Analyzing Clustering Algorithms for Mixed Type Data

STATS 4T06

Hanwen Ju
Supervisor: Dr. Paul McNicholas
Department of Mathematics & Statistics
McMaster University

March 31 2023

Abstract

The clustering of mixed data has garnered increased attention in recent years. This senior research project provides an in-depth review of three classic clustering algorithms: k-means clustering for mixed datasets (KMCMD), AUTOCLASS, and frequency neuron mixed self-organizing map (FMSOM). To gain deeper insights into the strengths and limitations of these algorithms across various scenarios, a practical analysis was conducted on three mixed type datasets. Additionally, we introduce a novel distance measurement method that utilizes weighted Jaccard similarity to develop two clustering algorithms for frequency neuron mixed self-organizing maps (FMSOM). Ultimately, the insights gained from this work can be used to help the development of clustering algorithms better suited for handling mixed type data in the future.

Acknowledgements

Firstly, I would like to express my deep gratitude to my supervisor Dr. Paul McNicholas for giving me the opportunity of this senior research project. I am grateful for all the invaluable guidance and support from him throughout the project. Additionally, I would like to thank all my instructors from the Department of Mathematics & Statistics and the Department of Computing and Software at McMaster University for their unwavering support and encouragement during my undergraduate studies.

Contents

Chapter 1

Introduction

Clustering is one of the essential techniques in data mining, which aims to find relations and hidden patterns behind data. The goal is to partition data objects into groups based on their similarity, while ensuring that objects within a group are more alike to each other but more dissimilar to objects in other groups. However, most clustering algorithms are only suitable for handling numerical or categorical data separately but not both. In reality, many data sets are mixed typed, and clustering these data types becomes essential.

Clustering methods for mixed-type data can be categorized into six main groups: partition-based algorithms, hierarchical clustering algorithms, incremental clustering algorithms, model-based clustering algorithms, fuzzy clustering algorithms, and artificial neural networks clustering algorithms. Partition-based clustering algorithms, such as K-Prototypes (cf.Huang 1997a), K-Means Clustering for Mixed Datasets (KMCMD) (cf.Ahmad and Dey 2007), build clusters based on partition and follow the structure of K-means. These algorithms are usually fast and easy to interpret. Hierarchical clustering algorithms, such as Similarity-Based Agglomerative Clustering (SBAC) (cf.Li and Biswas 2002), are used to create a hierarchical tree-like structure that merges or divides data objects into clusters based on similarity measurement. Incremental clustering algorithms, such as CAVE (cf.Noorbehbahani, Mousavi, and Mirzaei 2015) and MSOINN (cf.Shen and Hasegawa 2008), update the existing clustering structure rather than recalculating the entire clustering structure when new data points are added. Model-based clustering algorithms are based on statistical models and typically use a mixture model. Examples include AUTOCLASS (cf.Cheeseman, Stutz, et al. 1996a), Mixture of Latent Variables Model (cf.Browne and McNicholas 2012). Fuzzy clustering algorithms assign a degree of membership to each data point for each cluster. Examples include KL-FCM-GM (cf.Honda and Ichihashi 2005), Fuzzy K-Prototype (cf.Ji et al. 2012). Artificial

neural networks clustering algorithms are based on the idea of competitive learning techniques. For instance, GMixSOM (cf.Tai and Hsu 2012), FMSOM (cf.Del Coso et al. 2015). While each of these clustering methods has its strengths and weaknesses, they all aim to identify patterns and relationships within mixed-type datasets.

The goal of this work is to provide an in-depth review of three clustering algorithms: KMCMD, AUTOCLASS and FMSOM. We performed a practical analysis on four mixed type datasets to evaluate the effectiveness of these algorithms. Through this analysis, we aim to gain a better understanding of how these algorithms perform in different scenarios and identify their strengths and weaknesses. Furthermore, we introduced a novel distance measurement approach that utilizes weighted Jaccard similarity, and applied this distance function to developed two algorithms that extend the k-means (Hartigan, 1979) and k-medoids (Park et al., 2009) clustering techniques. These algorithms have been demonstrated to be effective and valid for clustering the FMSOM. Ultimately, the insights gained from this work can be used to inform the development of clustering algorithms better suited for handling mixed-type data in the future.

The rest of this thesis is organized as follows: Chapter 2 provides a detailed literature review of the selected clustering algorithms designed to handle mixed type data. In Chapter 3, we discussed our approach in detail, including various techniques for evaluating performance. Chapter 4 discussed about the proposed new distance function and introduced two clustering algorithms specifically designed for use with the FMSOM. Chapter 5 presents the results of testing the selected clustering algorithms on three datasets, and provides an analysis of parameter selection. And Chapter 6 is the conclusion and extension of this thesis.

Chapter 2

Literature Review

2.1 K-Means Clustering for Mixed Datasets (KMCMD)

KMCMD is a partition-based method for clustering mixed-type data sets. It was believed that the proposed distance measure by Huang (cf.Huang 1997b) is not perfect for mix-typed data since the distance function and cluster center representation both have information loss. Ahmad and Dey 2007 proposed a new cost function that improves Huang's cost function:

$$\zeta = \sum_{i=1}^{n} \vartheta(o_i, C_j), \tag{2.1}$$

where

$$\vartheta(o_i, C_j) = \sum_{t=1}^{m_r} (w_t(o_{it}^r - C_{jt}^r))^2 + \sum_{t=1}^{m_c} \Omega(o_{it}^C, C_{jt}^C)^2. \tag{2.2}$$

$\vartheta(o_i, C_j)$ is the distance between object o_i and C_j, which is the sum of the total weighted Euclidean distance for all numerical attributes and the total distance by the proposed method for all categorical attributes. And w_t represents the significance of the tth attribute. The distance for two categorical attribute values depends on the co-occurrence of all other attributes. It can compute by the average of $\delta^{ij}(x, y)$ for all $j \neq i$ where:

$$\delta^{ij}(x, y) = P_i(w/x) + P_i(\sim w/y) - 1. \tag{2.3}$$

here $\delta^{ij}(x, y)$ is the proposed distance measurement for x and y with respect to attribute A_j. $P_i(w/x)$ represents the probability of an object having a value that falls into the subset w in attribute A_j given value x for attribute A_i. Similarly, $P_i(\sim w/x)$ represents the conditional probability of an object having a value that falls into the complement of w in attribute A_j given value y for attribute A_i. Since both $P_i(w/x)$ and $P_i(\sim w/x)$

are probability between 0 and 1, the value of $\delta^{ij}(x, y)$ can be restricted between 0 and 1 by minus 1. Furthermore, the attributes with good separation of co-occurrence of values into different groups play a more important role in clustering. In other words, if an attribute has an overall large $\delta(x, y)$ for all pairs of x and y, then it has more significance. For instance, if one attribute has all the same values, the significance w_i is 0. It makes sense since this attribute does not help at all with clustering. Thus, the significance of numeric attributes can be computed in the following way:

$$w_i = \frac{\sum_{r=1}^{S} \sum_{s>r}^{S} \delta(u[r], u[s])}{S(S-1)/2}. \tag{2.4}$$

where we discretize all numerical attributes into S intervals, denote as $u[1], u[2], ..., u[S]$, it is basically the average distance for all pairs $(u[r], u[s])$. Instead of using mode, the center of cluster C for categorical attributes records the overall distribution without information loss:

$$1/N_c \left\langle (N_{1,1,c}, N_{1,2,c}, ..., N_{1,p1,c}), (N_{2,1,c}, N_{2,2,c}, ..., N_{2,p2,c}), ..., (N_{m,1,c}, N_{m,2,c}, ..., N_{m,pm,c}) \right\rangle \tag{2.5}$$

where N_c is the number of objects in cluster C, $N_{i,k,c}$ represents the number of objects in cluster C with attribute value k in attribute A_i, and assuming attribute A_i has p_i distinct values. And the cluster center for numerical data attributes is simply the normalized mean. The full expression of cluster center can be represented as its concatenation. Therefore, the distance from the object to its cluster center is simply computed by Euclidean distance between its attribute value and cluster normalized mean for numerical attributes. For categorical attributes, the distance between X and cluster center C for attribute A_i is computed by a weighted sum of distance for all other possible values of A_i:

$$\Omega(X, C) = (N_{i,1,c}/N_c) \cdot \delta(X, A_{i,1}) + (N_{i,2,c}/N_c) \cdot \delta(X, A_{i,2}) +, ..., +(N_{i,pi,c}/N_c) \cdot \delta(X, A_{i,pi}) \tag{2.6}$$

KMCMD follows the original k-mean procedures. The algorithm randomly allocates all data objects to k clusters at first, and each cluster center is computed, then assigning each object to its nearest cluster. Finally, repeat the last two steps until they converge.

