu_{l,j}) \tag{2}$$

1.5 The k-modes algorithm

Below is a practical implementation of the k-modes algorithm [3]:

1. Select k initial modes. The processes by which these are selected will be detailed in Section 2.
2. Allocate a point to the cluster whose mode is nearest according to 1.1. Update the mode of the cluster after each allocation.
3. When all points have been allocated, re-evaluate the dissimilarity of each point against the current modes. Reallocate points to the appropriate cluster if they are found to be less dissimilar to another mode. Update the modes of both the original and new clusters after each point is reallocated.
4. Go to 3 until no points move after a full cycle through the data set.

2 Initialisation processes

It has been shown that the initial choice of clusters impacts the final solution of the k-modes algorithm. This is typically controlled either by considering an alternative dissimilarity measure [2] or by changing the way that the k initial representative points are chosen [3] [1]. Two established methods of selecting these initial points are defined below.

2.1 The Huang method

In the most basic form of the k-modes algorithm, the k initial modes are chosen at random from $\mathbf{X}$. Below is an alternative method of selecting these modes to force diversity between them, as described in [3]:

1. Calculate the frequencies $f(c_{s,j})$ of all categories for each attribute $A_1, \ldots, A_m$ and arrange the categories $c_{s,j}$ in a matrix in descending order of frequency, breaking ties arbitrarily.
2. Assign the most frequent categories equally amongst k virtual modes $\bar{\mu_1}, \ldots, \bar{\mu_k}$.
3. Go through these modes in numerical order and select the record X_i most similar to $\bar{\mu_l}$. Replace $\bar{\mu_l}$ with X_i. Continue in this way until $\bar{\mu_k}$ is replaced. In these selections we maintain that $\bar{\mu_l} \neq \bar{\mu_t}$ for $l \neq t$ so as to avoid empty clusters.

A small example of this method is given below.

Example 1. Below are the first five rows of a random sample of 250 records from a data set used to determine the acceptability of a car. This dataset was chosen primarily for its number of attributes. However, it should be noted that one downfall of this particular data set is that some of the attributes could be considered as ordinal rather than purely categorical since there are clearly established and easily understandable differences between "high" and "low" prices, for instance.

Price	Maintenance	Doors	Passengers	Luggage	Safety
low	vhigh	2	5+	med	med
vhigh	high	2	4	big	med
high	med	2	2	small	low
vhigh	med	3	2	big	low
low	med	5+	2	big	low

The frequencies of our attributes' categories are given below:

Price	Maintenance	Doors	Passenger	Luggage	Safety
$f(c_{\text{low}}) = 61$	$f(c_{\text{low}}) = 53$	$f(c_2) = 71$	$f(c_2) = 81$	$f(c_{\text{small}}) = 88$	$f(c_{\text{low}}) = 76$
$f(c_{\text{med}}) = 63$	$f(c_{\text{med}}) = 66$	$f(c_3) = 71$	$f(c_4) = 85$	$f(c_{\text{med}}) = 78$	$f(c_{\text{med}}) = 91$
$f(c_{\text{high}}) = 63$	$f(c_{\text{high}}) = 50$	$f(c_4) = 53$	$f(c_{5+}) = 84$	$f(c_{\text{big}}) = 84$	$f(c_{\text{high}}) = 83$
$f(c_{\text{vhigh}}) = 63$	$f(c_{\text{vhigh}}) = 81$	$f(c_{5+}) = 55$			

Table 1: Frequencies of all attribute categories, $f(c_{s,j})$

Thus, from Table 1 we see that our category matrix is:

$$\begin{pmatrix} \text{vhigh} & \text{vhigh} & 3 & 4 & \text{small} & \text{med} \\ \text{high} & \text{med} & 2 & 5+ & \text{big} & \text{high} \\ \text{med} & \text{low} & 5+ & 2 & \text{med} & \text{low} \\ \text{low} & \text{high} & 4 & & & \end{pmatrix}$$

Acceptability is an attribute of this data which has been removed but indicates whether a car is one of 'very good', 'good', 'acceptable' or 'unacceptable'. From this we can suppose that we are looking for $k = 4$ clusters, and so, by distributing the most frequent categories 'equally' our initial set of modes is:

$$\begin{aligned} \mathbf{M} \ = \{\bar{\mu_1} = [\text{vhigh}, \text{med}, 5+, 4, \text{big}, \text{low}], \quad \bar{\mu_2} = [\text{high}, \text{low}, 4, 5+, \text{med}, \text{med}], \\ \bar{\mu_3} = [\text{med}, \text{high}, 3, 2, \text{small}, \text{high}], \quad \bar{\mu_4} = [\text{low}, \text{vhigh}, 2, 4, \text{big}, \text{med}]\} \end{aligned} \tag{3}$$

Now we would select the least dissimilar point in our data set to replace each $\bar{\mu_l} \in \mathbf{M}$ in numerical order according to Def 1.1 and continue with the rest of the algorithm.

