Embedded Feature Selection for Model-based Clustering

by

Yinlin Fu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2020 by the
Graduate Supervisory Committee:

Teresa Wu, Chair
Pitu Mirchandani
Jing Li
Giulia Pedrielli

ARIZONA STATE UNIVERSITY

May 2020

ABSTRACT

Model-based clustering is a sub-field of statistical modeling and machine learning. The mixture models use the probability to describe the degree of the data point belonging to the cluster, and the probability is updated iteratively during the clustering. While mixture models have demonstrated the superior performance in handling noisy data in many fields, there exist some challenges for high dimensional dataset. It is noted that among a large number of features, some may not indeed contribute to delineate the cluster profiles. The inclusion of these "noisy" features will confuse the model to identify the real structure of the clusters and cost more computational time. Recognizing the issue, in this dissertation, I propose a new feature selection algorithm for continuous dataset first and then extend to mixed datatype. Finally, I conduct uncertainty quantification for the feature selection results as the third topic.

The first topic is an embedded feature selection algorithm termed Expectation-Selection-Maximization (ESM) model that can automatically select features while optimizing the parameters for Gaussian Mixture Model. I introduce a relevancy index (RI) revealing the contribution of the feature in the clustering process to assist feature selection. I demonstrate the efficacy of the ESM by studying two synthetic datasets, four benchmark datasets, and an Alzheimer's Disease dataset.

The second topic focuses on extending the application of ESM algorithm to handle mixed datatypes. The Gaussian mixture model is generalized to Generalized Model of

Mixture (GMoM), which can not only handle continuous features, but also binary and nominal features.

The last topic is about Uncertainty Quantification (UQ) of the feature selection. A new algorithm termed ESOM is proposed, which takes the variance information into consideration while conducting feature selection. Also, a set of outliers are generated in the feature selection process to infer the uncertainty in the input data. Finally, the selected features and detected outlier instances are evaluated by visualization comparison.

This one's dedicated to my family,

who've blessed me in more ways than I can tell.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor, Teresa Wu, for her support, encouragement and patience. Without her support and nurturing, it would have not been possible to complete my dissertation.

I would like to extend my sincere thanks to members of my committee for their insightful comments, valuable interactions, and advice: Pitu Mirchandani, Jing Li, and Giulia Pedrielli. I am also grateful to Jing Li and Daniel McCarville who shared their tremendous experience with me while I was teaching with them.

I am also grateful to my friends and lab mates who support me through this wonderful experience: Xiaonan Liu, Lujia Wang, Hyunsoon Yoon, Congzhe Su, Fei Gao, Nathan Gaw and Yanzhe Xu.

Finally, I would like to thank my family for supporting me during the compilation of this dissertation.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background

Clustering is an unsupervised data mining technique to group the data into different segments by discovering the patterns within the dataset [1]. Intuitively, data points from the same group are more like each other than the data from other groups. An example of clustering is depicted in Figure 1. The input patterns are shown in Figure 1(a) and the desired clusters are shown in Figure 1(b). From Figure 1(b), points belonging to the same clusters are given the same shape. Due to the nature of the problem, clustering has been broadly applied to many fields including manufacturing [2], biology [3], finance [4] and astronomy [5], just to name a few.

Different approaches to clustering are proposed and can be described by Figure 2 from the clustering survey paper [1]. Generally, clustering algorithms are categorized as hierarchical and partitional methods.

The hierarchical clustering starts by assigning each data point as its own cluster and obtaining the proximity distance matrix (e.g., single, complete link, group average) between each pair of data points (clusters). The algorithm then finds the closest pair of clusters to merge into a single cluster, and the new distances between the clusters are reassessed accordingly. Iteratively, the process ends until all data points are formed into one cluster as the root of the cluster tree. The cluster tree, a.k.a. dendrogram is generated illustrating the cluster structures at different levels. Other than simplicity and intuitive visual presentation of the data, one known advantage of hierarchical clustering is its flexibility in deriving the clusters. That is, the user can visualize or use domain knowledge to decide the number of clusters as well as the level of the cluster structures to be investigated. However, as pointed out by [6], hierarchical clustering suffers from quadratic time complexity limiting its application to large dataset. In addition, for the data with noise, the performance of hierarchical clustering is unsatisfactory.

In traditional partitional methods such as k-means and its extensions k-prototype, the number of clusters usually needs to be pre-specified. Centroid, the center of each cluster is used to guide the assignment of the data point to the cluster. Centroid is iteratively updated during the clustering process. Due to its computational effectiveness, these partitional methods have great advantage over hierarchical algorithms for large scale datasets [1], [7]. However, it is a non-trivial task to determine the number of clusters. Often, it requires either domain knowledge as prior or empirical experiments to identify the appropriate number of clusters. In addition, k-means methods are known as “hard assignment” in which the data point is assigned to one specific group with certainty. However, during the clustering process, it is very likely that a data point is assigned to one cluster initially and is reverted in the subsequent steps. The robustness of the model, especially to a noisy dataset, is thus questionable.

Lately, mixture based partitional methods have attracted more attention. Example are Gaussian mixture models [8], Dirichlet process mixture models [9] and Latent Class model

[10]. These mixture models are built upon a well-studied statistical inference framework that provides guidelines for determining the optimal number of clusters. Also, the mixture model can generate statistical metrics from the data distribution instead of the simple distance between data points. The model uses the probability to describe the degree of the data point belonging to the cluster and the probability is updated iteratively during the clustering. Since most real-world problems are uncertain by nature, the use of this "soft assignment" approach may be a better alternative comparing to the "hard assignment" (e.g., k-means). Indeed, the Gaussian Mixture Model (GMM) is a generalized approach, and k-means is a particular case of GMM [11].

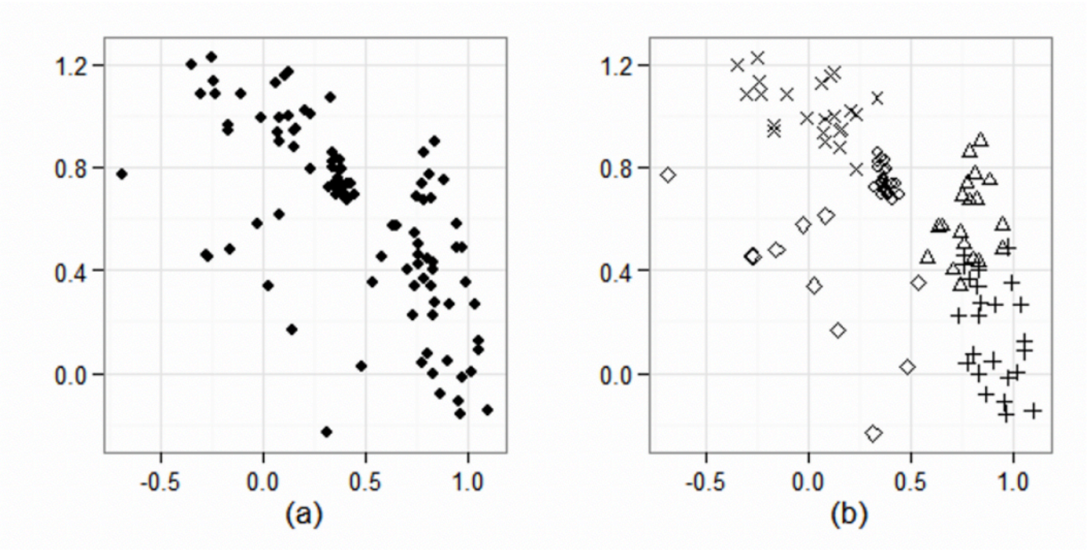


Figure 1. Clustering Example.

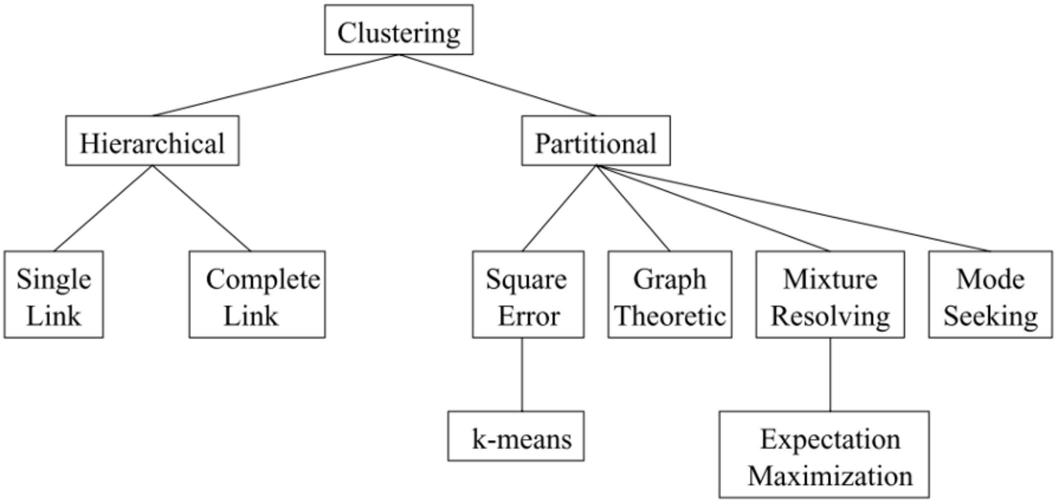


Figure 2. A Taxonomy of Clustering Approaches (Adopted From [1]).

1.2 Challenges and Research Scope

While mixture model-based clustering has demonstrated its superior performance in handling noisy data, there exists some challenges in employing mixture model to cluster high dimensional dataset [12].

The challenges of applying model-based clustering on high dimensional dataset are as follows:

- **Redundant Features**: Among the large number of features, some may not truly contribute to delineate the cluster profiles. Inclusion of these “noisy features” requires more parameter estimations for the mixture model, which are computationally costly. Additionally, the noisy features will confuse the model to identify the true structure of the clusters [13].

- **Mixed Data Type**: Real world applications usually contain mixed data type instead of a single type (e.g., continuous or categorical). However, most existing clustering algorithms including Gaussian mixture model can only handle single data type.
- **Lack of uncertainty quantification**: For machine learning, it is very important to assess the credibility of the models. However, for clustering and feature selection, it is very hard to evaluate the performance since the true labels are unknown. Also, the data collected are inherently uncertain due to noise, incompleteness, and inconsistency. Thus, rigorous quantification of uncertainty in the underlying data, the model, and the resulting predictions are critical and still a big challenge.

1.3 Expected Original Contribution

The objective of the research is to develop new embedded feature selection methods for model-based clustering that overcome the aforementioned challenges and demonstrate their utility in the health care application of Alzheimer's disease. The expected original contributions include:

- **Development of an Expectation-Selection-Maximization (ESM) algorithm**: I propose a new embedded feature selection algorithm for Gaussian Mixture Model. The embedded feature selection algorithm can simultaneously select features while estimating models. The proposed algorithm naturally embeds a feature selection step (S) in between the E step and M step, termed ESM. Specifically, we introduce a relevancy index of the feature based on the EM responsibility, a metric indicating the probability of assigning a data point to a certain clustering group. The relevancy index of a feature is defined as the differences between the responsibility measures for the feature sets including and excluding the feature. This index reveals the contribution of the feature in the clustering process thus can assist the feature selection. To demonstrate the efficacy of the proposed ESM algorithm, we develop two synthetic datasets. One synthetic data has 10 independent features with two of them are relevant. Another synthetic data includes 15 features with two of them are relevant. Among the 15 features some are correlated with each other aiming to mimic the correlation effect in real data sets. In both experiments, the ESM algorithm can select relevant features in 100% accuracy for data size not less than 300. In addition, to demonstrate the applicability of our proposed algorithm, a set of benchmark dataset and a real medical application of Alzheimer Disease data are studied. The results show that our proposed algorithm gains better performance than original GMM without feature selection in terms of correctly identifying cluster groups. The details of this work are illustrated in chapter 2.

- **Development of the Generalized Model of Mixtures (GMoM)**: I extend the Gaussian mixture model to the Generalized Model of Mixtures (GMoM) to handle mixed data type and then develop feature selection algorithm over GMoM. The basic assumption of GMoM is that data is generated from mixtures of distributions. Each mixture represents a cluster and can be expressed by a joint distribution of multiple types of features including continuous features (multivariate normal), binary features (Bernoulli) and categorical features (Multinomial). Under the framework of GMoM, I propose a feature selection algorithm by introducing the term Feature Index (FI), which is based on the posterior probability of assigning data points to cluster group called responsibility as before. The basic assumption behind the feature selection algorithm is that given a feature, if the inclusion and the exclusion of the feature show no significant differences in responsibility measure, then the feature is not truly contributing to the clustering. Similar to ESM, the assessment of the FI is embedded in between the E and M steps in the EM implementation for feature selection. To demonstrate the efficacy of the feature selection algorithm on mixed type data, one synthetic dataset, one benchmark and one real application dataset are studies. For the synthetic dataset with known relevant features, the algorithm can select relevant features in 100% accuracy. For the benchmark dataset, the results show that the proposed algorithm gains better performance than the other four exiting algorithms in literature. Finally, the results on Alzheimer's disease dataset show that the proposed algorithm can make full use of the data information (both categorical and continuous data types) and gain better performance than clustering algorithms without feature selection. The details of this work are illustrated in chapter 3.

- **Development of Uncertainty Quantification (UQ) for unsupervised feature selection results.** I propose a new algorithm termed ESOM based on the original ESM algorithm to evaluate the uncertainty in the input data and feature selection step. Specifically, the distribution of the delta values (difference between responsibilities before and after excluding certain feature) over all data points is generated and taken into consideration when doing feature selection. The confidence interval of the delta values is used to quantify the confidence of selecting a certain feature. In addition, for data points on the tail of the distribution are detected as candidate outliers, which are further quantified and visualized to represent the uncertainty of input dataset**.** To evaluate the performance of the ESOM algorithm, I conducted experiments on four benchmark datasets. The experiments show that the selected features form a new space that's easier to cluster compared with the original feature space. Also, the ESOM algorithm improves the clustering accuracy compared with the original ESM algorithm. The details of this work are illustrated in chapter 4.

1.4 Dissertation Organization

My dissertation research will be presented in three chapters, as shown in Figure 3. Chapter 2 presents the development of topic (I): embedded feature selection for Gaussian Mixture Model. Chapter 3 presents the development of topic (II): feature selection for Generalized Model of Mixture to handle mixed type dataset. Chapter 4 presents the development of topic (III): Uncertainty Quantification of the proposed embedded feature selection. Chapter 5 summarizes the dissertation with conclusion remarks and discussions on future work.

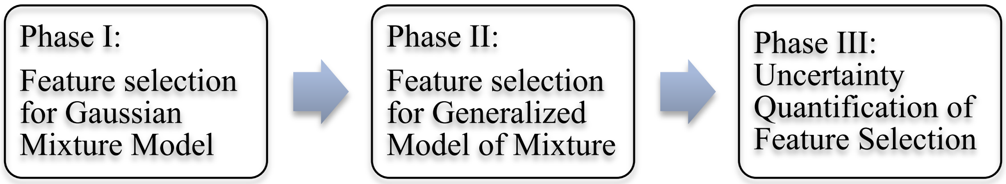

Phase I:
Feature selection for Gaussian Mixture Model
Phase II:
Feature selection for Generalized Model of Mixture
Phase III:
Uncertainty Quantification of Feature Selection

Figure 3. Dissertation Framework.