2.2 AUTOCLASS

AUTOCLASS is a model-based clustering algorithm using the Bayesian approach to obtain the best class distribution for the data. The observed data is modeled as a

finite mixture of distributions, and the likelihood of the whole data set is written as the product over all data objects:

$$P(\overrightarrow{X}|\overrightarrow{V},T,S,I)=\prod_i\left[\sum_j(\pi_j\prod_k P(X_{ik}|X_i\in C_j,\overrightarrow{V_{jk}},T_{jk},S,I))\right], \tag{2.7}$$

where $\overrightarrow{V}$ is the parameter sets of the model, T represent the model form of feature distribution, S,I are treated as some other implicit information. and the mixture proportion π_j. The main idea of the algorithm is to maximize the posterior of parameters $\overrightarrow{V}$ and the model form T by using non-informative priors. In the first stage, given a model form T and data to maximum a posteriori of parameter values:

$$P(\overrightarrow{V}|X,T,S,I)=\frac{P(\overrightarrow{V},X|T,S,I)}{P(X|T,S,I)}. \tag{2.8}$$

Given the number of classes, AUTOCLASS performs a search over $\overrightarrow{V}$, and a variation of EM algorithm[6] is applied to find the best parameter values. For E-step, the weighted assignment w_{ij} is computed, then using these weighting assignments to estimate the parameters in M-step. Re-estimation between E-step and M-step moves the parameters toward a mutually predictive and locally maximal stationary pointCheeseman, Stutz, et al. 1996b. Therefore, those estimated parameters are used to approximate and maximum a posteriori of the model form T:

$$P(T|X,S,I)\propto P(X|T,S,I), \tag{2.9}$$

The posterior of the current model form T is recorded. Next, selectively change the model form T and redo the above two-step process to get the posterior of the new model T. The best model form T will retain at the end of this two-step cycling process by comparing their posterior probability.

For categorical attributes, assuming categorical attribute k has L_k unique values, AUTOCLASS uses Bernoulli distribution with Dirichlet conjugate prior:

$$P(X_{ik}=l|X_i\in C_j,\overrightarrow{V_{jk}},T_{jk},S,I)=q_{jkl}, \tag{2.10}$$

$$P(q_{jk1},...,q_{jkL_k}|T_{jk},S,I)=\frac{\Gamma(L_k+1)}{\left[\Gamma(1+\frac{1}{L_k})\right]^{L_k}}\prod_{l=1}^{L_k}q_{jkl}^{\frac{1}{L_k}}. \tag{2.11}$$

For numerical attributes, let k_{max} and k_{min} be the max and min value of attribute k,

AUTOCLASS uses Gaussian densities with a uniform prior on the means, and a Jeffrey's prior on standard deviation:

$$P(X_{ik}|X_i \in C_j, \mu_{jk}, \sigma_{jk}, T_{jk}, S, I) = \frac{1}{\sqrt{2\pi}\sigma_{jk}} e^{-\frac{1}{2}(\frac{x_{ik}-\mu_{jk}}{\sigma_{jkl}})^2}. \tag{2.12}$$

$$P(\mu_{jk}|T_{jk}, S, I) = \frac{1}{\mu_{k_{max}} - \mu_{k_{min}}}, \qquad P(\sigma_{jk}|T_{jk}, S, I) = \sigma_{jk}^{-1}\left[\log\frac{\sigma_{k_{max}}}{\sigma_{k_{min}}}\right]^{-1}. \tag{2.13}$$

AUTOCLASS can also find the best number of classes. AUTOCLASS starts with fewer classes than the expected number of classes. If resulting classes have significant posterior probability π_j, then increase the number of classes until some classes have negligible π_j. At last, the classes with negligible π_j are removed, and the remaining classes are considered best classes distribution.

2.3 Frequency Neuron Mixed Self-Organizing Map (FMSOM)

FMSOM is an artificial neural network clustering algorithm that builds upon the Self-Organizing Map (SOM) (cf.Kohonen 1990) technique. During the training process, the algorithm adjusts the weights of the neurons to represent the input data, such that neurons that are nearby in the grid respond to similar input patterns. The algorithm can be divided into 4 stages:

- Initialization Process:
  A two-dimensional SOM topology is created, and the map is initialized by assigning random weights to each neuron. Number of iterations, initial radius, neighbourhood decay are also initialized.

- Competitive Process:
  or each data point, the algorithm traverses each neuron and calculates the dissimilarity or similarity between the data point and the neuron's weight vector. The neuron with the smallest dissimilarity is selected as the best matching unit (BMU). The similarity function is defined as:

  $$d(X_p, W_i) = d_n(X_p, W_i) + d_c(X_p, W_i), \tag{2.14}$$

  where $d_n(X_p, W_i)$ is the Euclidean distance for numerical attributes. And $d_c(X_p, W_i)$ is the categorical dissimilarity which measured by the probability of the weight of

the neuron not holding the category present on the input, and it is defined as follows:

$$d_n(X_p, W_i) = \sqrt{\sum_{z=1}^{n}(X_{pz} - W_i)^2}, \tag{2.15}$$

$$d_c(X_p, W_i) = \sqrt{\sum_{z=n+1}^{k}(1 - W_{iz}[X_{pz}])^2}. \tag{2.16}$$

- Cooperative Process:
  After identifying the best matching unit (BMU) in a self-organizing map (SOM), the neighbouring neurons around the BMU are also required to be updated. This is typically achieved using a Gaussian neighbourhood, which is defined as follows:

$$h(s, c(X_p), i) = \exp\left(\frac{-d^2}{2\sigma(s)^2}\right), \tag{2.17}$$

  where d is the distance in the lattice from the winer unit(BMU) to unit i. And $\sigma(s)$ is the radius which decreases exponentially with iterations.

$$\sigma(s) = \sigma(1)\exp\left(\frac{-s}{T}\right), \tag{2.18}$$

  The initial radius of the neighborhood function is denoted by $\sigma(1)$, and as the algorithm progresses, the radius is gradually reduced using a constant parameter T. A larger value of T results in a slower decrease in the radius.

- Adaptive Process:
  In the adaptive stage, the weight vector for each neuron is updated. For numerical features, the updated formula for iteration $s+1$ is given by the means:

$$W_{in}(s+1) = \frac{\sum_{p=1}^{P} h(s, c(X_p), i)X_{pn}}{\sum_{p=1}^{P} h(s, c(X_p), i)}. \tag{2.19}$$

  For categorical features, update the relative frequencies for each possible category in that attribute:

$$W_{ik}(s+1) = \left\{F(\alpha_k^1, W_{ik}(s)), F(\alpha_k^2, W_{ik}(s)), \ldots, F(\alpha_k^r, W_{ik}(s))\right\}, \tag{2.20}$$

where $\alpha_k^1, \alpha_k^2, \ldots, \alpha_k^r$ are the possible categories of attribute k, the relative frequency for attribute k is defined as:

$$F(\alpha_k^r, W_{1k}(s)) = \frac{\sum_{p=1}^{P} h(s, c(X_p), i | X_{pk} = \alpha_k^r)}{\sum_{p=1}^{P} h(s, c(X_p), i)}. \tag{2.21}$$

Finally, the weight vectors are updated for iteration $s + 1$. The whole weighting vector of unit i can be treated as a prototype, defined as:

$$W_i(s+1) = \{W_{in}(s+1), W_{ik}(s+1)\} \, [n = 1...N, k = 1...K]. \tag{2.22}$$

The algorithm then repeats the same four steps, continuing to update the weight vectors until convergence is reached. Convergence is defined as the point at which the best matching unit (BMU) for each data point no longer changes.

Chapter 3

Methodology

3.1 Performance Evaluation

3.1.1 Accuracy (Acc)

Clustering accuracy is the most straight forward method to evaluate the clustering performance. The accuracy of clustering can be evaluated by comparing clustering results with a pre-defined response variable. The clustering accuracy is defined as:

$$\text{Acc} = \frac{\sum_{i=1}^{k} a_i}{n}. \tag{3.1}$$

where a_i denotes the number of objects that are correctly assigned to cluster i based on their corresponding true class label, and n is the total number of data objects, while k indicates the number of clusters.

3.1.2 Adjusted Rand Index (ARI)

The adjusted Rand index (ARI) (cf.Rand 1971) is an extension of the Rand Index, which measures the agreement between the true class labels and the predicted clustering label. The formula for ARI is defined as:

$$\text{ARI} = \frac{\text{Index} - \text{Expected Index}}{\text{Max Index} - \text{Expected Index}}. \tag{3.2}$$

The ARI adjusts for chance agreement that can occur when randomly assigning samples to clusters. A score of 1 indicates perfect agreement between the two clusterings, a score closes to 0 indicates a random partition.