2.2 The Cao method

Cao's method selects representative points by the average density of a point in the dataset as opposed to relative frequency of the dataset's attribute values. This algorithm is considered deterministic as there is no random element - unlike the standard or Huang methods - and so results are completely reproducible.

Definition 2.1. Consider a data set $\mathbf{X}$ with attribute set $\mathbf{A}$. Then the average density of any point $X_i \in \mathbf{X}$ with respect to $\mathbf{A}$ is defined as:

$$\text{Dens}(X_i) = \frac{\sum_{a \in \mathbf{A}} \text{Dens}_a(X_i)}{|\mathbf{A}|}, \quad \text{where} \ \ \text{Dens}_a(X_i) = \frac{|\{X_j \in \mathbf{X} : x_{i,a} = x_{j,a}\}|}{|\mathbf{X}|}$$

Remark. Note that we have $\frac{1}{|\mathbf{X}|} \leq \text{Dens}(X_i) \leq 1$, since for any $a \in \mathbf{A}$:

- If $|\{X_j \in \mathbf{X} : x_{i,a} = x_{j,a}\}| = 1$, then $\text{Dens}(X_i) = \frac{\sum_{a \in \mathbf{A}} \frac{1}{|\mathbf{X}|}}{|\mathbf{A}|} = \frac{|\mathbf{A}|}{|\mathbf{A}||\mathbf{X}|} = \frac{1}{|\mathbf{X}|}$.
- If $|\{X_j \in \mathbf{X} : x_{i,a} = x_{j,a}\}| = |\mathbf{X}|$, then $\text{Dens}(X_i) = \frac{|\mathbf{A}|}{|\mathbf{A}|} = 1$.

The Cao selection process is as follows:

1. Set $= \emptyset$ and calculate $\text{Dens}(X)$ for each $X \in \mathbf{X}$.
2. Add to the point $X_{i_1} \in \mathbf{X}$ which satisfies $\max_{i=1}^{|\mathbf{X}|}\{\text{Dens}(X_{i_1})\}$.
3. To find the second cluster point, select and add to the point $X_{i_2} \in \mathbf{X}$ which satisfies:
$$d(X_{i_2}, X_m) \times \text{Dens}(X_{i_2}) = \max_{i=1}^{|\mathbf{X}|}\{d(X_i, X_m) \times \text{Dens}(X_i) | X_m \in\}$$
4. If $|| < k$ go to 5. Otherwise, end.
5. Select any point $X_{i_j} \in \mathbf{X}$ such that:
$$d(X_{i_j}, X_m) \times \text{Dens}(X_{i_j}) = \max\{\min_{X_m \in}\{d(X_i, X_m) \times \text{Dens}(X_i) | X_i \in \mathbf{X}\}\}$$
Add this point to and go to 4.

3 The Gale-Shapley algorithm

In this work, we will consider the initial set of virtual modes found by the Huang method together with some subset $\tilde{\mathbf{X}} \subset \mathbf{X}$ as a matching game.

Definition 3.1. A matching game of size N is defined by two disjoint sets, S and R, each of size N. Each element of S and R has associated with it a preference list of the other set's elements. Any bijection M between S and R is called a matching. If the pair (s, r) are matched by M then we write $M(s) = r$.

Definition 3.2. A pair (s, r) blocks M if $M(s) \neq r$ but s prefers r to $M(s)$ and r prefers s to $M^{-1}(r)$.

Definition 3.3. A matching with no blocking pairs is said to be stable.

The Gale-Shapley algorithm is known to find a unique stable matching of a game of size N which is considered to be suitor-optimal. In this method we do not necessarily have equally sized sets for suitors and reviewers, and though we don't allow reviewers to be matched to multiple suitors, an extension to the standard algorithm must be used. This extension is based on that used by the National Resident Matching Program (see: `http://www.nrmp.org/matching-algorithm/`) to solve the hospital-resident assignment problem.