CHAPTER 2

FEATURE SELECTION FOR GAUSSIAN MIXTURE MODEL

2.1 Introduction

Lately, Gaussian Mixture Model (GMM) has demonstrated its superior performance in handling noisy data in the field of image classification and segmentation [14], automatic speaker recognition [15]. However, there still exists some challenges in GMM research for high dimensional dataset [12]. It is noted that among the large number of features, some may not truly contribute to delineate the cluster profiles. Inclusion of these "noisy features" requires more parameter estimations for GMM, which are computationally costly. Additionally, the noisy features will confuse the model to identify the true structure of the clusters [13].

Recognizing this issue, researchers consider feature selections for solution. The current feature selection methods designed specific for GMM are divided into three groups: filters, wrapper, and embedded as defined for general feature selection approaches [16].

Filters address the problem of feature selection and model building independently and treat feature selection as a pre-processing step. For example, Krishnam et al. [17] propose a feature selection method for GMM based on the Fisher ratio. The Fisher ratio between two classes is defined as the mean differences square over the mean of variances. Then the method ranks features by Fisher ratio assuming that discriminating features have a higher Fisher ratio. As the Fisher ratio method, typical filters select features with no regard to the model building process. They are generally fast but can select feature subsets with low predictive accuracy.

Wrappers usually build models on the subset of the features and evaluate the model performance for the subset of features. The wrappers require iterations between searching from subset space and constructing models based on subsets. Under the umbrella of wrappers, one typical feature selection approach for GMM is to formulate the feature selection problem as model comparison problem. For example, Raftery and Dean [18] propose to use Bayesian information criteria (BIC) to compare models. In detail, the features are divided into three sets: relevant set, irrelevant set and the set of features proposed for inclusion or exclusion from relevant set. The univariate analysis is launched to identify the first most important feature for the clustering as the initial relevant feature set. Remaining features, one by one, is evaluated via Bayes factor, the likelihood of adding vs. excluding the feature from the relevant feature set. The features excluded from the clustering form the irrelevant feature set. A greedy stepwise forward selection approach is applied. In calculating the Bayes factor, Raftery and Dean [18] assume the irrelevant features are dependent on the features from the relevant feature set. Maugis et al. [13] argue such assumption may not hold. Therefore, Maugis et al. [13] propose a backward stepwise strategy starting with all features so the model takes block interactions between features into account. To optimize the search process, Scrucca [19] proposes to use genetic algorithm, which is a stochastic search algorithm inspired by evolutionary biology and natural selection. In all three approaches: forward selection (Raftery & Dean [18]), backward selection (Maugis et al. [13]) and genetic algorithm (Scrucca [19]), the comparison between two models: the model with and without the feature, is conducted to make the feature selection. Since the parameter estimation process for GMM is iterative, which in itself is computational expensive, the wrapper technique that needs iterations between selecting feature subset and model estimation can quickly become unfeasible. Besides the expensive computational cost, another drawback of wrappers is the difficulty of handling unsupervised data. Since the label is unknown for unsupervised data, the evaluation metric such as "accuracy" used in supervised data is infeasible and thus an unsupervised evaluation metric is needed. The proposed metrics such as Bayesian Information Criteria (BIC) often tradeoff between the likelihood and the number of features included in the model can be inaccurate and uninformative for feature selection [16].

Compared with filters that gives low accuracy and the wrappers that require high computational cost, embedded method that can simultaneously select features while constructing models is leading the trend [16]. Numerous embedded feature selection approaches for GMM have been developed. In general, they belong to three types. The first type is penalized model-based clustering, which introduces a penalty term in the forms of log-likelihood function to regularize parameter estimation in the EM [20]–[22]. The second type is feature saliency-based approach. Law et al. [23] introduce a latent variable for each feature indicating whether the feature is relevant or not and the probability of the feature being relevant is defined as feature saliency. The posterior probability estimates of the latent variables are updated in the EM procedure. One advantage of this approach is that the estimated feature saliency can be used as feature rank as provided by filters approach. The third type is Bayesian feature weighting methods, which treats mean and covariance as random variables [24]. To obtain analytical solutions, all three approaches require the diagonal covariance matrix for the EM implementation. In other words, they are under the strict assumption that the features are independent with each other.

In this research, we propose a new embedded feature selection approach for GMM, which considers the inter–dependences of the features and gives feature relevance rank automatically. The general idea is as follows. Delving into the detailed EM implementation, one interesting observation is responsibility metric calculated as an intermediate step. Responsibility is to measure the probability assigning the data point to a specific cluster. Given a feature, if the inclusion and the exclusion of the feature show no significant differences in responsibility measure, we conclude this feature is not truly contributing to the clustering. We define this responsibility difference as *RI*, the relevancy index for the feature. Based on *RI*, we propose ESM algorithm, the assessment of the *RI* is embedded naturally in between the E and M steps in the EM implementation for feature selection. One advantage of our proposed ESM is that it follows GMM principle on data dependencies, it is generalized thus can handle the dataset with dependent and independent variables. The second advantage is that the proposed ESM is embedded in the EM procedure which guarantees the convergence. The third advantage is that the relevancy index gives the feature relevancy rank automatically as provided in filters method.

The remainder of this study is organized as follows. Section 2.2 reviews the basics of GMM and EM algorithm. Section 2.3 presents the proposed ESM algorithm in detail with theoretical analysis on the *RI*. Next, three sets of experiments on two synthetic datasets and one Alzheimer's Disease dataset are illustrated in Section 2.4. In Section 2.5, the conclusion and future direction are presented.

2.2 Review of GMM and EM

A finite Gaussian mixture model is the weighted sum of K Gaussian components (clusters) and can be written as

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2.1}$$

where $\boldsymbol{\mu}_k$ is the mean vector of k^{th} component, $\boldsymbol{\Sigma}_k$ is the covariance matrix of k^{th} component, π_k is the mixing coefficient representing the proportion of k^{th} component, and $N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the probability distribution of k^{th} component shown in equation (2.2).

$$N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} exp\,\{-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)\} \tag{2.2}$$

As in [25], we use a *K*-dimensional binary random variable $\boldsymbol{z}$ having a one of *K* representation in which an element z_k is equal to 1 and all other elements are equal to 0. The marginal distribution over $\boldsymbol{z}$ is specified in terms of the mixing coefficient π_k,

$$p(z_k = 1) = \pi_k \tag{2.3}$$

where $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$. Suppose we have a data matrix $X \in R^{N \times D}$ with N data points and D features in which the n^{th} row is $\boldsymbol{x}_n^T$. If the data points are drawn independently from the distribution, the log likelihood function is given by

$$ln\, P(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} ln \left\{ \sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{2.4}$$

In a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters including means μ_k, covariances Σ_k and mixing coefficients π_k, *k*=1, 2…, *K*. EM is a commonly used four-step algorithm to estimate these parameters. The algorithm starts from initializing μ_k, Σ_k and π_k, and evaluates the initial value of log likelihood function. In the second step (known as E step), the EM evaluates the responsibilities under the current parameter settings. The responsibility is defined as the probability of assigning a data point to a specific clustering group:

$$\gamma(z_{nk}) = p(z_k = 1|\boldsymbol{x}_n) = \frac{p(z_k = 1)p(\boldsymbol{x}_n|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x}_n|z_j = 1)} = \frac{\pi_k N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j N(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{2.5}$$

In the third step, the EM re-estimates the parameters given the responsibilities. The estimation method is maximum likelihood, and therefore the third step is called “M step”. The final step is to check the convergence of log-likelihood. If the log likelihood difference between two iterations is small enough (e.g., less than a small number), it is converged. Otherwise, the algorithm goes back to the E step initiating the next iteration. For details of the EM algorithm, interested readers are to refer to [25]. We present our proposed ESM algorithm based on responsibility metric in the next section.

2.3 Proposed Method: ESM

2.3.1 Relevancy Index and ESM

We The proposed ESM algorithm takes advantage of the responsibility measures in the E step. Let us consider the responsibilities $\gamma(z_{nk})$, the probability of assigning the data point x_n to cluster k, if we remove one specific feature, responsibilities shall change. Specifically, let the full feature space with *D* features be $F = \{f_1, f_2, \dots, f_D\}$, the feature space excluding feature j be $F_j^- = \{f_1, f_2, \dots, f_D\} \backslash \{f_j\}$. Here, we denote the responsibility on the full feature space as $\gamma^F(z_{nk})$ and the responsibility on the reduced feature space (excluding feature j) as $\gamma^{F_j^-}(z_{nk})$ which is related to the n^{th} data point and the k^{th} cluster. Relevancy index (*RI*) is defined as the difference between two responsibilities averaged over *N* data points and *K* clustering groups to evaluate the importance of j^{th} feature to the clustering. It is written as:

$$RI(j) = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} |\gamma^F(z_{nk}) - \gamma^{F_j^-}(z_{nk})| \qquad (2.6)$$

The assumption behind our proposed method is that if $RI(j)$ is smaller than a pre-defined threshold, feature j is neglectable in assigning data points to clusters. Thus, feature j can be removed in feature selection process under the condition the $RI(j)$ converges over the iterations. Concerning the convergence criteria for RI, we evaluate the changes of $RI(j)$ between the current and previous iteration, let say, if it is less than a small number, e.g., 0.0005, we conclude $RI(j)$ converges. Regarding the pre-defined threshold for feature selection, it can be set based on the approximate number of features to be selected (see experiments in Section 4 for details). Table 1 summarizes the ESM algorithm with the proposed S step highlighted.

Table 1. ESM Algorithm Pseudo Code.

```
1. Initialize the means $\mu_k$, covariances $\Sigma_k$ and mixing coefficients $\pi_k$, and evaluate the initial value of the log likelihood.
2. E step. Evaluate the responsibilities using the current parameter values
       $\gamma^F(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k)}$
   and responsibilities after excluding each feature
       $\gamma^{F_j^-}(z_{nk}) = \frac{\pi_k N(x_n^*|\mu_k^*, \Sigma_k^*)}{\sum_{k=1}^{K} \pi_k N(x_n^*|\mu_k^*, \Sigma_k^*)} \ for\ j = 1,2,..D$
   where $x_n^*$, $\mu_k^*$ and $\Sigma_k^*$ are the corresponding vector of $x_n$, $\mu_k$ and $\Sigma_k$ after excluding $j_{th}$ variable.
3. S step. Calculate the difference between responsibilities before and after excluding $j_{th}$ feature at iteration t.
       $RI(j)^{(t)} = \frac{1}{NK} \sum_{n,k} |\gamma^F(z_{nk}) - \gamma^{F_j^-}(z_{nk})|$
   If $\left|RI(j)^{(t+1)} - RI(j)^{(t)}\right| < \epsilon$ (converged) and $RI(j)^{(t)}$ is small enough, then discard the feature with smallest $RI$ and update the full feature space $F$.
4. M step. For reduced data with feature space $F$, re-estimate the parameters using the current responsibilities
       $\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$
       $\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})'$
       $\pi_k^{new} = \frac{N_k}{N}$
5. Evaluate the log likelihood
       $ln\ P(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} ln \{\sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k)\}$
If the parameters or the log likelihood are not converged, go back to step 2.
```

2.3.2 Theoretical Analysis on Relevancy Index

In As stated in [26], for a specific feature, when the variances are the same among clusters, if the mean of a cluster on the feature is equal to the global mean, this feature is uninformative or irrelevant to clustering. Intuitively, the bigger difference between the means, the more relevant the feature is. The following theorem gives the relationship between *RI* and difference of means which justifies the use of *RI* can identify the irrelevant features to be removed.

Theorem 1. Given a dataset with *D* features and *K* clusters, let the conditional mean of cluster l on the j^{th} feature given all the other features be μ_{lj}, the conditional mean difference between two clusters l and m be $\epsilon_j(l,m) = |\mu_{lj} - \mu_{mj}|$, the lower bound of $RI(j)$ is an increasing function of $\epsilon_j(l,m)$ for some $l, m \in \{1,2,\dots,K\}$. Specifically, if $\epsilon_j(l,m) = 0$, $RI(j) = 0$.

Proof:

Let x_n be a data point with *D* features and the j^{th} variable is denoted as x_{nj}. Let x_n^* be the corresponding vector of x_n after excluding the j^{th} feature. Let μ_k, Σ_k be the mean and covariance of cluster k on full *D* feature dataset and μ_k^* and Σ_k^* be the corresponding vector of μ_k and Σ_k after excluding j^{th} variable.

The joint distribution of *D* features can be decomposed into the joint distribution of all the other features except j^{th} feature and the conditional distribution of j^{th} feature. That is,

$$\pi_k N(x_n; \mu_k, \Sigma_k) = \pi_k N(x_n^*; \mu_k^*, \Sigma_k^*) N\left(x_{nj} \middle| x_n^*; \mu_{kj}(x_n^*), \sigma_{kj}^2(x_n^*)\right) \qquad (2.7)$$

where $\mu_{kj}(x_n^*)$ is a linear function of x_n^*. Hence, we can rewrite $RI(j)$ as

$$RI(j) = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}|\gamma^{F}(z_{nk}) - \gamma^{F_j^-}(z_{nk})|$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\left|\frac{\pi_k N(x_n;\mu_k,\Sigma_k)}{\sum_{l=1}^{K}\pi_l N(x_n;\mu_l,\Sigma_l)} - \frac{\pi_k N(x_n^*;\mu_k^*,\Sigma_k^*)}{\sum_{l=1}^{K}\pi_l N(x_n^*;\mu_l^*,\Sigma_l^*)}\right|$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\left|\frac{\pi_k N(x_n^*;\mu_k^*,\Sigma_k^*) N\left(x_{nj}\middle|x_n^*;\mu_{kj}(x_n^*),\sigma_{kj}^2(x_n^*)\right)}{\sum_{l=1}^{K}\pi_l N(x_n^*;\mu_l^*,\Sigma_l^*) N\left(x_{nj}\middle|x_n^*;\mu_{lj}(x_n^*),\sigma_{lj}^2(x_n^*)\right)} - \frac{\pi_k N(x_n^*;\mu_k^*,\Sigma_k^*)}{\sum_{l=1}^{K}\pi_l N(x_n^*;\mu_l^*,\Sigma_l^*)}\right|$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\left|\frac{\pi_k N(x_n^*;\mu_k^*,\Sigma_k^*)}{\sum_{l=1}^{K}\pi_l N(x_n^*;\mu_l^*,\Sigma_l^*)\dfrac{N\left(x_{nj}\middle|x_n^*;\mu_{lj}(x_n^*),\sigma_{lj}^2(x_n^*)\right)}{N\left(x_{nj}\middle|x_n^*;\mu_{kj}(x_n^*),\sigma_{kj}^2(x_n^*)\right)}} - \frac{\pi_k N(x_n^*;\mu_k^*,\Sigma_k^*)}{\sum_{l=1}^{K}\pi_l N(x_n^*;\mu_l^*,\Sigma_l^*)}\right| \tag{2.8}$$

Given any data point x_n, where $n \in \{1,2 \dots, N\}$, there exists a component m_n, which has the largest likelihood:

$$m_n = \underset{l\in\{1,2..,K\}}{\operatorname{argmax}} N\left(x_{nj}\middle|x_n^*;\ \mu_{lj}(x_n^*),\sigma_{lj}^2(x_n^*)\right) \tag{2.9}$$

Let the ratio of likelihood between any two components, component *l* and component *k* be

$$Ratio(l,k;x_n) = \frac{N\left(x_{nj}\middle|x_n^*;\mu_{lj}(x_n^*),\sigma_{lj}^2(x_n^*)\right)}{N\left(x_{nj}\middle|x_n^*;\mu_{kj}(x_n^*),\sigma_{kj}^2(x_n^*)\right)} \tag{2.10}$$

where $l,k\ \in \{1,2,\dots,K\}$. By equation (2.8), we have

$$Ratio(l, m_n; x_n) = \frac{N\left(x_{nj} \middle| x_n^*; \mu_{lj}(x_n^*), \sigma_{lj}^2(x_n^*)\right)}{N\left(x_{nj} \middle| x_n^*; \mu_{m_nj}(x_n^*), \sigma_{m_nj}^2(x_n^*)\right)} \leq 1$$

Expanding the Ratio in equation (2.9), we obtain

$$Ratio(l, m_n; x_n) = \frac{\exp\{-\frac{1}{2\sigma_{lj}^2}\left(x_j - \mu_{lj}(x^*)\right\}}{\exp\{-\frac{1}{2\sigma_{m_nj}^2}\left(x_j - \mu_{m_nj}(x^*)\right\}}$$

$$= \exp\left\{-\frac{1}{2\sigma_j^2}\left(2x_j - \mu_{lj}(x^*) - \mu_{m_nj}(x^*)\right)\left(\mu_{m_nj}(x^*) - \mu_{lj}(x^*)\right)\right\} \qquad (2.11)$$

Since $Ratio(l, m_n; x_n) \leq 1$, we have $\left(2x_j - \mu_{lj}(x^*) - \mu_{m_nj}(x^*)\right)\left(\mu_{m_nj}(x^*) - \mu_{lj}(x^*)\right) \geq 0$, which can be rewritten as $\left|2x_j - \mu_{lj}(x^*) - \mu_{m_nj}(x^*)\right|\left|\mu_{m_nj}(x^*) - \mu_{lj}(x^*)\right| = \left|2x_j - \mu_{lj}(x^*) - \mu_{m_nj}(x^*)\right|\epsilon_j(m_n, l)$.