3.1.3 Quantization error

Quantization error is a commonly used index in Self-Organizing Maps that measures the average distance between a data sample and its closest matching unit within the SOM. The distance is usually measured by euclidean distance, here we replace it with the general distance function $\delta(x_i, \mathbf{m}_{b_i})$ used for selecting BMU in SOMs, we define the Quantization error as:

$$\text{QE} = \frac{1}{N}\sum_{i=1}^{N}\delta(x_i, \mathbf{m}_{b_i}). \tag{3.3}$$

3.1.4 Category Utility (CU) and CV-value

The category utility (CU), first introduced by (cf.Corter and Gluck 1992), has been found to be a valuable validity index for clustering categorical data. The goal of CU is to maximize the probability of a feature value belonging to a cluster and the probability of an instance in a cluster having a particular feature value. The Categoriy utility(CU) is defined as:

$$\text{CU} = \sum_k \frac{|C_k|}{|D|}\sum_i\sum_j \left[P(A_i = V_{ij}|C_k)^2 - P(A_i = V_{ij})^2\right], \tag{3.4}$$

where $|C_k|$ represents the size of cluster C_k and D is the total number of observations. $P(A_i = V_{ij}|C_k)$ is the conditional probability of attribute i having value V_{ij} in cluster C_k. The higher CU value, the better clustering performance. To extend the applicability of the CU measure to datasets with mixed type attributes, Hsu et al proposed a validity index named the CV value in (cf.Hsu and Chen 2007), and is defined as follows:

$$\text{CV} = \frac{\text{CU}}{1+\sigma^2}, \tag{3.5}$$

where

$$\sigma^2 = \sum_k \frac{1}{|C_k|}\sum_i\sum_j (V_{i,j}^k - V_{i,avg}^k)^2. \tag{3.6}$$

The CV value simply combines the CU measure and variance. A higher CV value implies a better clustering result.

3.1.5 k-fold Cross-Validation

Cross-validation (cf.Tarekegn, Michalak, and Giacobini 2020) is a powerful technique for evaluating the performance and generalization ability of a model. It involves partitioning

the data into k equally sized subsets, where one subset is designated for testing the model's performance, while the remaining $k-1$ subsets are used for training the model. Repeated this process k times, with each subset being used for testing exactly once. During each iteration, different external and internal performance metrics can be applied during the testing phase.

3.2 Clustering by KMCMD

The KMCMD algorithm was successfully implemented in R 4.2.2 for this project. To use the algorithm, the number of clusters k must be specified as an input. KMCMD automatically calculates the significance of each numerical attribute and generates a dissimilarity table for all possible values for categorical attributes. The algorithm then assigns each data point to a cluster and returns both the cluster assignments and cluster center information. In addition, the cluster center information generated by KMCMD can be useful for discovering the distribution of each cluster and identifying hidden patterns in the data.

3.3 Clustering by AUTOCLASS

The AUTOCLASS algorithm was originally implemented in C by NASA. However, due to time constraints, we opted not to implement it ourselves. Instead, we used a Python package called Autoclasswrapper (cf.Camadro and Poulain 2019), which provides a convenient wrapper for using Autoclass C. AUTOCLASS is a fuzzy clustering technique, it returns the probability of each data point belonging to each cluster. Also, it automatically determines the best number of clusters, and returns a hierarchical structure of clusters that can be visualized in a dendrogram. Therefore, we can cut the dendrogram into the clusters we want to perform further analysis.

3.4 FMSOM

3.4.1 Training FMSOM and Parameter selection

The FMSOM algorithm was successfully implemented in Python 3.11.1 for this project. While training an FMSOM, the following parameters should be specified for the function: map size, the initial radius $\sigma(1)$, constant T, and neighborhood function. Vesanto (cf.Rojas, Joya, and Catala 2015) proposes that the ideal size of the map should be $\lceil 5\sqrt{\mathrm{N}} \rceil$, where N represents the number of data points. However, in practice, this rule

of thumb may not always be perfect for FMSOM. We observed a relatively larger map size can have a better performance. The selection of initial radius $\sigma(1)$ and T is also not specified in the paper of FMSOM, different $\sigma(1)$ and T are tried in our simulation studies. Usually, parameters $\sigma(1)$ and T are depending on the map size. We tend to pick a relatively larger $\sigma(1)$ since it covers a larger region of the grid, then more units around BMUs during the cooperative process are affected. A parameter selection was conducted based on the evaluation methods described above in the later chapter.

3.4.2 FMSOM visualization and evaluation

One of the key advantages of SOMs is the ease of visualizing results in two-dimensional space. This is often accomplished using techniques such as U-matrix, clustering map, and heat map. The original paper on FMSOM only used a majority voting map and heat maps. For the majority voting map, each unit receives a color according to the most prominent class. If a unit has no data points assigned to it or in other words it is not BMU of any data points, then it is classified as "UNDEFINED" with a distinct color. To produce heat maps, we simply map each element of the weighting vector onto the two-dimensional map. For numerical variables, it is simply the weights. For categorical variables, we pick the categorical value with the highest probability to represent the node. One advantage of this method is that it highlights the significant values on the map. This can be useful for users as it helps them to detect correlations between different properties in the input data.

The authors in the original paper evaluated the performance of FMSOM using k-fold cross-validation with accuracy as the primary metric. Specifically, we counted each input vector as an error if its class was not the same as the most frequent class of its best matching unit (BMU), repeated this process k times, and use the average as the quality measure. Since accuracy can be sensitive to the class imbalance in the dataset, we use additional metrics, such as adjusted Rand index (ARI), quantization error, and CV value, to provide a more comprehensive evaluation of performance.

Chapter 4

Extended FMSOM

4.1 Limitations of FMSOM

FMSOM is a powerful unsupervised learning technique, but it still has some limitations. One significant challenge is evaluating its performance, which requires the use of cross-validation since predictions are based on the majority voting of true labels in the training set. This process is more akin to a classification algorithm rather than a clustering algorithm. In the regular SOM algorithms for numerical data, Vesanto (cf.Vesanto and Alhoniemi 2000) proposed a method that uses k-means and hierarchical agglomerative clustering to cluster the map. Another approach involves manually selecting the clusters based on the U-matrix (cf.Ultsch 2003). However, the FMSOM algorithm generates a weighting vector that comprises both numerical values and categorical frequency, as described in Equation 2.22. There is currently no similarity function available to measure the similarity between each weighting vector. As a result, the methods mentioned above cannot be applied to FMSOM. In the following section, we present an innovative solution to address the limitations of FMSOM for clustering tasks. We propose a novel distance function that is specifically designed for clustering the map generated by FMSOM.

4.2 The proposed distance function

4.2.1 Weighted Jaccard similarity and distance

Weighted Jaccard similarity (cf.Ruzicka 1953) and distance are extensions of the Jaccard similarity. In the basic Jaccard similarity and distance, only presence or absence are considered. Instead, the weighted Jaccard similarity and distance assign weights to each element based on their relative importance or relevance. If $x = (x_1, x_2, ..., x_n)$ and

$y = (y_1, y_2, ..., y_n)$ with all $x_i, y_i \geq 0$, then the weighted Jaccard similarity is defined as:

$$J_{\mathcal{W}}(x, y) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}, \tag{4.1}$$

and Jaccard distance is defined as:

$$d_{J\mathcal{W}}(x, y) = 1 - J_{\mathcal{W}}(x, y). \tag{4.2}$$

By taking into account the significance or weight of individual elements within the sets, the weighted Jaccard distance can offer a more precise assessment of similarity compared to the standard Jaccard distance, particularly in cases where certain elements hold greater importance than others.

4.2.2 Distance between two weighting vectors

The weighting vector generated by FMSOM has two components, one for numerical attributes, one for categorical attributes. Let $\hat{W}_i$ represents the weighting vector for neuron i after FMSOM training converge:

$$\hat{W}_i = \left\{ \hat{W_{i1}^r}, \hat{W_{i2}^r}, \ldots, \hat{W_{iN}^r}, \hat{W_{i1}^c}, \hat{W_{i2}^c}, \ldots, \hat{W_{iK}^c} \right\}. \tag{4.3}$$

where $\hat{W_{in}^r}$ represents the weighting vector for the nth numerical attribute, while $\hat{W_{ik}^c}$ represents the weighting vector for the kth categorical attribute. In total, there are N numerical attributes and K categorical attributes. Specifically, $\hat{W_{in}^r}$ contains one real number between 0 and 1, $\hat{W_{ik}^c}$ is consists of the relative frequencies for each possible category in attribute k, defined as:

$$\hat{W_{ik}^c} = \left\{ \hat{F}_i(\alpha_k^1), \hat{F}_i(\alpha_k^2), \ldots, \hat{F}_i(\alpha_k^r) \right\}. \tag{4.4}$$

where $\alpha_k^1, \alpha_k^2, ..., \alpha_k^r$ are the possible categories of attribute k. By taking the advantage of weighted Jaccard distance, we proposes the distance between two weighting vectors of FMSOM. Let $\hat{W}_i$ and $\hat{W}_j$ be ith and jth neuron of the map generated by FMSOM, and assume there're N numerical and K categorical attributes. The distance between $\hat{W}_i$ and $\hat{W}_j$ is defined as follows:

$$D(\hat{W}_i, \hat{W}_j) = \sum_{n=1}^{N} (\hat{W_{in}^r} - \hat{W_{jn}^r})^2 + \sum_{k=1}^{K} d_{J\mathcal{W}}(\hat{W_{ik}^c}, \hat{W_{jk}^c})^2. \tag{4.5}$$

The distance function defined in Eq.4.5 is consist of two components, the left part $\sum_{n=1}^{N}(\hat{W_{in}^r} - \hat{W_{jn}^r})^2$ is simply the total squared distance between numerical part of the weighting vectors. $\sum_{k=1}^{K} d_{J\mathcal{W}}(\hat{W_{ik}^c}, \hat{W_{jk}^c})^2$ denotes the total squared weighted Jaccard distance between each categorical part of weighting vectors.