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

Given a set of k hospitals H - with respective capacities $c_{h_1}, \ldots, c_{h_k} \in \mathbb{Z}_+$ - and a set of $N \geq k$ residents R, let each $h \in H, r \in R$ have ranked preferences of their complementary set's elements. Then we solve this capacitated matching game with the following algorithm:

1. Set all hospitals and residents to be unmatched, i.e. $M = \{\}$.
2. Take any unmatched resident, r, and their most preferred hospital, h. If r's preference list is empty, remove them from consideration.
   - If h has space, i.e. $|M(h)| < c_h$, then append r to $M(h)$.
   - Otherwise, for each resident currently matched with h, $\tilde{r} \in M(h)$, if $r \notin M(h)$:
     - If h prefers r to $\tilde{r}$, remove $\tilde{r}$ from $M[h]$ so it is unmatched and append r to $M[h]$.
     - If not, remove h from r's preference list and leave r unmatched.
3. Go to 2 until there are no unmatched residents up for consideration.

Remark. This implementation requires all residents to be ranked by all hospitals, and will produce a matching such that no hospital is left without at least one resident.

4 The proposed method

With the algorithm described above, we can build an alternative initialisation process for the k-modes algorithm.

Let $\mathbf{X}$ be a dataset with attribute set $\mathbf{A}$, and let $\mathbf{M}$ be the set of virtual modes found by the Huang method up to Step 3. Then we construct the following capacitated matching game:

- The set of hospitals H is $\mathbf{M}$, and each hospital has capacity 1.
- The set of residents, R, is made up of the k least dissimilar points $X_{l,1}, \ldots, X_{l,k} \in \mathbf{X}$ to each $\bar{\mu_l} \in \mathbf{M}$.

- Each hospital's preference list is simply their addition to the set of residents in descending order of similarity.
- The preference lists of the residents is more complicated. In this initial implementation, we take their preference list to be the set of hospitals in ascending order with respect to dissimilarity. Though, as will be seen in Section 5, other ways of generating these lists (such as randomly) can provide different results.

Now, by applying the capacitated Gale-Shapley algorithm to this game, we find a resident-optimal matching M. Let our set of modes $\mathbf{M} := M^{-1}(H)$. That is, the l^{th} mode is the resident matched with $\bar{\mu}_l$ when the algorithm concludes.

5 Experimental results

To give comparative results on the quality of the initialisation processes defined in Sections 2 & 4, four well-known, categorical, labelled datasets - soybean, mushroom, breast cancer, and zoo - will be clustered with the k-modes algorithm. Then the typical performance measures of accuracy, precision, and recall will be calculated and summarised below. As a general rule, each algorithm will be trained on approximately two thirds of the respective dataset and tested against the final third.

Definition 5.1. Let a dataset $\mathbf{X}$ have k classes $C_1, \ldots, C_k$, let the number of objects correctly assigned to C_i be denoted tp_i, let fp_i denote the number of objects incorrectly assigned to C_i, and let fn_i denote the number of objects incorrectly not assigned to C_i. Then our performance measures are defined as follows:

$$\textit{Accuracy}: \quad \frac{\sum_{i=1}^{k} tp_i}{|\mathbf{X}|}, \quad \textit{Precision}: \quad \frac{\sum_{i=1}^{k} \frac{tp_i}{tp_i + fp_i}}{k}, \quad \textit{Recall}: \quad \frac{\sum_{i=1}^{k} \frac{tp_i}{tp_i + fn_i}}{k}$$

5.1 The datasets

A bit on the structure of each dataset and links to access them.

5.2 Results

Tables of results for each dataset and each initialisation process. Credit to `https://github.com/nicodv/kmodes` for the Python implementation of both the Huang and Cao processes, as well as the k-modes algorithm itself.

6 Resident preference lists

Some examples and hopefully some mathematical reasoning to justify that certain choices of preference lists reduce down to near equivalent results of the Huang method (or others). This then suggests the proposed method is in fact a generalisation of the other method(s).

References

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

[2] Huang Z He Z Ng MK Li MJ. "On the impact of dissimilarity measure in k-modes clustering algorithm". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.3 (Mar. 2007).

[3] Huang Z. “Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values”. In: *Data Mining and Knowledge Discovery* 2.3 (Sept. 1998), pp. 283–304. DOI: 10.1023/A:1009769707641. URL: https://doi.org/10.1023/A:1009769707641.