Therefore,

$$RI(j) \geq \frac{1}{NK}\sum_{n=1}^{N}\left|\frac{\pi_{m_n}N\left(x_n^*; \mu_{m_n}^*, \Sigma_{m_n}^*\right)}{\sum_{l=1}^{K}\pi_l N(x_n^*; \mu_l^*, \Sigma_l^*)\frac{N\left(x_{nj} \middle| x_n^*; \mu_{lj}(x_n^*), \sigma_{lj}^2(x_n^*)\right)}{N\left(x_{nj} \middle| x_n^*; \mu_{m_nj}(x_n^*), \sigma_{m_nj}^2(x_n^*)\right)}} - \frac{\pi_{m_n}N\left(x_n^*; \mu_{m_n}^*, \Sigma_{m_n}^*\right)}{\sum_{l=1}^{K}\pi_l N(x_n^*; \mu_l^*, \Sigma_l^*)}\right|$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\left[\frac{\pi_{m_n}N\left(x_n^*; \mu_{m_n}^*, \Sigma_{m_n}^*\right)}{\sum_{l=1}^{K}\pi_l N(x_n^*; \mu_l^*, \Sigma_l^*)\frac{N\left(x_{nj} \middle| x_n^*; \mu_{lj}(x_n^*), \sigma_{lj}^2(x_n^*)\right)}{N\left(x_{nj} \middle| x_n^*; \mu_{m_nj}(x_n^*), \sigma_{m_nj}^2(x_n^*)\right)}} - \frac{\pi_{m_n}N\left(x_n^*; \mu_{m_n}^*, \Sigma_{m_n}^*\right)}{\sum_{l=1}^{K}\pi_l N(x_n^*; \mu_l^*, \Sigma_l^*)}\right]$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\frac{\pi_{m_n}N\left(x_n^*; \mu_{m_n}^*, \Sigma_{m_n}^*\right)}{\sum_{l=1}^{K}\pi_l N(x_n^*; \mu_l^*, \Sigma_l^*)\exp\left\{-\frac{1}{2\sigma_j^2}\left|2x_j - \mu_{lj}(x^*) - \mu_{m_nj}(x^*)\right|\epsilon_j(m_n, l)\right\}}$$

$$-\frac{1}{NK}\sum_{n=1}^{N}\frac{\pi_{m_n}N\left(x_n^*;\mu_{m_n}^*,\Sigma_{m_n}^*\right)}{\sum_{l=1}^{K}\pi_l N(x_n^*;\mu_l^*,\Sigma_l^*)}$$

Hence the lower bound of $RI(j)$ is an increasing function of conditional mean difference of two clusters $\epsilon_j(l,m)$. If $\epsilon_j(l,m) = 0$ for any $l,m \in \{1,2 \dots, K\}$ and conditional variances are the same for all clusters, the ratio between any two components equals to 1. By equation (2.8), $RI(j) = 0$.

End of Proof.

We prove that an irrelevant feature having the same mean and variance among all clusters given all the other features has an *RI* being zero. In addition, the bigger mean difference between clusters can lead to a bigger lower bound value of the *RI*. If the lower bound of $RI(j)$ is small, the mean differences are small indicating the j^{th} feature does not contribute to clustering. One advantage of *RI* over mean differences used in [26] is *RI* considers both mean difference as well as variance differences. As shown in equation (2.8), the mean and the variances are both incorporated in the formula. Often, there are cases that the means from two clusters are the same, but the variances differ. Simply using the mean difference will not capture the features contributing. Here we use an illustration example to explain this idea.

[**Illustration example**] For a two-dimensional dataset X, there are 100 samples. The first feature is generated from normal distribution $N(3{,}1)$. The second feature is generated from two normal distributions with 50 samples from $N(1.5{,}1)$ and another 50 samples from $N(1.5{,}10)$. In the setting, only the second feature contributes to clustering. If we use the mean differences, both clusters have [3, 1.5] as the mean, the conclusion is both features are not contributing to the clustering. Now, if we study the EM implementation, given some initialized mean and variances, the *RI* values are calculated by equation (2.8), we get:

$$RI(1) = 0.030 \quad RI(2) = 0.461$$

We conclude the second feature is contributing while the first feature should be treated as an irrelevant feature based on the *RI*. This simple example illustrates the advantages of *RI* over the mean differences. Some experiments to validate the efficacy of *RI* are discussed in the following sections.

2.4 Experiments

In the section, two synthetic datasets, four benchmark dataset, and one medical application (Alzheimer's Disease) dataset are studied to demonstrate the performance of the proposed ESM algorithm. On the benchmark dataset, we compare the proposed algorithm with other existing model-based feature selection algorithms from R software packages. Please note our proposed ESM develops one model within which we decide the features to be kept or removed. In comparison, both the forward feature selection algorithm [19] and backward feature selection algorithm [14] need to develop a large number of models with each requiring the parameter estimations. For the dataset with a large number of features, we expect the computational advantages of the proposed ESM will show. Also, our experiments on the synthetic datasets indicate all three algorithms, including ours, can identify the relevant features, here we choose to report the experimental results from our proposed ESM comparing to EM in the following sections.

Since the ground truth of all the datasets are known, we use the following two metrics for the performance evaluation: (1) Relevancy Feature Selection (RFS): the percentage of relevant features being selected; (2) Accuracy: the percent of instances correctly clustered.

2.4.1 Experiments on Synthetic Datasets

We design both synthetic datasets with two relevant features, and two clusters. Additional irrelevant features are generated as “noise features”. In the first dataset, we have 10 features, all independent. As in most real-world data, there are some dependencies between features. To capture the dependency between the feature sets, we generated some correlated features in the second dataset.

2.4.1.1 Experiments I

In this experiment, we get 10 features with f_1 and f_2 being the relevant features for clustering, and the other 8 features are irrelevant features. The relevant features are simulated from a two-component mixture of Gaussian distributions with the equal number of data points for each component. The irrelevant features are randomly generated from normal distributions. Since the data size may affect the performance of clustering performance, we create the 10 datasets from 100 to 1000 data points with 100 increments (see Table 2 for the experiment settings). For each setting, 10 experiment runs are conducted.

Figure 1 illustrates that the clustering performance varies with different data sizes. We observe that as the data size increases from 100 to 300, the clustering accuracy improves. The performance remains the same afterward. This concurs with the challenges commonly identified in the clustering research. That is, when the ratio between the number of data points vs. the number of features is small, it is difficult to derive the stable cluster structures. The second observation is that the standard deviation of the accuracy has similar pattern. However, the performance is relatively unstable (large standard deviation) for the experiments on 100 data points and 200 data points depending on the ESM initializations. In the 10 repeated runs, about three to five with good initializations are able to successfully identify the relevant features. Here we argue this should not be a concern since a run with a bad initialization tends to converge slower. It is our intention to design an initialization mechanism to improve the performance as a future task.

Table 2. Experiment Setting for Synthetic Dataset I.

# of features	2 relevant +8 irrelevant
# of groups/components	a mixture of 2 components
# of data points	100, 200,300, 400,500,600,700,800,900,1000
# of repeated runs	10
Distribution of relevant features for each component	$\mu_1 = \begin{pmatrix}-1\\1\end{pmatrix}$ $\mu_2 = \begin{pmatrix}2\\-1\end{pmatrix}$ $\Sigma_1 = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$ $\Sigma_2 = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$

Distribution of irrelevant features	$f_3 = N(1.5, 1)$ $f_4 = N(3, 0.5)$ $f_5 = N(1.8, 0.9)$ $f_6 = N(2.7, 1.5)$ $f_7 = N(0.3, 0.5)$ $f_8 = N(0.8, 0.9)$ $f_9 = N(-2, 0.5)$ $f_{10} = N(-3, 0.9)$
# of repeated runs	10

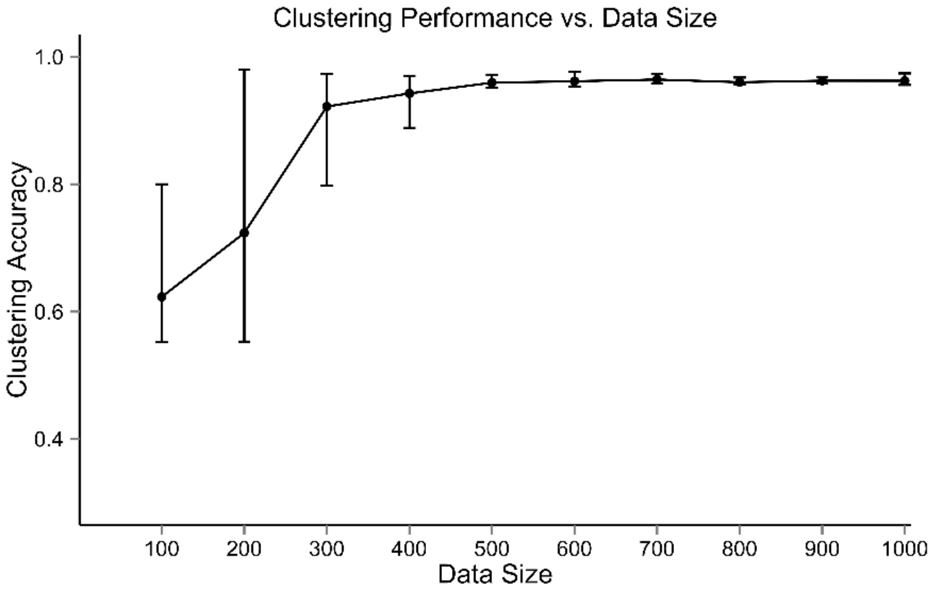


Figure 4. Synthetic Dataset I: The Clustering Performance vs. Data size.

As shown in Figure 4, the ESM shows good performance when the ratio reaches 30 (300 data points, 10 features). We will use this experiment setting to illustrate the value of the *RI* in feature selection.

Figure 5 shows the trace of the RI for each feature over the iterations. One sharp observation is both f_1 and f_2 have significantly higher RI values than the other 8 irrelevant features. Besides, the accuracy of ESM is 97.0% while the accuracy of EM (on the full feature set) is only 71.5%. We conclude ESM can identify the relevant features (independent features) and exclude the irrelevant "noisy features" resulting in much-improved clustering performance.

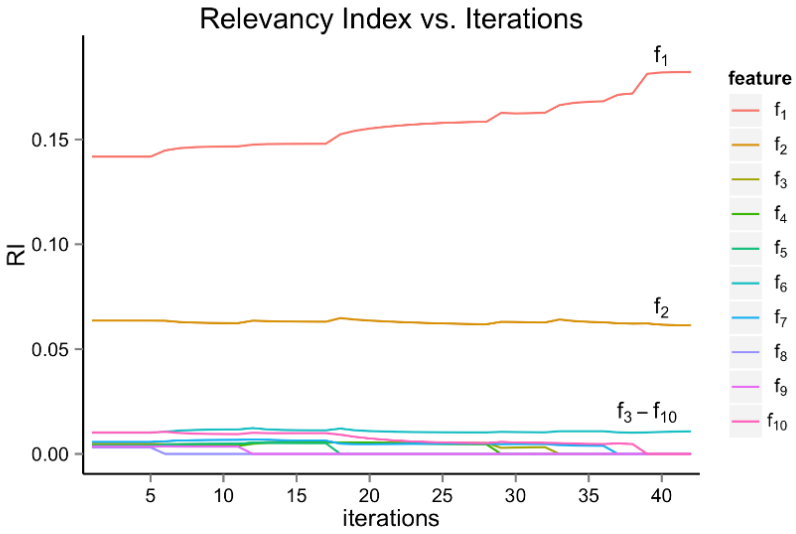


Figure 5. Relevancy Index Value for Each Feature Over Iterations on 300 Synthetic Data Points.

2.4.1.2 Experiments II

In this experiment, we include 15 features with f_1 and f_2 being the relevant features for clustering and remaining 13 features being irrelevant features. The relevant features are generated from two-component mixture of Gaussian distributions with 225 data points for each component. In addition, we purposely add correlations between the features including $f_1 \sim f_2$, $f_{12} \sim f_{13}$ and $f_{14} \sim f_{15}$ correlations. The experiment setting is summarized in Table

3. Similarly, 10 runs are conducted for each experiment. Figure 6 shows both f_1 and f_2 have significantly higher RI values than the other 13 irrelevant features. In addition, the accuracy of ESM is 97.5% while the accuracy of EM on the full feature set is only 84.7%. We conclude ESM is able to identify the relevant features with dependencies and exclude the irrelevant "noisy features" resulting much improved clustering performance.

Table 3.Experiment Setting for Synthetic Dataset II.

# of features	2 relevant + 13 irrelevant
# of groups/components	a mixture of 2 components
# of data points	450
# of repeated runs	10
Distribution of relevant features for each component	$\mu_1 = \begin{pmatrix}-1\\2\end{pmatrix}$ $\mu_2 = \begin{pmatrix}2\\-1\end{pmatrix}$ $\Sigma_1 = \begin{pmatrix}1 & 0.1\\0.1 & 1\end{pmatrix}$ $\Sigma_2 = \begin{pmatrix}1 & 0.1\\0.1 & 1\end{pmatrix}$
Distribution of irrelevant features	$f_3 = N(1.5, 1)$ $f_4 = N(3,0.5)$ $f_5 = N(1.8,0.9)$ $f_6 = N(0.3,0.5)$ $f_7 = N(2,1)$ $f_8 = N(-2,2)$ $f_9 = N(4,3)$ $f_{10} = N(1.5,0.1)$ $f_{11} = N(-4,2)$ f_{12} *and* f_{13} *are correlated with* $\mu_{12\_13} = \begin{pmatrix}2\\2\end{pmatrix}$ $\Sigma_{12\_13} = \begin{pmatrix}1 & 0.3\\0.3 & 1\end{pmatrix}$ f_{14} *and* f_{15} *are correlated with* $\mu_{14\_15} = \begin{pmatrix}-3\\2\end{pmatrix}$ $\Sigma_{14\_15} = \begin{pmatrix}1 & 0.1\\0.1 & 1\end{pmatrix}$
# of repeated runs	10

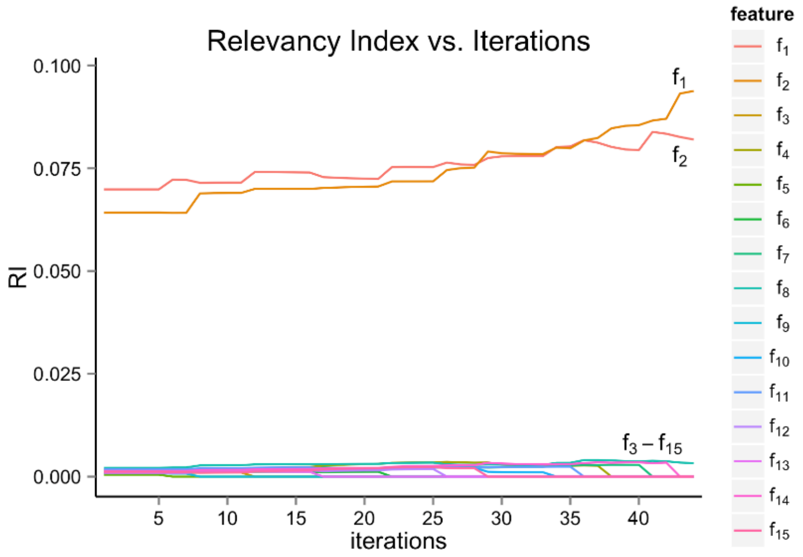


Figure 6. Relevancy Index for Each Feature Over Iterations on 450 Synthetic Data Points.