4.3 The proposed algorithm

Based on the proposed distance function above, we proposes two partition-based algorithms to cluster FMSOM. The first algorithm is an extension of the k-means algorithm that uses our proposed distance function on the weighting vectors. As the k-means algorithm is sensitive to outliers, we have also developed another variation that extends the k-medoids algorithm.

4.3.1 A k-means clustering algorithm for FMSOM

Our first clustering algorithm for FMSOM shares similarities with the generic k-means algorithm, with a modified distance function and a modified cluster center. The goal is to minimize the following cost function:

$$\mathcal{C} = \sum_{i=1}^{M}\sum_{j=1}^{J} r_{ij} D(\hat{W}_i, C_j), \tag{4.6}$$

where

$$D(\hat{W}_i, C_j) = \sum_{n=1}^{N}(\hat{W_{in}^r} - C_{jn}^r)^2 + \sum_{k=1}^{K} d_{J\mathcal{W}}(\hat{W_{ik}^c}, C_{jk}^c)^2, \tag{4.7}$$

and r_{ij} is the indicator function, which represent assign the ith neuron to the closest cluster center. Formally, it can be defined as:

$$r_{ij} = \begin{cases} 1, & \text{if } l = \arg\min_j D(\hat{W}_i, C_l). \\ 0, & \text{otherwise.} \end{cases} \tag{4.8}$$

The cluster center C_j can be viewed as a prototype that contains a total of $N + K$ elements. To be more precise, the first N elements represent the cluster center for numerical attributes, and are determined by calculating the mean of the weights of all weighting vectors that assigned to the jth cluster. Meanwhile, for the remaining K elements, we compute the mean of the relative frequency for each possible category variable across all categorical attributes. Hence, the k-means algorithm for FMSOM is presented below:

Algorithm 1 A k-means clustering algorithm for FMSOM

Input: $\hat{W}$ (a set of all weighting vectors); k (number of clusters)

Output: K clusters with its centroid.

initialization Randomly select K weighting vectors from $\hat{W}$ as initial cluster centers

repeat

(Re)assign each weighting vector in $\hat{W}$ to closest cluster centroid using proposed distance function $D(\hat{W}_i, C_j)$, and generate k partitions $S_{C_1}, S_{C_2}, \ldots, S_{C_k}$;

Compute new centroid $C_j = \left\{C^r_{j1}, C^r_{j2}, \ldots, C^r_{jN}, C^c_{j1}, C^c_{j2}, \ldots, C^c_{jK}\right\}$ of each cluster, where

$$C^r_{jn} = \frac{\sum_i r_{ij} \hat{W}^r_{in}}{\sum_i r_{ij}}, \tag{4.9}$$

$$C^c_{jk} = \left\{ \frac{\sum_i r_{ij} \hat{F}_i(\alpha^1_k)}{\sum_i r_{ij}}, \frac{\sum_i r_{ij} \hat{F}_i(\alpha^2_k)}{\sum_i r_{ij}}, \ldots, \frac{\sum_i r_{ij} \hat{F}_i(\alpha^r_k)}{\sum_i r_{ij}} \right\}. \tag{4.10}$$

until *No more change in cluster assignment*;

4.3.2 A k-medoids clustering algorithm for FMSOM

The k-medoids clustering algorithm for FMSOM operates exactly like k-means but, instead of taking the mean value of objects in a cluster as centroid, it chooses the most centrally located object as centroid within the cluster. This difference results in a more robust clustering solution that is less sensitive to outliers. Despite this difference, both algorithms use the same cost function to optimize cluster assignment, making k-medoids a viable alternative to k-means for clustering tasks. The method is presented in Algorithm 2.

4.3.3 Random initialization

Both k-means and k-medoids clustering algorithms are highly dependent on the initial selection of centroids, and a suboptimal initialization may result in poor cluster quality. To mitigate this issue, we can use the k-means++ (cf. Arthur and Vassilvitskii 2007) initialization method, which selects the first centroid randomly and then iteratively selects the next centroid to be the point farthest away from the previously selected centroids. By using this method, we can ensure that the centroids are well-spaced and represent different regions of the data, leading to more accurate and stable cluster assignments.

Algorithm 2 A k-medoids clustering algorithm for FMSOM

Input: $\hat{W}$ (a set of all weighting vectors); k (number of clusters)

Output: k clusters with its centroid.

initialization Randomly select K weighting vectors from $\hat{W}$ as initial cluster centers

repeat

(Re)assign each weighting vector in $\hat{W}$ to closest cluster centroid using proposed distance function $D(\hat{W}_i, C_j)$, and generate k partitions $S_{C_1}, S_{C_2}, \ldots, S_{C_k}$;

Compute new centroid $C_j = \hat{W}_k$, where

$$k = \arg\min_{k} \sum_{\hat{W}_i \in S_{C_j}} D(\hat{W}_i, \hat{W}_k). \tag{4.11}$$

until *No more change in cluster assignment*;

4.4 FMSOM Clustering Process

To cluster data using FMSOM, we follow a general two-level process. Firstly, we normalize numerical attributes to ensure they are on the same scale. Next, we perform the first-level FMSOM training, which produces weighting vectors for each neuron. Then, we proceed to the second-level clustering, which involves grouping the $m \times n$ neurons into k clusters. The process of clustering data into k groups is illustrated in Figure 4.1 using a flowchart.

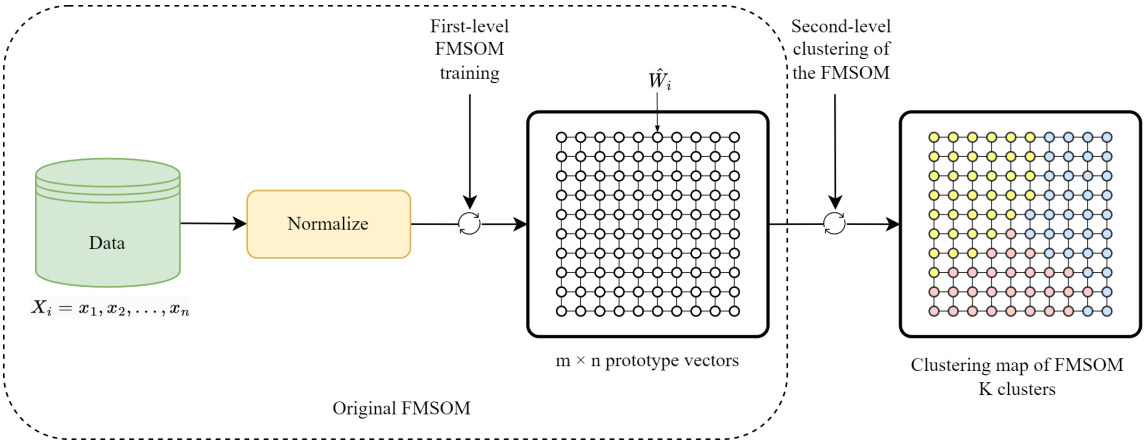


Figure 4.1: Flowchart of FMSOM Clustering Process.

Chapter 5

Simulation Studies

5.1 Acute Inflammations Data

The Acute Inflammations Data is aimed at developing an expert system algorithm for the presumptive diagnosis of two urinary system diseases - acute inflammations of the urinary bladder and acute nephritises. This dataset serves as an illustrative example of how to diagnose these diseases using an expert system. In all, there are 120 patients samples: 59 patients with inflammation of the urinary bladder, while the other 61 samples do not. Each patient sample consists of six symptom variables - one numerical variable for body temperature and five categorical variables with binary values of "yes" or "no." We use the methods presented above to perform clustering without using response variables.

The clustering performance of KMCMD is good (AC = 0.83; ARI = 0.63; Table 5.3). The cluster centers information obtained from KMCMD is presented in Table 5.1. It is evident from the table that there is a significant difference between groups $C1$ and $C2$. Group $C1$ shows a slightly higher body temperature and frequently has urine pushing, and all has micturition pains. On the other hand, group $C2$ does not exhibit symptoms of nausea and micturition pains. These observations suggest that micturition pains can be a useful indicator in determining the presence of inflammation in the urinary bladder.

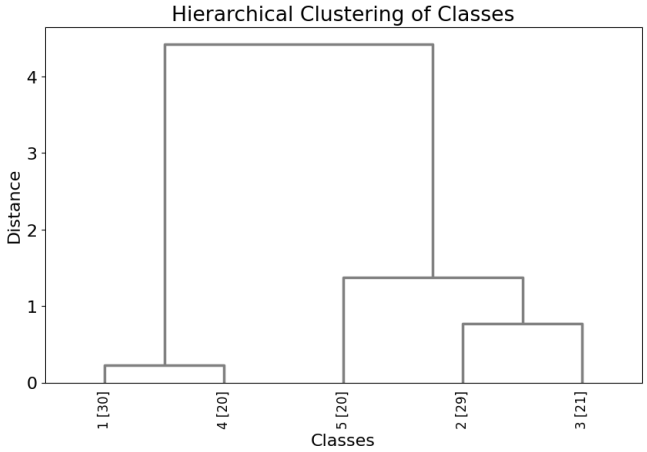


Figure 5.1: Hierarchical Clustering of Classes from AUTOCLASS for Acute Inflammations data.

Table 5.1: Cluster centres from KMCMD for 1 numerical (Temperature) and 5 binary categorical (Lumbar pain - Burning of urethra) attributes of Acute Inflammations data.

Attributes	C1	C2
Temperature	38.836	38.616
Occurrence of nausea	{Yes: 0.492 No: 0.508}	{Yes: 0 No: 1}
Lumbar pain	{Yes: 0.491 No: 0.508}	{Yes: 0.672 No: 0.328}
Urine pushing	{Yes: 0.831 No: 0.169}	{Yes: 0.508 No: 0.492}
Micturition pains	{Yes: 1 No: 0}	{Yes: 0 No: 1}
Burning of urethra	{Yes: 0.491 No: 0.508}	{Yes: 0.344 No: 0.656}

Figure 5.1 shows the hierarchical structure of the 5 classes generated by AUTOCLASS. AUTOCLASS automatically determines 5 is the optimal number of classes, in our case, we merge clusters 1 and 4 into cluster C1, and clusters 2, 3, and 5 into cluster C2. After merging, we can see AUTOCLASS has a poor performance on Acute Inflammation Data (AC = 0.592; ARI = 0.026; Table 5.3).

Next, we perform FMSOM on Acute Inflammations data. Since we have 120 samples, we consider the sizes 8×8, 10×10 and 12×12 as reasonable choices for this dataset. To obtain a comprehensive understanding of parameter selection, we performed a naive search on parameter space. and compare the average behavior of various quality measurements for different values of $\sigma(1)$ and T across 10 random seed attempts to determine the strategy of selecting optimal parameters. Figure 5.2 shows a comparison of the quantization error (QE), CV values, accuracy and ARI for different values of $\sigma(1)$ and T. The

plot indicates that there are some outliers for quantization error with the parameter set $\sigma(1) = 4$ (orange) and $\sigma(1) = 5$ (red). and the parameter set with $\sigma(1) = 2$ (blue) and $\sigma(1) = 3$ (green) exhibits a more robust behavior and a relatively higher CV value, accuracy and ARI. In addition, we can see QE and CV values are negatively correlated, the higher the QE, the lower the CV value. However, our analysis did not reveal any significant pattern of T in the results. Thus, we selected the parameter set $(\sigma(1) = 3; T = 50)$ which minimizes the QE to train FMSOM and conduct further analysis.

Figure 5.3 displays the output of FMSOM after 58 iterations. The U-matrix depicted in Figure (a) exhibits a bunch of darker, connected units that separate the upper left region from the rest of the map. This indicates that the data points in this area are far away from each other in the high-dimensional space. We can therefore classify the data points in the upper left region as belonging to cluster 1, and the remaining points as belonging to cluster 2. To determine the cluster performance, we simply compare the majority voting map based on the class labels of the data points within each unit. The color-coded units in (b) clearly indicate the majority voting results. The orange-colored units indicate inflammation, while the red-colored units suggest the absence of inflammation. The green unit represents as "UNDEFINED", which it is not a best matching unit (BMU) for any observations. We can clearly see the red and orange units are well separated. This verifies the U-matrix successfully extract the relation between data points in high-dimensional space.

Table 5.2 displays the average performance of FMSOM for clustering the Acute Inflammations dataset, as evaluated using the metrics described above. We perform similar parameter selection for map size 8×8 and 12×12. The FMSOM with a map size of 10×10 achieved a very high accuracy of 0.95, while the map size of 8×8 had the highest ARI of 0.97. Furthermore, we noted that increasing the map size led to improved QE and CV performance, but accuracy decreased. These findings suggest that relying on QE and CV value to select the map size of the FMSOM may not be a reliable approach.

Table 5.2: Performance evaluation for FMSOM using 5-fold cross-validation on Acute Inflammations dataset.

Size	Acc		ARI		QE		CV	
	Mean	±	Mean	±	Mean	±	Mean	±
8×8	0.93	0.10	0.97	0.07	0.08	0.01	2.23	0.07
10×10	0.95	0.10	0.90	0.20	0.06	0.04	2.24	0.08
12×12	0.90	0.13	0.85	0.19	0.01	0.01	2.25	0.07

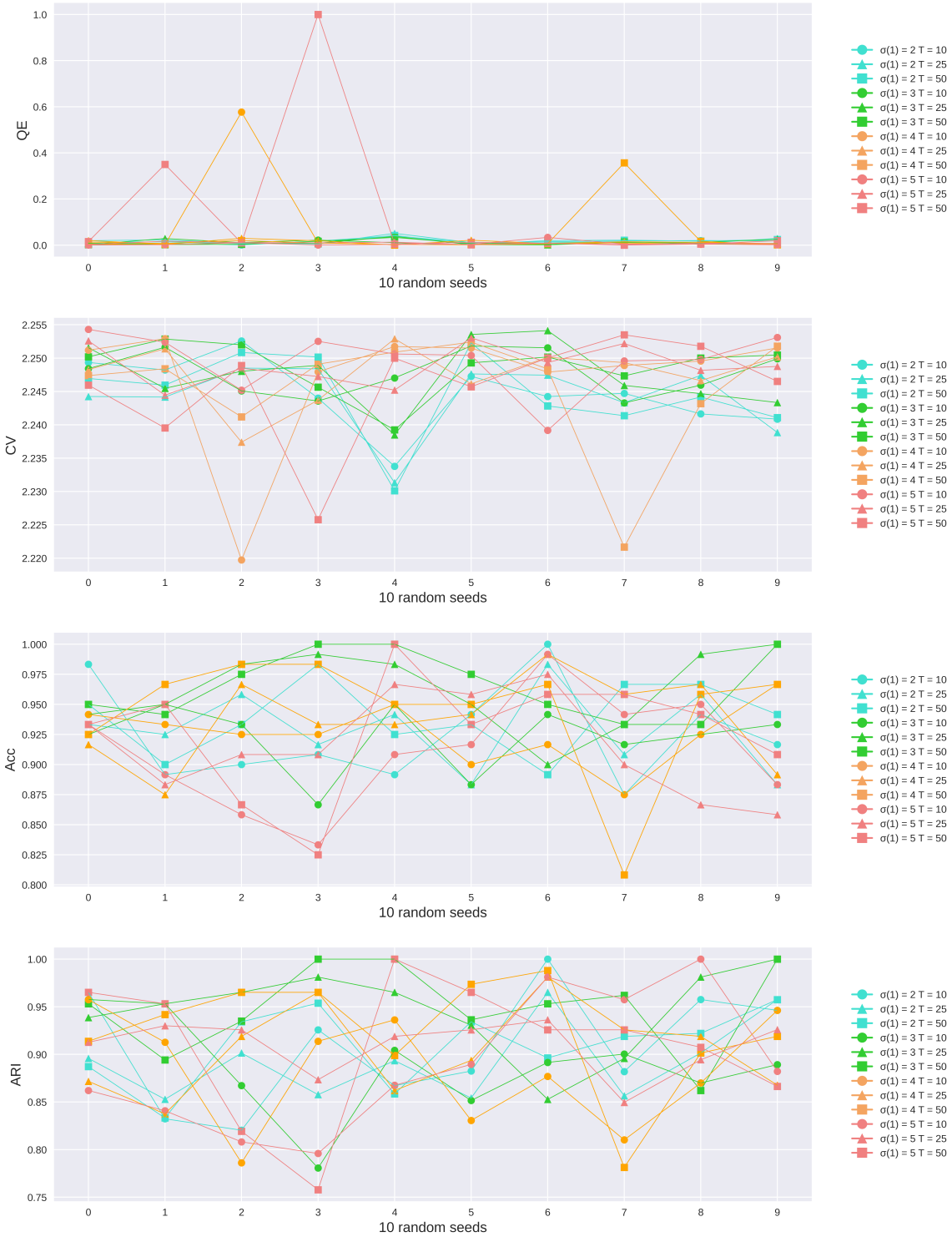
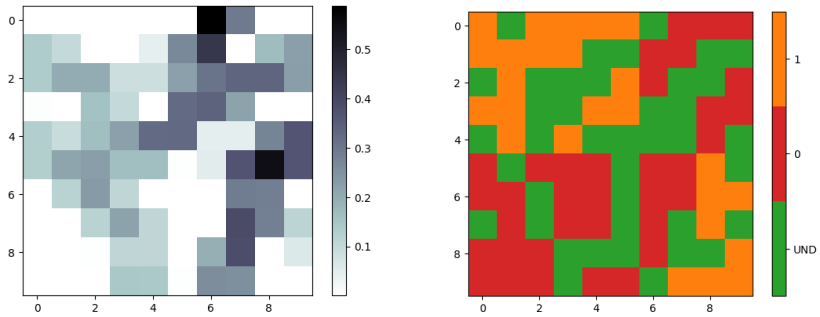