2.4.2 Benchmark Datasets

In this part, we compare the proposed algorithm with other existing algorithms. To get a fair comparison, we focus on the feature selection algorithms for model-based clustering. The survey paper [27] summarized six GMMs variable selection R packages named as sparcl[28], clustvarsel [29], VarSelLCM [30], vscc [31], SelvarMix [32], bclust [33]. Among the six methods, 'sparcl' mainly performs sparse hierarchical and sparse K-means clustering; the 'bclust' package requires a deliciated initial transformed model. Thus, we exclude these two methods and compare the proposed method with the other four methods. To evaluate the cluster quality, two evaluation metrics, accuracy (ACC) and adjusted rand index (ARI) were computed. In addition, we also report the running time on the following datasets:

G2: A synthetic Gaussian cluster datasets with 2048 rows and 128 columns. The variation is 40 indicating medium level of overlap. The data has two groups [34].

Vowel: An empirical dataset with 990 rows and 9 context-sensitive features. The problem is to recognize a vowel spoken by an arbitrary speaker. The observations fall in eleven groups (different vowels) [35].

Wine: A chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The wine data contains three types of wines with 178 rows and 27 columns.

Crab: An empirical dataset with 200 rows and 8 columns, among which five columns describe the morphological measurements while the remaining three columns are the color (orange and blue), sex (Female and Male) and index that divide the 200 crabs evenly into four groups by the combination of color and sex.

The results are summarized in Table 4. From the results of the experiments, we found that, on this set of benchmark datasets, the proposed ESM method has comparable accuracy with other algorithms but faster speed. The model-based feature selection algorithms are generally slow and can't handle high dimensional data easily. To evaluate the performance on a higher dimensional data, we included the synthetic G2 data, which has a higher dimension than the other three empirical datasets. On this dataset, all algorithms take much longer time than on low dimension empirical datasets. Specifically, we did not report the accuracy for 'clustvarsel' since it takes a stepwise approach, for which the computational time increases exponentially as dimension increases (takes over eight hours). One interesting observation is that all the algorithms can achieve 100% accuracy on G2 synthetic dataset. The reason is that the G2 synthetic dataset is generated from the Gaussian Mixture model without any noise; thus, the model can be easily fitted.

Table 4. Performance Results on Benchmark Datasets.

DATASET	**ALGORITHM**	**ACC**	**ARI**	**TIME(SEC)**
G2	clustvarsel	--	--	-- > 8 hr
	VarSelCluster	1	1	**29.184**
	vscc	1	1	178.893
	selvarclustLasso	1	1	4610.932
	ESM	**1**	**1**	494.896
VOWEL	clustvarsel	0.302	0.118	143.445
	VarSelCluster	**0.377**	**0.211**	64.623
	vscc	0.371	0.181	7.979
	selvarclustLasso	0.339	0.151	16.094
	ESM	0.358	0.169	**3.973**
WINE	clustvarsel	0.910	0.739	13.910
	VarSelCluster	0.944	0.830	4.086
	vscc	**0.978**	**0.931**	8.316
	selvarclustLasso	0.843	0.585	**2.380**
	ESM	0.916	0.755	2.953
CRAB	clustvarsel	**0.935**	**0.840**	1.982
	VarSelCluster	0.355	0.038	1.723
	vscc	0.620	0.428	**0.196**
	selvarclustLasso	0.595	0.354	0.914
	ESM	0.910	0.783	1.255

Another observation from the results table is that the clustering performance depends a lot on the dataset. One algorithm can perform well one dataset while badly on the other. For example, ‘VarSelCluster shows the best accuracy on ‘Vowel’ dataset while worst on the ‘Crab’ dataset. ‘vscc’ shows the best accuracy on ‘Wine’ dataset but wicked accuracy on ‘Crab’ dataset. But luckily, the ESM algorithm has competitive accuracy on all benchmark datasets here. The facts show the potential robustness of the ESM algorithm on different datasets.

2.4.3 Real World Application: Alzheimer’s Disease

Alzheimer’s Disease (AD) is a progressive neurodegenerative disease that is the most frequent type among elderly dementia patients. In the U.S., approximately 5.2 million people over 60 are afflicted by AD [36]. The situation drives a significant amount of research investigating ways to slow down the AD progression and detect AD at an early stage for better treatment or even prevent the disease. Mild cognitive impairment (MCI) is a syndrome defined as cognitive decline more significant than expected for individuals during aging but that does not interfere notably with activities of daily life [37]. It is an intermediate stage between healthy aging with mild cognitive decline and dementia, where cognitive impairment is more severe, even impacting daily function. Though it is distinct from dementia, MCI patients with memory complaints and deficits (amnestic mild cognitive impairment) have high risks of progression to AD [37], [38]. The early diagnosis of the MCI stage is becoming essential when the interventional strategies may be more effective.

Extensive research has investigated the predictive model for AD in hoping to predict the risk of each patient converting to AD, and this is still on-going effort. The focus of this research is to identify the underlying patient cohort structures which may discover patient subtypes for interventional treatment. In this study, we have collected 317 patients’ data from Alzheimer's disease neuroimaging initiative [39], a large-scale online repository designed to identify more sensitive and accurate methods to detect Alzheimer's disease at an earlier stage and mark its progress via biomarkers. Specifically, the baseline data is collected to evaluate the efficacy of our proposed ESM method in AD early detection. Among all these patients, 22 are AD, 172 are MCI, and 123 are Normal Controls (NCs). For each patient, we obtain PET, MR images and cognitive tests (see Table 5).

Table 5. Summary of Features for Alzheimer's Disease Data.

Feature	Notation	Mean ± SE	Category
Age	f_1	72.9 ± 7.3	Demographic
Mini-Mental State Examination (MMSE)	f_2	28.2 ± 2.3	Cognitive Test
Clinical Dementia Rating (CDR) Score	f_3	1.1 ± 1.6	Cognitive Test
Volume of hippocampus	f_4	7157 ± 1160	MRI biomarkers
Volume of ventricles	f_5	35086 ± 19333	MRI biomarkers
Whole Brain	f_6	1048732 ± 112296	MRI biomarkers
Entorhinal	f_7	3677 ± 741	MRI biomarkers
Volume of Intracranial	f_8	1514196 ± 156568	MRI biomarkers
Hypometabolic Convergence Index (HCI)	f_9	10.9 ± 5.6	FDG-PET biomarkers
Statistical region of interest (sROI)	f_{10}	1.2 ± 0.07	FDG-PET biomarkers
mean cortical Standard Uptake Value Ratio with cerebellum as reference region (mcSUVRcere)	f_{11}	1.1 ± 0.2	F-AV45-PET biomarkers
mean cortical Standard Uptake Value Ratio with corpus callosum and centrum semiovale combined as reference region (mcSUVRwm)	f_{12}	0.77 ± 0.17	F-AV45-PET biomarkers

Table 6 presents the correlation matrix of the features. Clearly, some features are highly dependent to each other, for example, the clinical test scores MMSE (f_2) vs. CDR (f_3), MRI biomarkers volume of hippocampus (f_4) vs. whole brain (f_6), FDG-PET biomarkers HCI (f_9) vs. sROI (f_{10}).

One challenge in applying the traditional clustering approach to the data is that some noise features degrade the performance of GMM. Table 7 shows the clustering results from original EM using all 12 features. The overall accuracy of correctly identifying patients to disease types is only 58.68%. For the AD cohort, EM clusters 20 out of 22 AD patients correctly (90.91%). For the NC cohort, EM though, identifies 87 out of 123 correctly, 2 NCs are put into the AD group, and another 34 NCs are labeled as MCI. The accuracy of the NC cluster is 70.73%. The results on the MCI cohort are even worse, with 79 out of 172 are correctly labeled, and 58 MCIs are put into the NC group; remaining 35 are grouped to AD cohorts resulting in 45.93% accuracy.

Table 6. Correlation Matrix

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
f_1	1.00	-0.09	-0.02	-0.37	0.41	-0.34	-0.15	0.02	0.03	-0.35	0.12	0.19
f_2		1.00	**-0.72**	0.44	-0.19	0.19	0.41	0.03	-0.48	0.41	-0.38	-0.47
f_3			1.00	-0.41	0.22	-0.14	-0.42	0.02	**0.55**	-0.45	0.38	0.48
f_4				1.00	-0.33	**0.56**	**0.64**	0.29	-0.39	**0.54**	-0.27	-0.39
f_5					1.00	0.06	-0.11	0.45	0.28	-0.43	0.11	0.44
f_6						1.00	0.49	**0.79**	0.00	0.35	-0.07	-0.05
f_7							1.00	0.30	-0.27	0.38	-0.19	-0.25
f_8								1.00	0.15	0.05	0.04	0.14
f_9									1.00	**-0.65**	0.32	0.49
f_{10}										1.00	-0.30	-0.41
f_{11}											1.00	**0.84**
f_{12}												1.00

Table 7. Confusion Matrix of EM Using All Features

		GroundTruth			
		NC	AD	MCI	Overall accuracy
Clustering	NC	87	0	58	
	AD	2	20	35	
	MCI	34	2	79	
	Total	123	22	172	
	Accuracy	70.73%	90.91%	45.93%	58.68%

Applying the ESM algorithm, three out of 12 features are selected: CDR(f_3), HCI (f_9) and mcSUVRcere (f_{11}). As shown in Figure 7, the three features have significantly higher *RIs* than other features. If we only use the three features, the overall clustering accuracy is 84.86% (Table 8). In comparing the results from Table 7 and Table 8, the accuracy of AD improves from 90.91% to 100%. The accuracy of NC improves from 70.73% to 92.68%, where one NC is mislabeled as AD, and 8 out of 123 NCs are labeled as MCIs. The accuracy of MCI improves from 45.93% to 77.33%, with the majority of the MCIs (133 out of 172) are correctly labeled, one MCI is put into the NC group, and 38 MCIs are put to the AD group.

We want to emphasize that it is not surprising to see the current results on MCI. Clinically, the MCI cohort has subtypes: MCI converter and MCI non-converter. MCI converter refers to the patient positively diagnosed as AD in the follow-up exams. Fortunately, ANDI is a rich data repository with longitudinal data available. We collect the updated patient staging information from the follow-up visit to explore the composition of the MCI group. For illustration purposes, we show a 2D plot of the two most relevant features CDR (f_3) and mcSUVRcere (f_{11}).

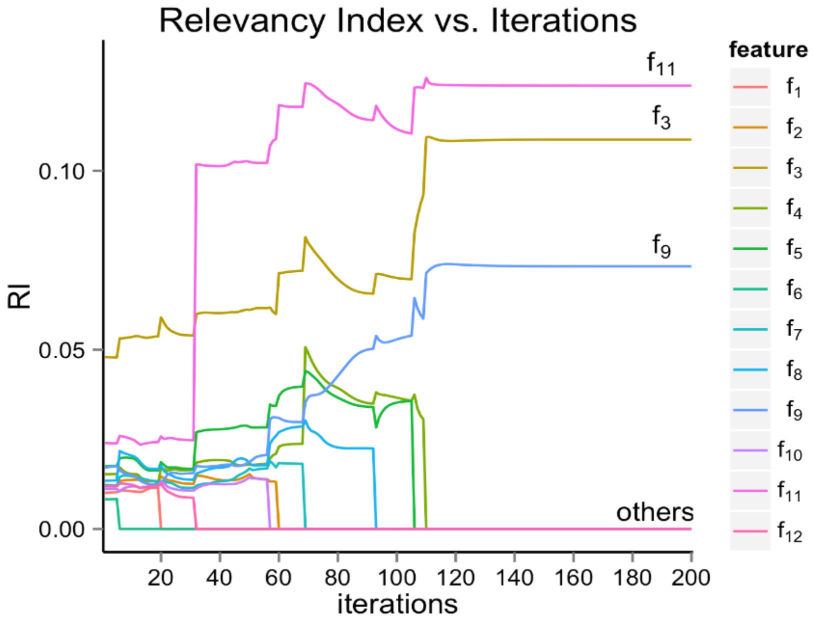


Figure 7. Relevancy Index for Each Feature Over Iterations on AD Data.

Table 8. Confusion Matrix for ESM Clustering.

		GroundTruth			
		NC	AD	MCI	Overall accuracy
Clustering	NC	114	0	1	
	AD	1	22	38	
	MCI	8	0	133	
	Total	123	22	172	
	Accuracy	92.68%	100%	77.33%	**84.86%**

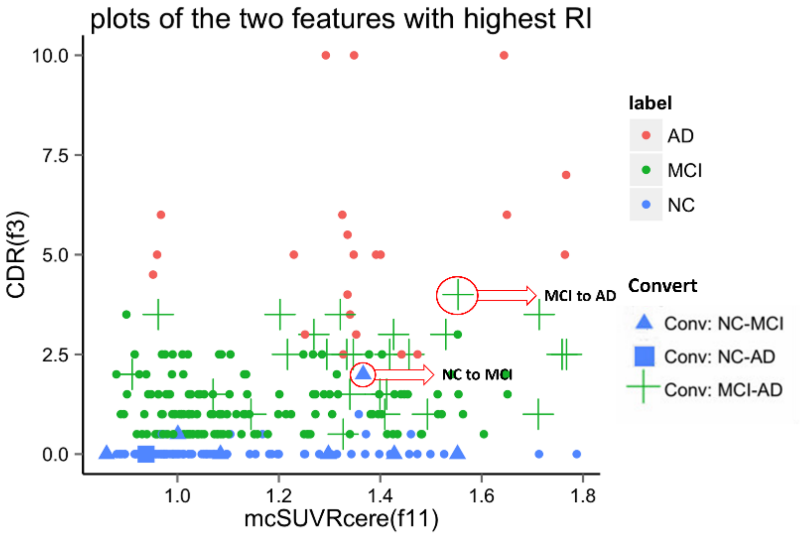


Figure 8. Clustering Shown in Two Feature Space: CDR (f_3) and mcSUVRcere (f_{11}).

The points are colored by the real diagnosis results: AD (red), MCI (green) and NC (blue). The shapes signify the converters: triangle for NC converting to MCI, square for NC converting to AD, and cross for MCI converting to AD. In Figure 8, 16 out of 26 green crosses are on the boundary between MCI and AD clusters, but close to AD. That is, among the 172 MCIs, 26 are staged as AD in the follow-up visit. Using baseline data, the 17 out of 38 MCIs mislabeled as AD (Table 8) are indeed converted. The blue triangle represents the patient converting from NC to MCI. In the 7 blue triangles, there is one particular point, which is diagnosed as NC in the first visit. However, in clustering, the point is closer to MCI than NC in Figure 8 and mislabeled as MCI in Table 8, which is verified by the diagnosis of MCI in the second visit. Indeed, the clustering technique can help to capture the convert between disease types.

2.5 Conclusion

Gaussian Mixture Model (GMM), as a soft clustering methodology, has attracted considerable attention due to the distinct advantages from its statistical foundation. However, its performance deteriorates notably if the dataset has many noisy features irrelevant to the clustering process. This research proposes an ESM algorithm based on a new metric: relevancy index (*RI*). The traditional EM algorithm for GMM modeling parameter estimation is extended with an S step using *RI* for feature selection. ESM preserves the good properties of the EM algorithm, such as guaranteed convergence and optimum determination of the clustering number. To evaluate the performance of ESM algorithm, we conduct experiments on two synthetic datasets (with independent features, with dependent features), four benchmark datasets and one Alzheimer's Disease (AD) dataset. The experiments on synthetic datasets show that ESM can identify the relevant features and improved clustering accuracy comparing to EM. The experiments on four benchmark datasets show that ESM has a competitive performance on accuracy and running time compared with existing algorithms. Other than improved clustering results, the experiment on AD indicates that ESM may potentially identify the patient subtypes, which is crucial for patient treatment planning. While promising, the algorithm is limited for applications on more complex data such as mixed data with both continuous and categorical features. In the future, we may tackle the issue for more general data types.

CHAPTER 3

FEATURE SELECTION FOR GENERALIZED MODEL OF MIXTURE

3.1 Background

Most existing clustering algorithms are designed to tackle single data type (e.g., continuous, categorical) thus limit its applications to the real-world problems which often contain mixed data types. The intuitive solution to a mixture dataset is to convert the data set to a single type by either transforming the categorical features to numbers or converting the continuous features into categorical features. Classic clustering methods can then be applied afterwards. One example is to dummy coding all categorical features to continuous features [40]. There are three issues associated with this. First, the dimension of data set is increased, and this may cause problems when the number of categorical features or the level of the categorical feature is large. Second criticism is the semantic similarity in the original data set may be lost during the transformation [41]. The third issue is it is non-trivial to give correct numeric values to categorical values like color [42]. While converting continuous features to categorical ones may be less problematic, the discretization process may again lose information [42]. An alternative approach on mixture dataset is to define new distance measures and cost function designed specifically for the types of the data. Enormous efforts have been invested on improving k-means and k-prototypes clustering algorithm. For example, based on k-prototypes, new measures such as Gower's distance [43] are introduced to calculate the dissimilarity between data objects and prototypes of clusters. Gower's distance usually involves weights specified for the distance of continuous features and the distance of categorical features. As criticized by Foss et al. (2016), determining the weights is critical for the clustering and yet there is currently no explicit guideline on how to assign the weights for optimal clustering outcomes. The third approach on mixed data takes ensembled methods [44]. The idea is simple. The mixed data set is first divided into two sub-datasets: the categorical dataset and the continuous dataset. Traditional clustering algorithms designed for different types of datasets are applied respectively and the clustering results on the two sub-datasets are combined via a sequential combination method. The main issue of the ensemble method is that the clustering algorithms are biased by partial of the dataset. And, using single type of data does not take advantage of the complementary information from other data.

The three approaches reviewed above are mostly instance-based focusing on the data points instead of the dataset distributions. One may argue that converting the categorical features into continuous features from the first approach does use the normal distribution as the guideline. However, this is under the assumption that the categorical data can be represented as continuous data from a normal distribution. The true distribution on the original categorical data is not utilized. In contrast, model-based methods look into the probabilistic distribution from the true data. It assumes that the instances (data points) are generated from a mixture of underlying probability distribution. Literature terms this approach as model-based clustering (Banfield & Raftery 1993; Bensmail et al. 1997; Fraley & Raftery 1998a, 1998b), mixture likelihood clustering (McLachlan & Basford 1988;

Everitt 1993), mixture-model clustering (Jorgensen & Hunt 1996; McLachlan, et al. 1999) and Latent Class cluster analysis (Vermunt & Mgidson, 1996). In general, model-based approaches (a.k.a. Latent Class models) have several advantages in clustering mixed type data. (1) Statistical metrics (e.g., mean and variance) derived from the data distribution instead of the distance between the data points are used in the modeling. The model uses the probability to describe the degree of the data point belonging to the cluster and the probability is updated iteratively during the clustering. Since most real world problems are uncertain by nature, the use of this “soft assignment” approach may be a better alternative comparing to the “hard assignment” (e.g., k-means) [45][8]. (2) Latent Class model enjoys the flexibility in choosing the distribution forms for each component. The continuous features can be modeled as normal distribution while the categorical features can be estimated from multinomial or Poisson distribution. In addition, some restrictions can be imposed on the model parameters to simplify the model structure and avoid overfitting. For example, the covariance matrix can be restricted to be diagonal for high dimensional data to reduce the number of parameters. Formal statistical test can also be applied to check the validity of parsimonious model [8] [9]. (3) The third advantage of the Latent Class model is it is scale-free, that is, the clustering results are independent from the data being normalized or not. For distance-based clustering algorithms like k-means, the scaling has been one of the main criticisms. Especially when handing mixed type of data, the categorical feature may influence the scaling of continuous features [1]. (4) Latent Class models have formal criteria such as Akaike information criterion (AIC) and Bayesian information criterion (BIC) to decide the number of clusters [18].

While the Latent Class model may have great potentials in handling mixed type of data, it faces some challenges for high-dimensional dataset. It is noted that among the large number of features, some may not truly contribute to delineate the cluster profiles. Inclusion of these "noisy features" requires more parameter estimations, which cost unnecessary computational overhead. Additionally, the noisy features will confuse the model to identify the true structure of the clusters [13]. Recognizing the issue, in this research, we first develop a Generalized Model of Mixtures (GMoM) where a multivariate normal distribution is employed to describe continuous features, Bernoulli distribution and multinomial distribution are used to model binary and nominal features respectively. Next, we propose a novel Feature Index (*FI*) based on Kullback-Leibler (KL) divergence. *FI* is a measure based on the posterior probability of assigning data points to the cluster groups. If the inclusion and exclusion of a feature show no significant difference on the *FI* measure, we conclude this feature is not truly contributing to the clustering. Thus, *FI* can be used to rank and select important features for clustering and reduce the dimensionality of the feature space. One unique advantage of our proposed approach is the assessment of *FI* can be naturally embedded in the model parameters estimation procedure. The parameter estimation procedure for mixture models such as Expectation Maximization (EM) are often iterative and can be computational expensive. The classical feature selection technique such as wrapper that cycles between selecting feature subsets and estimating model parameters, can become unfeasible in the feature selection for mixture model setting. This embedded approach that simultaneously select features and estimate model parameters can be more computational efficient compared to wrapper methods.

The remainder of this study is organized as follows. Section 2 reviews the basics of Latent Class Model and EM algorithm. Section 3 presents the proposed algorithm in detail. Next, three sets of experiments on one synthetic dataset, one benchmark dataset and one real application dataset are illustrated in Section 4. In Section 5, the conclusion and future direction are presented.

3.2 Review of Latent Class Model and EM

Suppose we have a data matrix $X \in R^{N\times p}$ with *N* data points and p features in which the n^{th} row is $\boldsymbol{x}_n^T = (x_{n1}, x_{n2}, \dots, x_{np})$. Let $(x_1, x_2, \dots, x_p)$ be a vector of p features where each feature can be continuous, binary or nominal. Let $x_{n,i}$ be the value of the n^{th} sample for the i^{th} feature.

In the Latent Class model, we assume that the data can be grouped into *K* clusters. For each cluster k is an associated probability π_k. The joint distribution of the observed features is a finite mixture of probabilities $g(\boldsymbol{x}_n|k)$:

$$f(x_n) = \sum_{k=1}^{K} \pi_k g(\boldsymbol{x}_n|k)$$

The probability density function $g(\boldsymbol{x}_n|k)$ is discussed for binary, nominal, and continuous features separately [45]. Specifically,

- For a binary feature, we take the Bernoulli distribution:

$$g(x_i|k) = p_{ik}^{x_i}(1 - p_{ik})^{1-x_i} \tag{3.1}$$

- For nominal features, the indicator feature x_i is replaced by a vector-valued indicator function with its sth element being defined as

$$x_{i(s)} = \begin{cases} 1, & if\ the\ response\ falls\ in\ category\ s, \quad for\ s = 1,2,..c_i, \\ 0, & otherwise \end{cases}$$

where c_i denotes the number of categories of feature i and $\sum_{s=1}^{c_i} x_{i(s)} = 1$. The distribution assumed is multinomial

$$g(x_i|k) = \prod_{s=1}^{c_i} \left(p_{ik(s)}\right)^{x_{i(s)}} \tag{3.2}$$

where $p_{ik(s)}$ is the probability that an object who is in class k will belong to category s for feature i.

- For continuous features, normal distribution is employed for each single continuous feature:

$$g(x_i|\mu_{ik}, \sigma_i^2) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{1}{2\sigma_i^2}(x_i - \mu_{ik})^2\right] \tag{3.3}$$

where μ_{ik} is the location parameter of the continuous feature x_i in class j and σ_i^2 is the variance of the i^{th} feature taken as constant across classes. However, normal distribution neglects the correlations between the continuous features. We extend Latent Class model by relaxing the independence assumptions on the continuous features using multivariate normal distribution. Among the p features, let the first c features are continuous, we have:

$$g\left(\boldsymbol{x}_{n(1:c)}\middle|k\right) = N\left(\boldsymbol{x}_{n(1:c)}\middle|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{x}_{\boldsymbol{n}(1:c)} - \boldsymbol{\mu}_k\right)' \Sigma_k^{-1}\left(\boldsymbol{x}_{n(1:c)} - \boldsymbol{\mu}_k\right)\right\} \quad (3.4)$$

where $\boldsymbol{\mu}_k$ is the mean vector of the continuous feature $\boldsymbol{x}_{n(1:c)}$ in class k and Σ_k is the covariance matrix.

Then the joint likelihood of n^{th} sample belonging to k^{th} group is

$$g(\boldsymbol{x}_n|k) = g\left(\boldsymbol{x}_{n(1:c)}\middle|k\right) \times \prod_{i=c+1}^{p} g\left(x_{n,i}\middle|k\right) = \prod_{i=1}^{p} h(x_{ni}|k), \quad (3.5)$$

Note the correlations among continuous features can be well represented by multivariate normal distribution. For mixed-type data, literature shows existing methods typically adopt a normal-multinomial finite mixture [46]–[50]. Location model [51] can be employed to allow a distinct distribution for the continuous variables for each unique combination of categorical levels. This approach accounts for any possible dependence structure between continuous and categorical variables, however, it becomes infeasible when the number of categorical variables or number of levels within each categorical variable is large [41]. In addition, deriving all possible dependence structures indeed is to “blend” the categorical features into the structure. As a result, the true identity of the categorical features may be lost which in turn, will impede the feature selection process. Therefore, in this research, we relax the dependence assumptions on the categorical features (binary and nominal) and assume the categorical features are independent to each other.

Given the probability density function $g(\boldsymbol{x}_n|k)$ for binary, nominal, and continuous features, Latent Class model aims to the maximize the log-likelihood of the full data for all features, which is shown in equation (3.6).

$$L = \sum_{n=1}^{N} \ln f(x_n) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k g(\boldsymbol{x}_n|k) \tag{3.6}$$

EM is a commonly used four-step algorithm to estimate the parameters including mixing coefficients π_k and parameters for each distribution, e.g., mean μ_k and covariance Σ_k for normal distribution. The algorithm starts from initializing parameters and evaluates the initial value of log likelihood function. In the second step (known as E step), the EM evaluates the responsibilities under the current parameter settings. The responsibility is defined as the probability of assigning a data point to a specific clustering group:

$$\gamma(z_{nk}) = p(z_k = 1|\boldsymbol{x}_n) = \frac{p(z_k = 1)p(\boldsymbol{x}_n|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x}_n|z_j = 1)} = \frac{\pi_k g(\boldsymbol{x}_n|k)}{\sum_{j=1}^{K} \pi_j g(\boldsymbol{x}_n|k)} \tag{3.7}$$

In the third step, the EM re-estimates the parameters given the responsibilities. The estimation method is maximum likelihood, and therefore the third step is called “M step”. The final step is to check the convergence of log-likelihood. If the log likelihood difference between two iterations is small enough (e.g., less than a small number), it is converged. Otherwise, the algorithm goes back to the E step initiating the next iteration. For details of the EM algorithm, interested readers are referred to [25]. In the next section, we present the Feature Index used for feature selection on this Generalized Model of Mixtures embedded in the EM algorithm.

3.3 Proposed Method

3.3.1 Feature Index

Our proposed Feature Index (*FI*) takes advantage of the responsibility measures in the E step. Let us consider the responsibilities $\gamma(z_{nk})$, the probability of assigning the data point $\boldsymbol{x}_n$ to cluster *k*, if we remove one specific feature, responsibilities shall change. Specifically, let the full feature space with p features be $F = \{x_1, x_2, \dots, x_p\}$, the feature space excluding feature j be $F_j^- = \{x_1, x_2, \dots, x_p\} \backslash \{x_j\}$. Here, we denote the responsibility on the full feature space as $\gamma^F(z_{nk})$ and the responsibility on the reduced feature space (excluding feature j) as $\gamma^{F_j^-}(z_{nk})$ which is related to the n^{th} data point and the k^{th} cluster.

To capture the difference between $\gamma^F(z_{nk})$ and $\gamma^{F_j^-}(z_{nk})$, we use the Kullback-Leibler, or simply KL divergence [52]. The KL divergence is closely related to relative entropy, information divergence, and information for discrimination. It is a non-symmetric measure of the distance between two probability distributions $p(x)$ and $q(x)$. The KL divergence of $q(x)$ from $p(x)$ is defined as

$$D_{KL}(p(x)|q(x)) = \sum_{x \in X} p(x) \, ln \frac{p(x)}{q(x)},$$

which measures the information lost when $q(x)$ is used to approximate $p(x)$. The reason of choosing KL divergence instead of other distance measure such as Euclidian metric is that KL divergence has a statistical meaning, which is helpful in handling probability distributions. The statistical properties are further explained in the theoretical analysis section.

Based on the definition of KL divergence, the proposed *FI* is defined as the KL divergence between two responsibilities averaged over *N* data points and *K* clustering groups. Since the two responsibilities are calculated under the full feature set and the feature set excluding the j^{th} feature respectively, the divergence between the two responsibilities reveals the importance of j^{th} feature to the clustering. It is written as:

$$FI(j) = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^{F}(z_{nk})\ln\frac{\gamma^{F}(z_{nk})}{\gamma^{F_j^-}(z_{nk})} \tag{3.8}$$

The assumption behind our proposed method is that if $FI(j)$ is smaller than a pre-defined threshold, the contribution of feature j in deciding the assignments of data points to clusters is trivial thus can be neglected. The feature j can be removed during the feature selection process under the condition that $FI(j)$ converges over the iterations. Concerning the convergence criteria for FI, we evaluate the changes of $FI(j)$between the current and previous iteration, let say, if it is less than a small number, e.g., 0.0005, we conclude $FI(j)$ converges. Regarding the pre-defined threshold for feature selection, it can be set based on the approximate number of features to be selected (see experiments in Section 4 for details). The pseudo code of the algorithm is described in Table 9. The algorithm starts with the initialization step and iterates between E step and M step until converged as traditional EM algorithm. On top of the traditional EM algorithm, an additional step termed S step is added to select features while updating the parameters of the Generalized Model of Mixtures.