Figure 5.2: Comparison of QE, CV, Acc, ARI value for FMSOM on Acute Inflammation Data with 10 Different Random Seeds for map size 10×10.



(a) U-matrix generated from resulting weight matrix

(b) SOM majority voting result after iteration 58

Figure 5.3: 10 × 10 SOM result for Acute Inflammations data.

We performed a more detailed analysis of the SOM that results for each attributes from the Acute Inflammations Data. Figure 5.4 shows neuron weight vector in all dimensions, corresponding to six different features (one numerical and five categorical). The categorical feature for each neuron neuron was taken to be the feature the neuron had highest probability with, and the numerical feature is just the value of the weight for that neuron.It becomes obvious that certain patterns are consistently present among the majority of patients with inflammation. We can see significant proportion of these patients with inflammation report experiencing micturition pains and urine pushing. This observation is consistent with the findings obtained through KMCMD described above, further validating the accuracy and reliability of the results.

Next, we applied both algorithm 1 and algorithm 2 to cluster the FMSOM for Acute Inflammation Data. The results obtained from both algorithms were quite similar and were deemed satisfactory (AC = 0.83; ARI = 0.44; see Table 5.3).

Table 5.3: Cross-tabulation and Performance evaluation for 4 clustering algorithms on Acute Inflammations dataset.

Class	KMCMD		AUTOCLASS		Algorithm 1		Algorithm 2	
	C1	C2	C1	C2	C1	C2	C1	C2
Yes	49	10	30	20	51	10	51	10
No	10	51	29	41	10	49	10	49
Acc	0.83		0.59		0.83		0.83	
ARI	0.63		0.03		0.44		0.44	

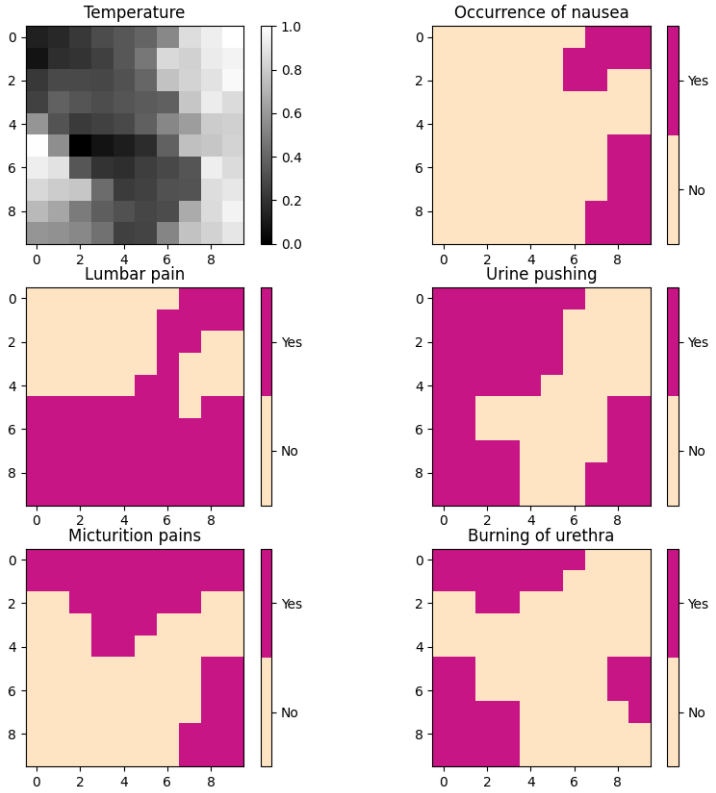


Figure 5.4: SOM components for each attribute in Acute Inflammation Data

5.2 Irish Educational Transitions Data

The dataset collected by Greaney and Kelleghan in 1984 records educational transitions for a sample of 500 schoolchildren in Ireland who were 11 years old in 1967. After removing all records with missing data, the sample size was reduced to 468, with 211 children taking leaving certification and 257 children not taking leaving certification. We selected this dataset because it has a small number of attributes, and the distribution of attribute types is well-balanced, comprising 1 binary feature (Sex), 2 nominal features (Educational level attained and Type of school), and 2 numerical features (Drumcondra Verbal Reasoning Test Score and Prestige score for father's occupation).

The clustering result from KMCMD is good (ACC = 0.80, ARI = 0.32, Table: 5.4). Based on AUTOCLASS result, the optimal number of classes for the dataset was determined to be 8. We further merged clusters 2 and 3 into cluster C1, and clusters 1, 4, 5, 6, 7, and 8 into cluster C2. And AUTOCLASS illustrates a better performance (ACC

= 0.80, ARI = 0.35, Table: 5.4). The performance for algorithm 1 is not bad (ACC = 0.76, ARI = 0.29, Table: 5.4). However, algorithm 2 illustrate a very poor performance (ACC = 0.50, ARI = -0.002, Table: 5.4)

Table 5.4: Cross-tabulation and Performance evaluation for 4 clustering algorithms on Irish Educational Transitions dataset.

Class	KMCMD		AUTOCLASS		Algorithm 1		Algorithm 2	
	C1	C2	C1	C2	C1	C2	C1	C2
Yes	211	0	224	62	156	7	134	123
No	101	156	33	149	101	204	111	100
Acc	0.80		0.80		0.76		0.50	
ARI	0.32		0.35		0.29		-0.002	

Next, we perform analysis using FMSOM. Since the size of datasets is around 500, we think 11×11, 15×15, and 20×20 are reasonable choice of the size of the map. As usual, we use 4 metrics described above with 10-Folds cross validation with each parameter combination. Table 5.5 shows the clustering performance for different parameter sets. Overall, performance of FMSOM is very good. For better visualization purpose, we use the FMSOM with 15×15 neurons result. The U-matrix, majority voting map and heat maps are presented in Figures 5.6 and 5.7. We can clearly see there's a darker boundaries separate the map into 3 regions and validate the clustering result. The group of students who have taken the leaving certification mostly appear on the right side of the map, while those who have not taken it are mostly on the other side. Moreover, we find that the group of students who have taken the leaving certification tend to have higher Prestige scores and are mostly from secondary schools with $l3$, $l7$ and $l10$ education level.

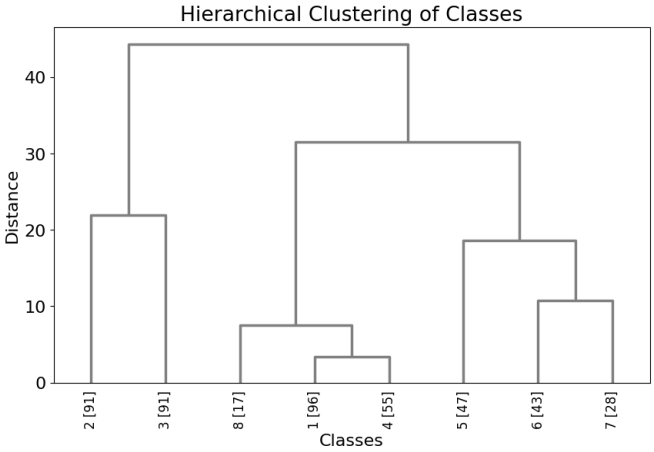
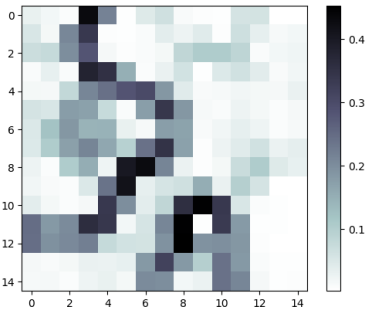


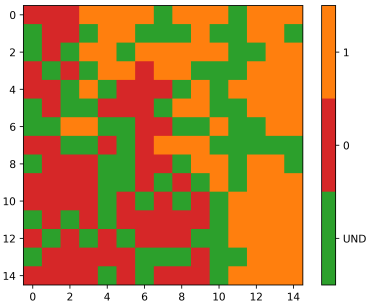
Figure 5.5: Hierarchical Clustering of Classes from AUTOCLASS for Irish Educational Transitions data.

Table 5.5: Performance evaluation for FMSOM using 10-fold cross-validation on Irish Educational Transition dataset.

Map Size	$\sigma(1)$	T	Acc		ARI		QE		CV	
			Mean	±	Mean	±	Mean	±	Mean	±
11×11	3	10	0.98	0.03	0.94	0.09	9.07	0.03	0.78	0.03
		25	0.99	0.02	0.98	0.04	9.43	0.77	0.76	0.04
	4	10	0.98	0.03	0.94	0.09	9.06	1.43	0.80	0.05
		25	0.99	0.02	0.98	0.04	12.48	5.02	0.77	0.02
	5	10	0.98	0.02	0.98	0.04	11.99	3.22	0.76	0.04
		25	0.99	0.02	0.98	0.04	13.08	4.71	0.77	0.03
15×15	5	10	0.95	0.04	0.88	0.09	4.10	0.64	0.88	0,07
		25	0.94	0.03	0.87	0.08	3.72	0.46	0.89	0.04
	6	10	0.97	0.02	0.93	0.06	5.00	4.53	0.90	0.06
		25	0.96	0.03	0.92	0.06	4.32	0.78	0.88	0.07
	7	10	0.97	0.02	0.95	0.04	4.19	0.57	0.87	0.03
		25	0.97	0,03	0.94	0.05	5.30	3.74	0.86	0.04
20×20	6	10	0.87	0.06	0.74	0.10	1.24	0.16	1.08	0.11
		25	0.88	0.06	0.74	0.13	1.54	0.57	1.10	0.10
	8	10	0.86	0.05	0.73	0.08	1.25	0.18	1.08	0.08
		25	0.86	0.06	0.73	0.11	1.81	0.61	1.09	0.09
	10	10	0.84	0.05	0.67	0.10	1.31	0.25	1.13	0.07
		25	0.88	0.09	0.77	0.17	1.39	0.36	0.99	0.12



(a) U-matrix generated from resulting weight matrix



(b) Majority voting map after iteration 21

Figure 5.6: 15 × 15 FMSOM for Irish Educational Transition.

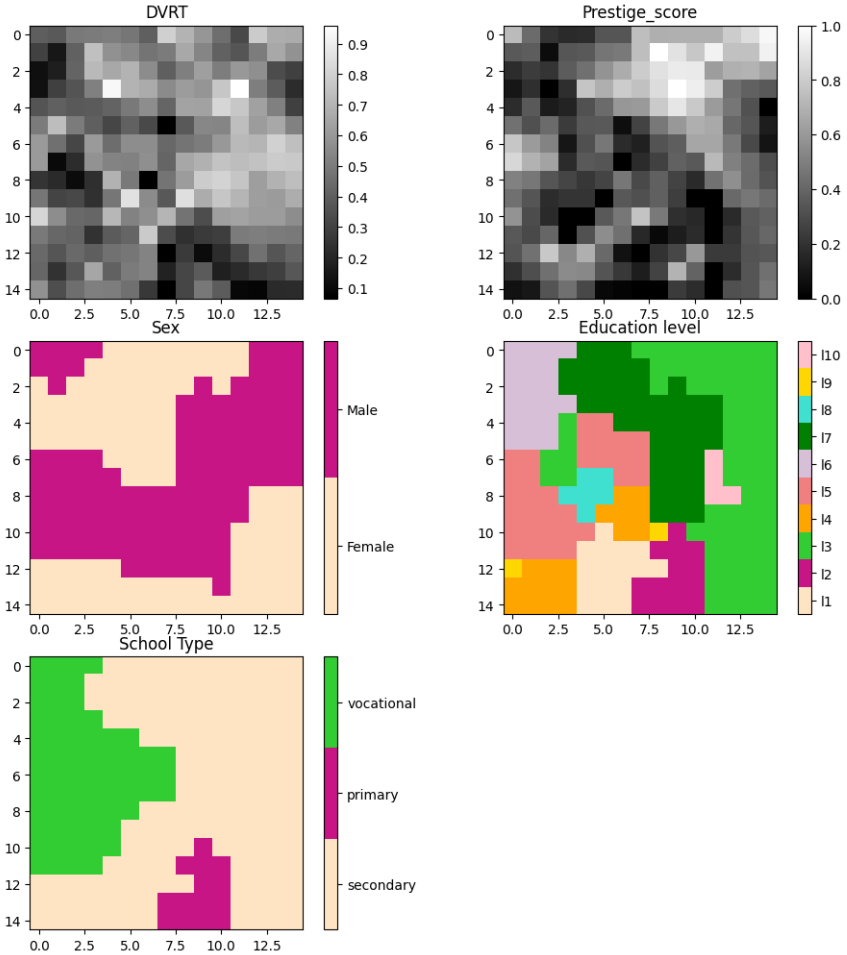


Figure 5.7: SOM components for all attributes in Irish Educational Transitions Data

5.3 Heart Disease Data

This Heat Disease Dataset was compiled in 1988 and comprises four separate databases: Cleveland, Hungary, Switzerland, and Long Beach V. It comprises 76 different attributes, including the predicted attribute, but in all published experiments, only a subset of 14 attributes were utilized. The dataset contains 1025 instances with 9 categorical attributes: Sex, Chest pain type, fasting blood sugar, thal, resting electrocardiographic result, maximum heart rate, exercise-induced angina, the slope, and the number of major vessels, and 4 numerical attributes: Age, Resting blood pressure, serum cholestoral, and ST depression. The "target" field specifically indicates whether a patient has heart disease, with a value of 0 representing the absence of the disease and 1 indicating its presence. We have chosen this dataset due to its relatively large size compared to other datasets we have considered. It is worth checking the effectiveness of the clustering algorithm on larger datasets.

After testing, all 4 algorithms complete clustering under a acceptable time. The result from KMCMD is unexpected good (AC = 0.83; ARI = 0.82; Table 5.6). However, AUTOCLASS has a poor performance (AC = 0.59; ARI = 0.034; Table 5.6). The optimal number of clusters automatically identified by AUTOCLASS is 202 which significantly larger than the true number of classes. As usual, we cut the dendrogram to form 2 clusters. Also, algorithm 1 has a good performance (AC = 0.80; ARI = 0.37; Table 5.6), whereas algorithm 2's performance is moderate (AC = 0.75; ARI = 0.25; Table 5.6). Tables 5.7 present the performance obtained from FMSOM on the Heart Disease Data using 10-fold cross-validation. Overall, the result is satisfactory. We observed that a relatively larger map size 20×20 give the best result. Figure 5.9 shows the U-matrix and majority voting map after iteration 27. In this case, we are not able to find the clear boundaries to find clusters.

Table 5.6: Cross-tabulation and Performance evaluation for 4 clustering algorithms on Heart Disease Data.

Class	KMCMD		AUTOCLASS		Algorithm 1		Algorithm 2	
	C1	C2	C1	C2	C1	C2	C1	C2
Yes	450	76	431	95	408	111	379	137
No	96	403	322	177	91	415	120	389
Acc	0.83		0.59		0.80		0.75	
ARI	0.82		0.03		0.37		0.25	

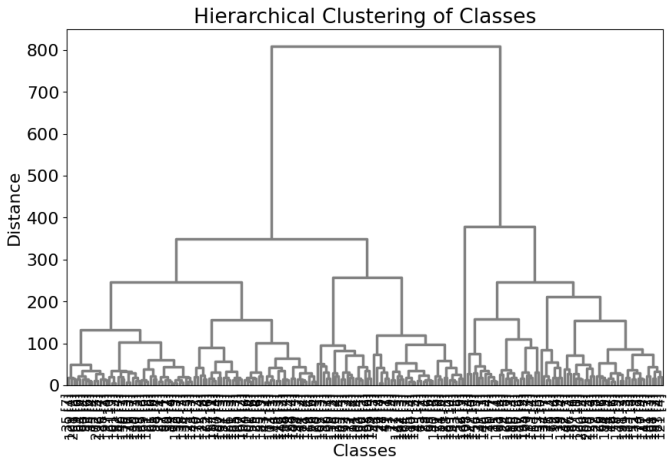
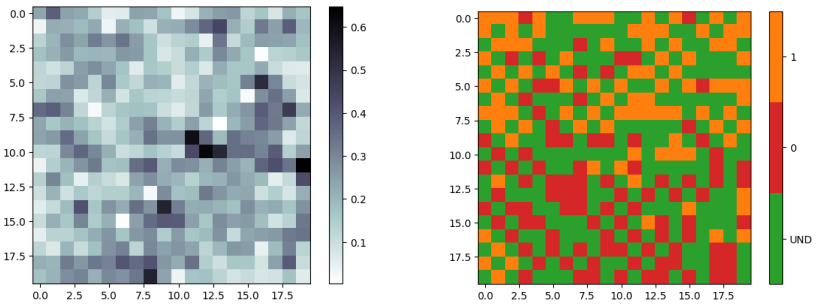


Figure 5.8: Hierarchical Clustering of Classes from AUTOCLASS for Heart Disease data.