Table 9. Pseudo Code on Extending EM With Feature Index on Generalized Model of Mixtures.

```
1.  Initialize the parameters which is from a converged Latent Class model.
2.  E step. Evaluate the responsibilities using the current parameter values
        $$\gamma^F(z_{nk}) = \frac{\pi_k \prod_{i=1}^{p} h(x_{ni}|k)}{\sum_{l=1}^{K} \pi_l \prod_{i=1}^{p} h(x_{ni}|l)}$$
    and responsibilities after excluding each feature
        $$\gamma^{F_j^-}(z_{nk}) = \frac{\pi_k \prod_{i\neq j}^{p} h(x_{ni}|k)}{\sum_{l=1}^{K} \pi_l \prod_{i\neq j}^{p} h(x_{ni}|l)} \; for\ j = 1,2,..p$$
3.  S step. Calculate the difference between responsibilities before and after excluding $j^{th}$ feature at iteration t.
        $$FI(j)^{(t)} = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K} \gamma^F(z_{nk}) \ln \frac{\gamma^F(z_{nk})}{\gamma^{F_j^-}(z_{nk})}$$
    If $\left|FI(j)^{(t+1)} - FI(j)^{(t)}\right| < \epsilon$ (converged) and $FI(j)^{(t)}$ is small enough, then discard the feature with smallest $FI$ and update the full feature space $F$.
4.  M step. For reduced data with feature space $F$, re-estimate the parameters using the current responsibilities
    for binary features:      $p_{ik}^{new} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk}) x_{n,i}$
    for nominal features:     $p_{ik(s)}^{new} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk}) x_{n,i(s)}$
    for continuous features:  $\boldsymbol{\mu}_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n$
                              $$\Sigma_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{new})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{new})'$$
    posterior probability:    $\pi_k^{new} = \frac{N_k}{N}$
5.  Evaluate the log likelihood
        $$L = \sum_{n=1}^{N} \ln f(x_n) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k g(\boldsymbol{x}_n|k)$$
    If the parameters or the log likelihood are not converged, go back to step 2.
```

3.3.2 Theoretical Analysis on Feature Index

In this section, the theoretical analysis on the statistical properties of *FI* is conducted to justify the use of feature index.

The proposed *FI* is the KL divergence between two responsibilities averaged over *N* data points and *K* clustering groups. In the following equation, $g(x)$ function is defined as in equation (3.1), (3.2) and (3.4) for bernoulli, multinomial and multivariate normal distribution respectively. The *FI* is calculated by equation (3.8). We decompose the *FI* into two parts as shown in equation (3.9) and (3.10). Note that in the following decomposition, the continuous features are the first c out of p features as used in equation (3.4) before.

If the j^{th} feature is binary or nominal,

$$FI(j) = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^{F}(z_{nk})\ln\frac{\gamma^{F}(z_{nk})}{\gamma^{F_{\bar{j}}}(z_{nk})}$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^{F}(z_{nk})\ln\left\{\frac{\dfrac{\pi_k g\left(\boldsymbol{x}_{n(1:c)}\middle|k\right)\times\prod_{i=c+1}^{p} g\left(x_{n,i}\middle|k\right)}{\sum_{l=1}^{K}\pi_l g\left(\boldsymbol{x}_{n(1:c)}\middle|l\right)\times\prod_{i=c+1}^{p} g\left(x_{ni}\middle|l\right)}}{\dfrac{\pi_k g\left(\boldsymbol{x}_{n(1:c)}\middle|k\right)\times\prod_{i\neq j}^{(c+1):p} g\left(x_{ni}\middle|k\right)}{\sum_{l=1}^{K}\pi_l g\left(\boldsymbol{x}_{n(1:c)}\middle|l\right)\times\prod_{i\neq j}^{(c+1):p} g\left(x_{ni}\middle|l\right)}}\right\}$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^{F}(z_{nk})\ln\left\{g\left(x_{nj}\middle|k\right)\frac{\sum_{l=1}^{K}\pi_l g\left(\boldsymbol{x}_{n(1:c)}\middle|l\right)\times\prod_{i\neq j}^{(c+1):p} g\left(x_{ni}\middle|l\right)}{\sum_{l=1}^{K}\pi_l g\left(\boldsymbol{x}_{n(1:c)}\middle|l\right)\times\prod_{i=c+1}^{p} g\left(x_{ni}\middle|l\right)}\right\}$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^{F}(z_{nk})\ln\{g\left(x_{nj}\middle|k\right)\}$$

$$+\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^{F}(z_{nk})\ln\left\{\frac{\sum_{l=1}^{K}\pi_l g\left(\boldsymbol{x}_{\boldsymbol{n}(\boldsymbol{1}:\boldsymbol{c})}\middle|l\right)\times\prod_{i\neq j}^{(c+1):p} g\left(x_{ni}\middle|l\right)}{\sum_{l=1}^{K}\pi_l g\left(\boldsymbol{x}_{\boldsymbol{n}(\boldsymbol{1}:\boldsymbol{c})}\middle|l\right)\times\{\prod_{i\neq j}^{(c+1):p} g\left(x_{ni}\middle|l\right)\}\times g\left(x_{nj}\middle|l\right)}\right\} \qquad (3.9)$$

If the j^{th} feature is continuous, let $\boldsymbol{x}^{*}_{\boldsymbol{n}\,(\boldsymbol{1}:\,\boldsymbol{c})}$ be the corresponding vector of $\boldsymbol{x}_{\boldsymbol{n}(\boldsymbol{1}:\,\boldsymbol{c})}$ after excluding the j^{th} feature.

$$FI(j) = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^{F}(z_{nk})\ln\frac{\gamma^{F}(z_{nk})}{\gamma^{F_{\bar{j}}}(z_{nk})}$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^{F}(z_{nk})\ln\left\{\frac{\dfrac{\pi_k g\left(\boldsymbol{x}_{n(1:c)}\middle|k\right)\times\prod_{i=c+1}^{p} g\left(x_{ni}\middle|k\right)}{\sum_{l=1}^{K}\pi_l g\left(\boldsymbol{x}_{n(1:c)}\middle|l\right)\times\prod_{i=c+1}^{p} g\left(x_{ni}\middle|l\right)}}{\dfrac{\pi_k g\left(\boldsymbol{x}^{*}_{n(1:c)}\middle|k\right)\times\prod_{i=c+1}^{p} g\left(x_{ni}\middle|k\right)}{\sum_{l=1}^{K}\pi_l g\left(\boldsymbol{x}^{*}_{n(1:c)}\middle|l\right)\times\prod_{i=c+1}^{p} g\left(x_{ni}\middle|l\right)}}\right\}$$

Using the conditional probability formula,

$$g(\boldsymbol{x}_{n(1:c)}|l) = g(x_{nj}|\boldsymbol{x}^*_{\boldsymbol{n(1:c)}}) \times g(\boldsymbol{x}^*_{n(1:c)}|l),$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^F(z_{nk})\ln\left\{g(x_{nj}|\boldsymbol{x}^*_{n(1:c)})\frac{\sum_{l=1}^{K}\pi_l g(\boldsymbol{x}^*_{n(1:c)}|l)\times\prod_{i=c+1}^{p} g(x_{ni}|l)}{\sum_{l=1}^{K}\pi_l g(\boldsymbol{x}_{n(1:c)}|l)\times\prod_{i=c+1}^{p} g(x_{ni}|l)}\right\}$$

$$= \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^F(z_{nk})\ln\{g(x_{nj}|\boldsymbol{x}^*_{\boldsymbol{n(1:c)}})\}$$

$$+\frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma^F(z_{nk})\ln\left\{\frac{\sum_{l=1}^{K}\pi_l g(\boldsymbol{x}^*_{n(1:c)}|l)\times\prod_{i=c+1}^{p} g(x_{ni}|l)}{\sum_{l=1}^{K}\pi_l g(x_{nj}|\boldsymbol{x}^*_{\boldsymbol{n(1:c)}})\times g(\boldsymbol{x}^*_{n(1:c)}|l)\times\prod_{i=c+1}^{p} g(x_{ni}|l)}\right\} \quad (3.10)$$

From equations (3.9) and (3.10), the *FI* is decomposed into two: (1) weighted conditional log-likelihood of feature *j* given all other features; (2) weighted log-likelihood ratio for all features vs. all except feature *j*. The first term reveals the absolute contribution of feature *j* to the model and the second term can be considered as the relative contribution of the feature.

Let us delve more into the second terms from equations (3.9) and (3.10). For equation (3.9), the second term is the likelihood ratio shown in equation (3.11),

$$\frac{\sum_{l=1}^{K}\pi_l g(\boldsymbol{x}_{\boldsymbol{n(1:c)}}|l)\times\prod_{i\neq j}^{(c+1):p} g(x_{ni}|l)}{\sum_{l=1}^{K}\pi_l g(\boldsymbol{x}_{\boldsymbol{n(1:c)}}|l)\times\prod_{i\neq j}^{(c+1):p} g(\boldsymbol{x}_{ni}|l)\times g(\boldsymbol{x}_{nj}|l)} \quad (3.11)$$

In equation (3.11), the denominator is the likelihood of the full feature set and the numerator is the likelihood of the feature set excluding j^{th} feature. For each group $l(l = 1,..K)$, the denominator is the numerator times $g(x_{nj}|l)$, the likelihood of n^{th} data point on l^{th} feature, which is between 0 and 1. Hence, this second term is greater or equal to 1. Similarly, for equation (3.10), the likelihood ratio in the second term as follows:

$$\frac{\sum_{l=1}^{K}\pi_l g\left(\boldsymbol{x}_{n(1:c)}^{*}\middle|l\right)\times\prod_{i=c+1}^{p}g(x_{ni}|l)}{\sum_{l=1}^{K}\pi_l g\left(x_{nj}\middle|\boldsymbol{x}_{\boldsymbol{n}(\boldsymbol{1}:\boldsymbol{c})}^{*}\right)\times\boldsymbol{g}\left(\boldsymbol{x}_{\boldsymbol{n}(\boldsymbol{1}:\boldsymbol{c})}^{*}\middle|\boldsymbol{l}\right)\times\prod_{i=c+1}^{p}g(x_{ni}|l)} \tag{3.12}$$

In equation (3.12), for each l, the denominator is the numerator times $g\left(\boldsymbol{x}_{n(1:c)}^{*}\middle|l\right)$, which is between 0 and 1. Thus, the likelihood ratio in Equation (3.10) is also greater or equal to 1.

From equation (3.11) and (3.12), we obtain the conclusion that the second term of $FI(j)$ increases as the likelihood difference between the full model and the model excluding j^{th} feature increases. In special case, when the likelihood of the full model is the same as the model excluding one feature, then the second term is equal to one.

From the analysis, we conclude *FI* is an integrated measure of the absolute and relative contribution of a given feature. A feature deemed to be important to the clustering model can be revealed from two aspects: a large likelihood showing dominating absolute contribution or a large log-likelihood ratio, indicating relative contribution to the full feature set.

In the next section, we will conduct experiments to evaluate the plausibility of the algorithm and illustrate the typical values of the Feature Index.

3.4 Experiments

In the section, one synthetic dataset, one benchmark dataset and one medical application dataset are studied to demonstrate the performance of proposed algorithm. Since the ground truth of all the datasets are known, we use the following two metrics for the performance evaluation: (1) RFS: the percentage of relevant features being selected; (2) Accuracy: the percentage of instances correctly clustered.

3.4.1 Experiments on Synthetic Dataset

We design the synthetic dataset with four relevant features, and two clusters. Additional irrelevant features are generated as "noise features". In this experiment, we include 15 features with f_1, f_2(continuous) and f_{11}, f_{12}(categorical) being the relevant features for clustering and remaining 11 features being irrelevant features. The continuous relevant features are generated from two-component mixture of Gaussian distributions with 300 data points for each component. The categorical relevant features are generated from multinomial distributions for each component. In addition, we purposely add correlations between the features including f_1~ f_2, f_7 ~ f_8 and f_9 ~ f_{10} correlations. The experiment setting is summarized in Table 10. Similarly, 10 runs are conducted for each experiment.

In the experiment, we first test the classical Latent Class Model on the simulated data. The accuracy is 82.83%. Then we test our proposed algorithm on the synthetic dataset. That is, we do feature selection on latent class model. The accuracy improves to 94.63%. In addition, the selected features are f_1, f_2, f_{11} and f_{12}, which are exactly the relevant features we used to simulate the clustering group.

Figure 9 shows all relevant features f_1, f_2, f_{11} and f_{12} have significantly higher *FI* values than the other 11 irrelevant features. We conclude *FI* is able to identify the relevant features with dependencies and exclude the irrelevant "noisy features" resulting much improved clustering performance.

Table 10. Experiment Setting for Synthetic Dataset.

# of features	10 continuous (2 relevant +8 irrelevant) +5 categorical (2 relevant+3 irrelevant)
# of groups/components	a mixture of 2 components
# of data points	600
# of repeated runs	10
Distribution of relevant features for each component (continuous) f_1, f_2	$\mu_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ $\mu_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $\Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & 0.2 \\ 0.2 & 1 \end{pmatrix}$
Distribution of irrelevant features (continuous) $f_3 - f_{10}$	$f_3 = N(1.5, 1)$ $f_4 = N(3, 0.5)$ $f_5 = N(1.8, 0.9)$ $f_6 = N(2.7, 1.5)$ f_7 *and* f_8 *are correlated with* $\mu_{7\_8} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ $\Sigma_{7\_8} = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}$ f_9 *and* f_{10} *are correlated with* $\mu_{9\_10} = \begin{pmatrix} -3 \\ 2 \end{pmatrix}$ Σ_{9_10} $= \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix}$
Distribution of relevant feature (categorical) f_{11}	Component 1: $Multinomial(1, p = (0.7{,}0.2))$ Component 2: $Multinomial(1, p = (0.2{,}0.7))$
Distribution of relevant feature (categorical) f_{12}	Component 1: $Multinomial(1, p = (0.7{,}0.2{,}0.2))$ Component 2: $Multinomial(1, p = (0.2{,}0.4{,}0.5))$
Distribution of irrelevant feature (categorical) $f_{13} - f_{15}$	$f_{13} = Multinomial(1, p = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}))$ $f_{14} = Multinomial(1, p = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}))$ $f_{15} = Multinomial(1, p = (\frac{1}{10}, \frac{1}{10}, \ldots, \frac{1}{10}))$
# of repeated runs	10

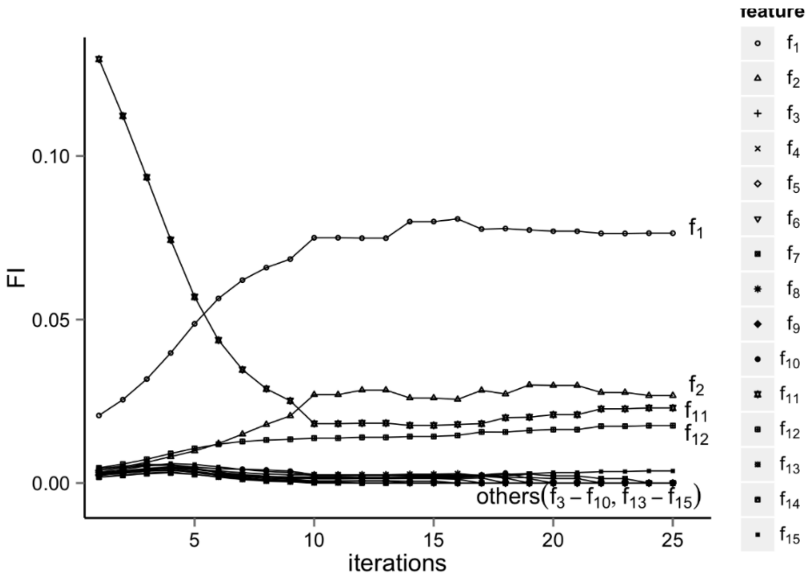


Figure 9. Feature Index for Each Feature Over Iterations on 600 Data Points.

3.4.2 Benchmark Dataset

In this section, we compare our proposed algorithm with other clustering approaches that can handle mixed data on benchmark dataset obtained from UCI Machine Learning Repository [30]. The studied dataset is the heart rate disease dataset provided by Cleveland Clinic. The dataset has 303 instances with 6 numeric and 8 categorical features. The instances are labelled as two classes: healthy or sick (with heart disease). The summary of the features of the heart disease dataset is summarized in Table 11.

In literature, there are four other clustering approaches reporting accuracy on the heart disease data. The K-prototype and K-medoids are two classical clustering approaches that can handle mixed data. The reported best accuracies on heart disease data are 81.0% and 76.5% respectively [53][44]. The third approach is ensemble clustering proposed by Z.He and X.Xu etc. [44]. The general idea of ensemble clustering is to first divide the original mixed dataset into two subsets: the pure categorical and the pure numeric dataset. Then they use the existing clustering algorithms designed for one certain type of dataset to cluster each dataset. Finally, they combine the clustering results on two datasets and generate final clusters. The best accuracy on the heart disease data is 81.3%, which is slightly better than K-prototype. The fourth approach is called UFL Fuzzy ART [53], which extends the basic unsupervised feature learning (UFL) for mixed type data using fuzzy adaptive resonance theory (ART). The best reported accuracy on heart disease data is 81.5%, the highest among all current approaches.

In addition to the four approaches provided in literature, we also test the classical latent class model (LCM) on the heart disease data. The best accuracy obtained by LCM is 78.9%, which is slightly lower than other approaches. Then finally, we employ our proposed feature selection algorithm using feature index on the heart disease data and the best average accuracy is 83.3%, which is higher than any other algorithms listed before. To have a better view of the comparison, the performance of each algorithm is summarized in Figure 10.

After running the proposed feature selection algorithm, we show the importance of each feature by Feature Index illustrated in Figure 11. Feature no.4 (blood pressure), feature no.5 (serum cholestoral) and feature no.11 (the slope of the peak exercise ST segment) have higher Feature Index values than all other features.

Table 11. Summary of Features for Heart Disease Data.

Feature	Notation	Mean ± SE	Category
Age	f_1	54.4 ± 9.1	Continuous
Sex	f_2	0,1	Binary
Chest pain type	f_3	1,2,3,4	Nominal
Resting blood pressure	f_4	131.3 ± 17.9	Continuous
Serum cholestoral in mg/dl	f_5	249.7 ± 51.7	Continuous
Fasting blood sugar>120 mg/dl	f_6	0,1	Binary
Resting electrocardiographic results	f_7	0,1,2	Nominal
Maximum heart rate achieved	f_8	149.7 ± 23.2	Continuous
Exercise induced angina	f_9	0,1	Binary
Oldpeak=ST depression induced by exercise relative to rest	f_{10}	1.1 ± 1.1	Continuous
The slope of the peak exercise ST segment	f_{11}	1,2,3	Nominal
Number of major vessels colored by flourosopy	f_{12}	0,1,2,3	Nominal
Thal: 3=normal; 6=fixed defect; 7=reversible defect	f_{13}	3,6,7	Nominal

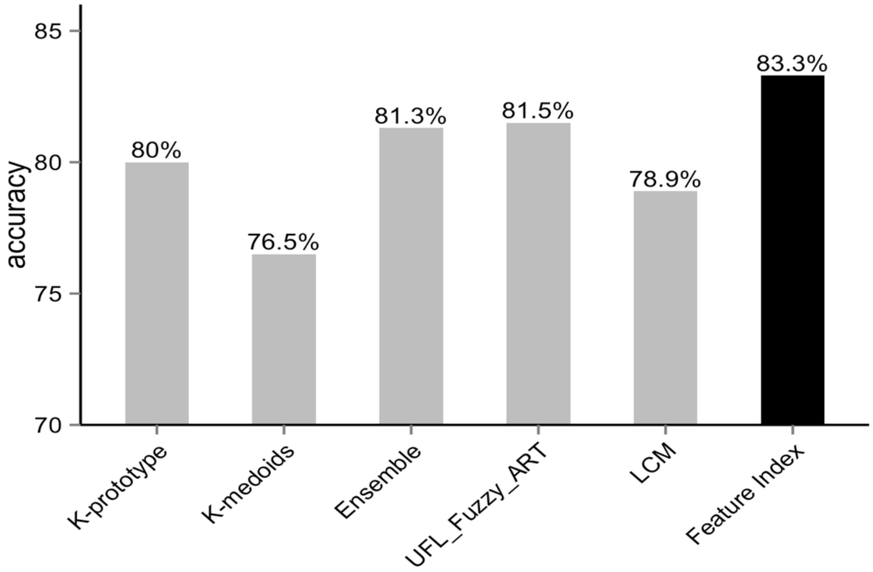


Figure 10. Summary of Accuracy Performance on Heart Disease Data.

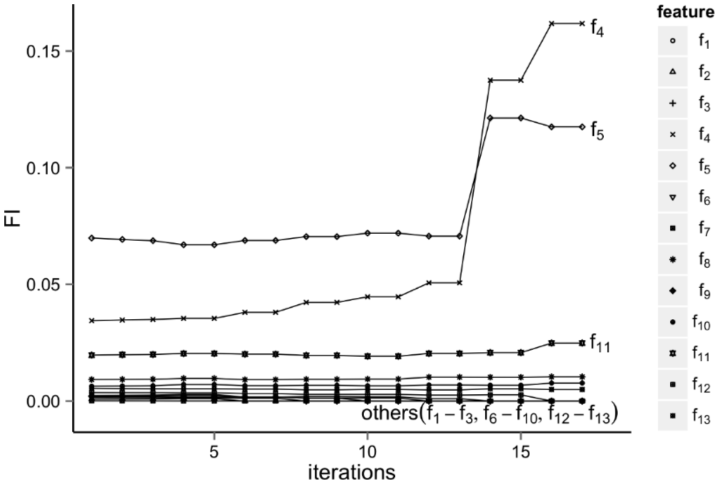


Figure 11.Feature Index for Each Feature Over Iterations on Heart Disease Dataset.

3.4.3 Real World Application: Alzheimer’s Disease

Alzheimer’s Disease (AD) is a progressively neurodegenerative disease which is the most frequent type among elderly dementia patients. In the U.S., approximately 5.2 million people over 60 are afflicted by AD (Alzheimer's Association,2008). This drives a great amount of research investigating ways to slow down the AD progression and detect AD at early stage for better treatment or even prevent the disease. Mild cognitive impairment (MCI) is a syndrome defined as cognitive decline greater than expected for individuals during the course of aging but that does not interfere notably with activities of daily life [37]. It is an intermediate stage between normal aging with mild cognitive decline and dementia where cognitive impairment is more severe even impacting daily function. Though it is distinct from dementia, MCI patients with memory complaints and deficits (amnestic mild cognitive impairment) have high risks of progression to AD [37], [38]. The early diagnosis of MCI stage is becoming essential when the interventional strategies may be more effective.

Extensive research has investigated predictive model for AD in hoping to predict the risk of each individual patient converting to AD and this is still on-going effort. The focus of this research is to identify the underlying patient cohort structures which may discover patient subtypes for interventional treatment. In this study, we have collected 317 patients' data from Alzheimer's disease neuroimaging initiative [39], a large scale online repository designed to identify more sensitive and accurate methods to detect Alzheimer's disease at earlier stage and mark its progress via biomarkers. Specifically, the baseline data is collected to evaluate the efficacy of our proposed method in AD early detection. Among all these patients, 22 are AD, 172 are MCI and 123 are Normal Controls (NCs). For each patient, we obtain PET, and MR images and cognitive tests (see Table 12).

Table 12. Summary of Features for Alzheimer's Disease Data.

Feature	Notation	Mean ± SE	Category
Age	f_1	72.9 ± 7.3	Demographic
Mini-Mental State Examination (MMSE)	f_2	28.2 ± 2.3	Cognitive Test
Clinical Dementia Rating (CDR) Score	f_3	1.1 ± 1.6	Cognitive Test
Volume of hippocampus	f_4	7157 ± 1160	MRI biomarkers
Volume of ventricles	f_5	35086 ± 19333	MRI biomarkers
Whole Brain	f_6	1048732 ± 112296	MRI biomarkers
Entorhinal	f_7	3677 ± 741	MRI biomarkers
Volume of Intracranial	f_8	1514196 ± 156568	MRI biomarkers
Hypometabolic Convergence Index (HCI)	f_9	10.9 ± 5.6	FDG-PET biomarkers
Statistical region of interest (sROI)	f_{10}	1.2 ± 0.07	FDG-PET biomarkers
mean cortical Standard Uptake Value Ratio with	f_{11}	1.1 ± 0.2	F-AV45-PET biomarkers

cerebellum as reference region (mcSUVRcere)			
mean cortical Standard Uptake Value Ratio with corpus callosum and centrum semiovale combined as reference region (mcSUVRwm)	f_{12}	0.77 ± 0.17	F-AV45-PET biomarkers
Gender	f_{13}	Two levels (176:141)	Demographic
AV45	f_{14}	Two levels (202:115)	F-AV45-PET biomarkers
APOE	f_{15}	Three level (22:103:192)	Gene information

Table 13 presents the correlation matrix of the features. Clearly, some features are highly dependent to each other, for example, the clinical test scores MMSE (f_2) vs. CDR (f_3), MRI biomarkers volume of hippocampus (f_4) vs. whole brain (f_6), FDG-PET biomarkers HCI (f_9) vs. sROI (f_{10}).

Table 13. Correlation Matrix.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
f_1	1.00	-0.09	-0.02	-0.37	0.41	-0.34	-0.15	0.02	0.03	-0.35	0.12	0.19
f_2		1.00	**-0.72**	0.44	-0.19	0.19	0.41	0.03	-0.48	0.41	-0.38	-0.47
f_3			1.00	-0.41	0.22	-0.14	-0.42	0.02	**0.55**	-0.45	0.38	0.48
f_4				1.00	-0.33	**0.56**	**0.64**	0.29	-0.39	**0.54**	-0.27	-0.39
f_5					1.00	0.06	-0.11	0.45	0.28	-0.43	0.11	0.44
f_6						1.00	0.49	**0.79**	0.00	0.35	-0.07	-0.05
f_7							1.00	0.30	-0.27	0.38	-0.19	-0.25
f_8								1.00	0.15	0.05	0.04	0.14
f_9									1.00	**-0.65**	0.32	0.49
f_{10}										1.00	-0.30	-0.41
f_{11}											1.00	**0.84**
f_{12}												1.00

One challenge in applying classical latent class model to the data is that the correlations between continuous features are not captured. Another challenge is that some noise features degrade the performance of Latent Class model.

Table 14 shows the clustering results from classical Latent class model using all 15 features. The overall accuracy of correctly identifying patients to disease types is only 59.31%. Table 15 shows the clustering results after feature selection on the generalized model of mixtures. The overall accuracy improves from 59.31% to 84.86% compared with original LCM. For the AD cohort, the proposed algorithm clusters 22 out of 22 AD patients correctly (100%). For the NC cohort, our algorithm identifies 114 out of 123 correctly, one NC is put into the AD group, and another eight NCs are labeled as MCI. The accuracy of NC cluster is 92.68%. The results on MCI cohort also improves compared with LCM using full feature set. The feature selection on the GMoM can identify 133 out of 172 MCI correctly (77.33%) with only one mislabeled as NC and 38 mislabeled as AD.

Figure 12 shows the feature index for all 15 features. The three features with highest feature index are: CDR(f_3), HCI (f_9) and mcSUVRcere (f_{11}).

Table 14. Confusion Matrix of Classical Latent Class Model Using All Features.

		GROUNDTRUTH			
		NC	AD	MCI	Overall accuracy
CLUSTERING	NC	92	0	88	
	AD	0	21	9	
	MCI	31	1	75	
	Total	123	22	172	
	Accuracy	74.80%	95.45%	43.60%	**59.31%**

Table 15. Confusion Matrix for Clustering After Feature Selection Using Feature Index.

		GROUNDTRUTH			
		NC	AD	MCI	Overall accuracy
CLUSTERING	NC	114	0	1	
	AD	1	22	38	
	MCI	8	0	133	
	Total	123	22	172	
	Accuracy	92.68%	100%	77.33%	**84.86%**

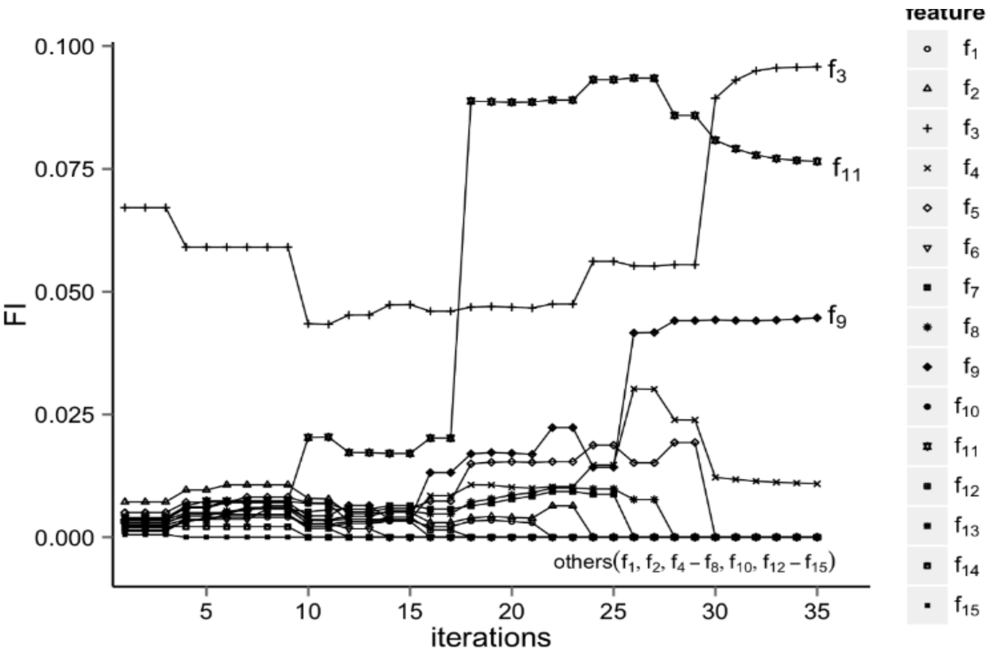


Figure 12. Feature Index for Each Feature Over Iterations on AD Data.

We want to emphasize that the results on MCI are not surprising. Clinically, MCI cohort has subtypes: MCI converter and MCI non-converter. MCI converter refers to the patient positively diagnosed as AD in the follow-up exams. Fortunately, ANDI is a rich data repository with longitudinal data available. We collect the updated patient staging information from the follow-up visit to explore the composition of the MCI group. For illustration purpose, we show a 2D plot of the two most relevant features CDR (f_3) and mcSUVRcere (f_{11}).