Table 5.7: Performance evaluation for FMSOM using 10-fold cross-validation on Heart disease dataset.

Map Size	Acc		ARI		QE		CV	
	Mean	±	Mean	±	Mean	±	Mean	±
10×10	0.85	0.03	0.50	0.09	388.87	27.20	0.55	0.04
15×15	0.87	0.03	0.55	0.10	181.87	17.11	0.64	0.03
20×20	0.88	0.03	0.59	0.07	74.50	7.33	0.84	0.03



(a) U-matrix generated from resulting weight matrix

(b) Majority voting map after iteration 27

Figure 5.9: 20×20 SOM for Heart Disease Data.

Chapter 6

Discussion

6.1 Comparison of Algorithms

Based on the simulation result from the previous chapter. Each clustering algorithm exhibited its unique strengths and limitations. Partition-based algorithm KMCMD unexpectedly illustrates a strong and stable performance among 4 datasets. The result obtain from KMCMD is relatively easy to interpret, k clusters formed can be interpreted as k groups. We can easily obtain some useful information from the cluster center representation. Also, we observe that KMCMD is more effectively for clustering datasets with more catgorical attributes. Hierarchical algorithm SBAC generate a hieracrchical tree-like structure which provide useful information for the user. But it is extermely computational expensive. Unfortunely, after simulating few datasets, we decide to exclude it for current project. The reason is that SBAC need to consider all unique values in numerical attributes to compute the similarity, and finally a distance matrix also need to be constructed which make the complexity very high. The model-based algorithm AUTOCLASS was unstable and did not perform as well as we expected, and the reason behind this is unclear and requires further analysis. We noticed that the optimal number of clusters automatically obtained by AUTOCLASS was typically much higher than the actual number of classes for all six datasets. Merging these clusters into the desired number of clusters by the hierarchical structure may have contributed to the errors. Artificial neuron networks algorithm FMSOM provided a good performance and yielded more insights into the data. FMSOM generated a 2-dimensional visualization of the mixed type data, an we can easily find the relationship of the distribution of data

with each of its features. The weakness of original FMSOM need subjective judge, and the performance without using cross-validation are relatively difficult to compute. Our proposed methods successfully address the aforementioned issue by automatically partitioning the FMSOM into k groups, which allows us to measure clustering performance using standard metrics. Overall, the behavior of our algorithm is good. Despite the use of improved initialization methods, we have found that the poor initialization problem persists, causing our algorithm's performance to sometimes fall short of optimal.

6.2 Summary and Future work

For this project, we have focused on evaluating the performance of three algorithms selected from partition-based, model-based, and artificial neural network clustering algorithms. In future work, we intend to explore more advanced and complex clustering algorithms within each of these categories.

In addition to the proposed FMSOM algorithm, we have developed a new distance measurement based on weighted Jaccard similarity to measure the distance between each neuron. We also proposed two new algorithms that extend k-means and k-medoids with our proposed distance measurement. However, we have identified that our algorithms suffer from poor initialization issues and can become unstable. Therefore, a better initialization method needs to be developed to address these challenges.

Another direction for future work is to build a more robust and comprehensive distance measurement between each neuron. While our proposed distance measurement has shown promising results, a more appropriate distance measurement is always needed for usual clustering algorithms. Investigating other distance measurements such as combining variable importance could be a potential direction for future research.

Moreover, our proposed algorithms are all partition-based methods. In future work, we intend to explore other types of clustering algorithms such as hierarchical clustering techniques to achieve a more robust and comprehensive version of FMSOM. By incorporating hierarchical clustering techniques, we may be able to identify nested structures within the neurons and improve the clustering performance.

Bibliography

Ahmad, Amir, and Lipika Dey. 2007. "A k-mean clustering algorithm for mixed numeric and categorical data." *Data & Knowledge Engineering* 63 (2): 503–527.

Arthur, David, and Sergei Vassilvitskii. 2007. "K-means++ the advantages of careful seeding." In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms,* 1027–1035.

Browne, Ryan P, and Paul D McNicholas. 2012. "Model-based clustering, classification, and discriminant analysis of data with mixed type." *Journal of Statistical Planning and Inference* 142 (11): 2976–2984.

Camadro, Jean-Michel, and Pierre Poulain. 2019. "AutoClassWrapper: a Python wrapper for AutoClass C classification." *Journal of Open Source Software* 4 (39): 1390.

Cheeseman, Peter C, John C Stutz, et al. 1996a. "Bayesian classification (AutoClass): theory and results." *Advances in knowledge discovery and data mining* 180:153–180.

———. 1996b. "Bayesian classification (AutoClass): theory and results." *Advances in knowledge discovery and data mining* 180:153–180.

Corter, James E, and Mark A Gluck. 1992. "Explaining basic categories: Feature predictability and information." *Psychological bulletin* 111 (2): 291.

Del Coso, Carmelo, Diego Fustes, Carlos Dafonte, Francisco J Nóvoa, José M Rodrıguez-Pedreira, and Bernardino Arcay. 2015. "Mixing numerical and categorical data in a Self-Organizing Map by means of frequency neurons." *Applied Soft Computing* 36:246–254.

Honda, Katsuhiro, and Hidetomo Ichihashi. 2005. "Regularized linear fuzzy clustering and probabilistic PCA mixture models." *IEEE Transactions on Fuzzy Systems* 13 (4): 508–516.

Hsu, Chung-Chian, and Yu-Cheng Chen. 2007. "Mining of mixed data with application to catalog marketing." *Expert Systems with Applications* 32 (1): 12–23.

Huang, Zhexue. 1997a. "Clustering large data sets with mixed numeric and categorical values." In *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD),* 21–34. Citeseer.

———. 1997b. "Clustering large data sets with mixed numeric and categorical values." In *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD),* 21–34. Citeseer.

Ji, Jinchao, Wei Pang, Chunguang Zhou, Xiao Han, and Zhe Wang. 2012. "A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data." *Knowledge-Based Systems* 30:129–135.

Kohonen, Teuvo. 1990. "The self-organizing map." *Proceedings of the IEEE* 78 (9): 1464–1480.

Li, Cen, and Gautam Biswas. 2002. "Unsupervised learning with mixed numeric and nominal data." *IEEE Transactions on knowledge and data engineering* 14 (4): 673–690.

Noorbehbahani, Fakhroddin, Sayyed Rasoul Mousavi, and Abdolreza Mirzaei. 2015. "An incremental mixed data clustering method using a new distance measure." *Soft Computing* 19:731–743.

Rand, William M. 1971. "Objective criteria for the evaluation of clustering methods." *Journal of the American Statistical association* 66 (336): 846–850.

Rojas, Ignacio, Gonzalo Joya, and Andreu Catala. 2015. *Advances in Computational Intelligence: 13th International Work-Conference on Artificial Neural Networks, IWANN 2015, Palma de Mallorca, Spain, June 10-12, 2015. Proceedings, Part I.* Vol. 9094. Springer.

Ruzicka, Leopold. 1953. "The isoprene rule and the biogenesis of terpenic compounds." *Experientia* 9 (10): 357–367.

Shen, Furao, and Osamu Hasegawa. 2008. "A fast nearest neighbor classifier based on self-organizing incremental neural network." *Neural networks* 21 (10): 1537–1547.

Tai, Wei-Shen, and Chung-Chian Hsu. 2012. "Growing self-organizing map with cross insert for mixed-type data clustering." *Applied Soft Computing* 12 (9): 2856–2866.

Tarekegn, Adane Nega, Krzysztof Michalak, and Mario Giacobini. 2020. "Cross-validation approach to evaluate clustering algorithms: An experimental study using multi-label datasets." *SN Computer Science* 1:1–9.

Ultsch, Alfred. 2003. "U*-matrix: a tool to visualize clusters in high dimensional data."

Vesanto, J., and E. Alhoniemi. 2000. "Clustering of the self-organizing map." *IEEE Transactions on Neural Networks* 11 (3): 586–600. https://doi.org/10.1109/72.846731.