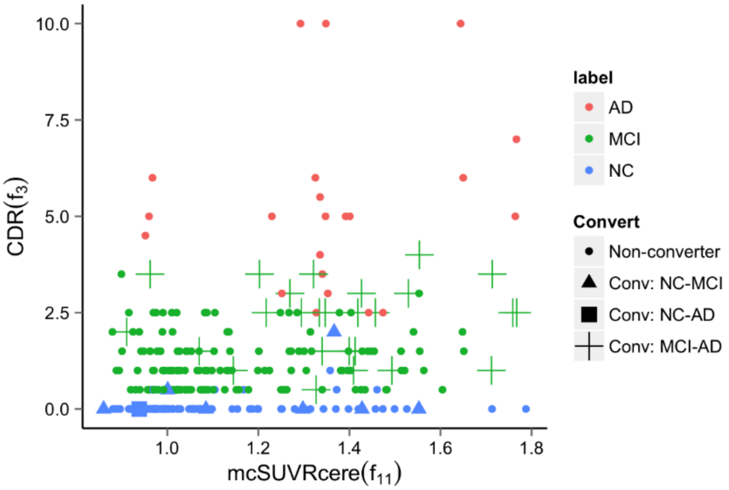


Figure 13. Clustering Shown in Two Feature Space: CDR (f_3) and mcSUVRcere (f_{11}).

The points are colored by the real diagnosis results: AD (red), MCI (green) and NC (blue). The shapes signify the converters: triangle for NC converting to MCI, square for NC converting to AD and cross for MCI converting to AD. In Figure 13, 16 out of 26 green crosses are on the boundary between MCI and AD clusters, but close to AD. That is, among the 172 MCIs, 26 are staged as AD in the follow-up visit. Using baseline data, the 17 out of 38 MCIs mislabeled as AD (Table 15) are indeed the converted. The blue triangle represents the patient converting from NC to MCI. In the 7 blue triangles, there is one special point, which is diagnosed as NC in the first visit. However, in clustering, the point is closer to MCI than NC in Figure 4 and mislabeled as MCI in Table 15, which is verified by the diagnosis of MCI in the second visit. Indeed, the clustering technique can help to capture the convert between disease types.

3.5 Conclusion

Latent Class Model, as a soft clustering methodology, has attracted great attention due to the distinct advantages from its statistical foundation. However, its performance deteriorates notably if the dataset has many noisy features irrelevant to the clustering process. This research proposes a new metric: Feature Index (*FI*). Traditional EM algorithm for Latent Class modeling parameter estimation is extended with a S step using *FI* for feature selection. Our proposed embedding the feature selection into the EM algorithm preserves the good properties of EM algorithm such as guaranteed convergence and optimum determination of the clustering number. To evaluate the performance of the proposed algorithm, experiments on one synthetic data set, one benchmark dataset and one Alzheimer's Disease (AD) dataset are conducted. The experiments on synthetic and benchmark dataset show the proposed *FI* is able to identify the relevant features and improved clustering accuracy comparing to classical Latent Class model without feature selection. Other than improved clustering result, experiment on AD indicates the model-based clustering with feature selection may potentially identify the patient subtypes which is crucial for patient treatment planning.

CHAPTER 4

UNCERTAINTY QUANTIFICATION OF FEATURE SELECTION RESULTS

4.1 Introduction

Recently, computational models and machine learning algorithms are extensively applied in the safety-critical areas such as automotive, aerospace, and structural engineering industries [11] as massive amounts of datasets collected from sensors networks, cyber-physical systems, and the Internet of Things (IoT) become available [55]. However, the data collected are inherently uncertain due to noise, incompleteness, and inconsistency[55][56]. As a result, rigorous quantification of uncertainty in the underlying data, the model, and the resulting predictions becomes critical. The interest in Uncertainty Quantification (UQ) has grown as part of a driver for rigorous and formal approaches to assess the credibility of computational models.

Compared with the conventional heuristic approaches, model-based approaches have an advantage in providing the results in a probabilistic way. By using a probabilistic model, the experimental noise can be included explicitly in the model and estimated from the data, thus more robust to noise. However, existing feature selection algorithms for model-based clustering algorithms do not consider the quantification of the feature selection results in a probabilistic manner [57]. The feature selection processes generally compare the likelihood before and after adding (or removing) a feature. The likelihood measures the "average" reaction of all data points towards the feature space change. We contend that different data points shall have different reactions to the change. Some data points do not depend on this newly added feature to choose a clustering group, while others do.

Based on our literature review, we conclude the sufficient variance information has not been taken into consideration when making feature selection in existing feature selection algorithms. In our proposed ESM algorithm (Chapter 2), the "responsibility" difference values can easily capture the variation information. In this chapter, we propose to extend our work on Relevancy Index with the variance information and develop a new ESM algorithm. We first review types of uncertainties and uncertainty quantification methods. We then present our extension to ESM algorithm incorporating uncertainty consideration followed by experiments to demonstrate the performance improvement.

4.2 Review of Uncertainty

"uncertainty is a situation which involves unknown or imperfect information" [55]. Uncertainty quantification is essential for evaluating and predicting the performance of the complex engineering systems, especially when the experimental or real-world data is not adequate or not even exist [58]. For the specific data modelling application, uncertainty can exist in every phase and from many different sources, such as data collection, model training and numerical approximation. The handling of the uncertainty in each phase has a significant impact on the learning results from the data.

4.2.1 Source of Uncertainty

There are several ways to categorize the source of uncertainty. One way is to classify uncertainty in two categories: Aleatoric uncertainty and Epistemic uncertainty [59][60]. The Aleatoric uncertainty is also known as inherent randomness. The word aleatory derives from the Latin alea, which means the rolling of dice [59]. Since the aleatory uncertainty is the intrinsic randomness, it is usually irreducible. The Epistemic uncertainty is also known as reducible uncertainty. The word epistemic derives from the Greek $\epsilon\pi l\sigma\tau\eta\mu\eta$ (episteme), which means knowledge. Thus, the epistemic uncertainty is caused by lack of knowledge or data [59]. In such scenario, the uncertainty can be reduced by enhancing knowledge or by performing measurements.

Most engineering systems involve both types of uncertainties. However, in the modeling phase, often it is difficult to determine which category a particular uncertainty falls in. For computational models, the uncertainty sources can be categorized into physical variability, data uncertainty, structural uncertainty and numerical uncertainty [58] [61].

Physical variability: This type of uncertainty is from natural or inherent random variability of physical processes and variables, thus also known as irreducible uncertainty. There is uncertainty regarding the precise values of the model inputs. Such quantities can be represented by random variables with statistical parameters such as mean and standard deviations.

Data uncertainty: It is also known as reducible uncertainty or epistemic uncertainty (knowledge or information uncertainty). The uncertainty can be reduced by collecting more information. There are at least four forms of this uncertainty: (1) sparse data, meaning the data is too small to estimate the distribution parameters; (2) interval data, meaning some variables in the data are only available as a range of values instead of a number; (3) missing data; (4) measurement error in the laboratory or in the field.

Structural uncertainty: The structural uncertainty, also known as the model form error, means the difference between the chosen model and the real system that it represents. The models may not be fully calibrated.

Numerical uncertainty: The numerical uncertainty is the solution approximation error during computing. The numerical uncertainty is introduced when using an approximation to the true solution of the equations of the mathematical model or a fast surrogate model is used when the optimal solution is very expensive to solve [61].

4.2.2 Uncertainty Quantification Problems

There are two types of uncertainty quantification problems: the forward and inverse uncertainty propagation. Forward uncertainty propagation quantifies uncertainties in the system outputs propagated from uncertain inputs. The forward uncertainty propagation focuses on the output uncertainty derived from the parametric variability in the sources of uncertainty. Inverse uncertainty quantification focuses on inferring the input parameters given measurement data through mathematical formulation. The difference between the two problems are illustrated in Figure 14 adopted from [62].

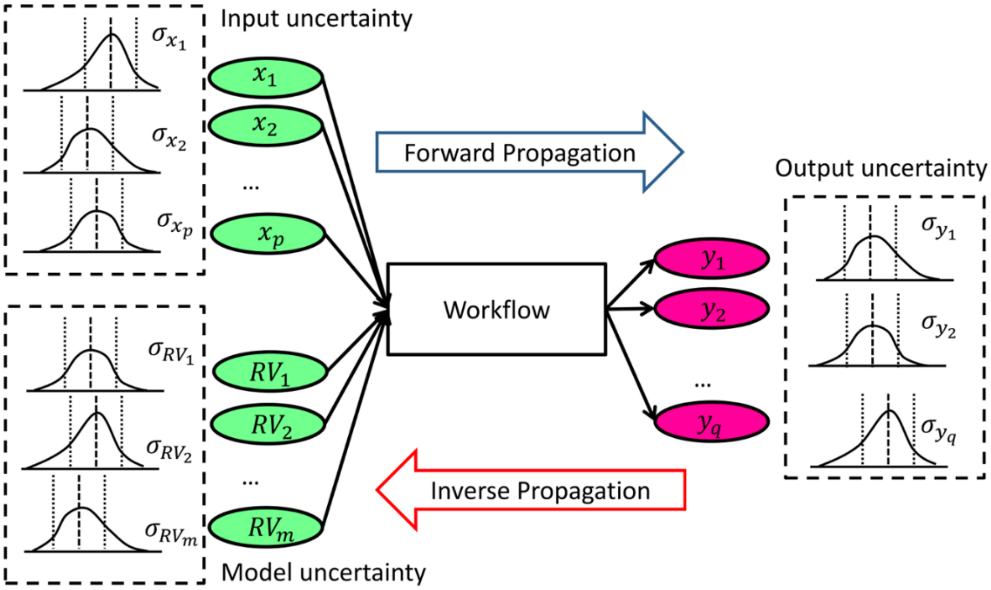


Figure 14. Illustration of Forward and Inverse Propagation Uncertainty Problems (Adopted From [62]).

The classic machine learning problem maps observed data to unobservable properties of interest, thus it is under the umbrella of the inverse uncertainty propagation problem [56]. Figure 15 illustrates the relationship between machine learning and uncertainty quantification. Since we are solving a machine learning problem, we will only consider approaches for inverse uncertainty quantification.

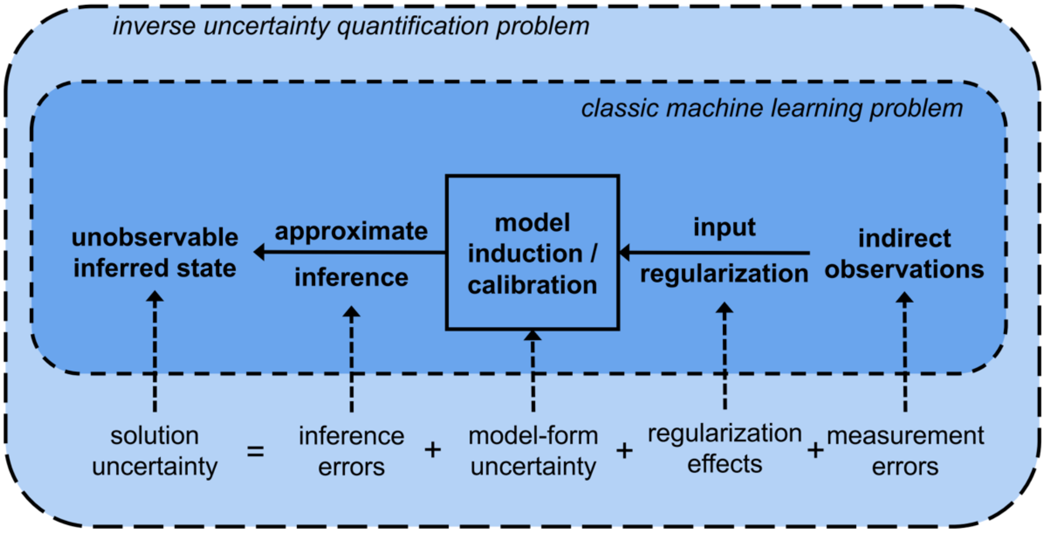


Figure 15. The Relation Between Machine Learning Problems and Uncertainty Quantification Problem (Adopted From [56]).

4.2.3 Uncertainty Quantification Method

To tackle the challenges raised from uncertainties, theories and techniques have been developed to model the uncertainties in various forms. Yet, there is currently no general definition for uncertainty that fits any situation. It is observed the uncertainties are studied often under a given context. Here since we are interested in a machine learning problem (specifically, a clustering problem), we will only consider approaches for inverse uncertainty quantification which can be categorized into probabilistic and non-probabilistic methods [63]. Probabilistic uncertainty approaches are based on rigorous probability theory under the availability of sufficient data, whereas non-probabilistic approaches are developed to cope with a lack of information or data.

4.2.3.1 Probabilistic Uncertainty Approach

Probability theory is the main tool to estimate the uncertainty. It defines the random variable to describe the random events and use stochastic processes to analyze random phenomena [64].

Probabilistic uncertainty approaches can be broadly categorized into frequentist and Bayesian approaches. In the Frequentist interpretation, probabilities represent long run frequencies of events; the true properties about parameters, which represent information about an event or system of interest, are revealed in the long run. In the Bayesian interpretation, probabilities represent the knowledge about a parameter. The knowledge without observing data is represented by a prior distribution. As more data being collected, parameters will be updated with a likelihood function. The posterior distribution represents the updated knowledge about the parameter after observing data [65]. Since the posterior is a probability distribution, it is used to quantify uncertainty about an event occurring [66], [67].

Uncertainty in probabilistic model predictions can be presented as numerical values or as visual graphs. Numerical indicators are descriptive statistical measures including mean, median, percentile, standard deviation, and quantile. Confidence Intervals (CI) express the uncertainty with the minimum information (lower and upper bound), thus are the most understandable and widely used uncertainty quantification mechanism. Typical graphical methods include histogram, density plot, cumulative distribution functions, and box plots [68].

4.2.3.2 Non-probabilistic Approach

Starting with Lotfi Zadeh's introduction of fuzzy set in 1965, alternative theories to classical probability approach have emerged for describing uncertainty. In general, for non-probabilistic approaches, there four types [69]:

The first category is around fuzziness. Fuzziness is used to measure uncertainty in classes, notably in human language [70]–[72]. Fuzzy logic then handles the uncertainty associated with human perception by creating an approximate reasoning mechanism [73], [74].

The second category is Shannon's entropy, which quantifies the amount of information in a variable, which ties to how difficult or easy it is to guess that information without looking at the variable. If it is very easy to guess the value of the variable, then the variable does not have enough "surprise" inside, and thus the variable contains less information [75]. Specifically, for a random variable x with values in a finite set X, Shannon entropy is defined as

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \geq 0$$

Shannon entropy quantifies the unevenness in the probability distribution $p(x)$. In particular, for a constant random variable with determined value, the Shannon entropy achieves minimum as zero. At the opposite extreme, for a uniform distribution, the Shannon entropy achieves the maximum [76].

The third category is classification entropy. Classification entropy measures the impurity of the class distribution [69]. Specially, for a two-class problem, if we divide the dataset S into positive or negative class, the classification entropy is defined as:

$$CE_2(P) = -[\frac{|S_+|}{|S|}\log_2\frac{|S_+|}{|S|} + \frac{|S_-|}{|S|}\log_2\frac{|S_-|}{|S|}],$$

where $|S|$ is the number of all samples in S, $|S_+|$ is the number of positive samples and $|S_-|$ is the number of negative samples. When the class distribution is pure, that is, when all the samples are in positive class or all in negative class, the classification entropy reaches the minimum value of zero. In the opposite, when the number of positive samples are equal to the number of negative samples, the entropy reaches maximum. In general, the classification entropy for a C-class problem is defined as

$$CE_c(P) = -\sum_{k=1}^{c}\frac{|S_k|}{|S|}\log_2\frac{|S_k|}{|S|},$$

where S_k is the number of samples for k^{th} class.

The last category is Rough set theory. Rough set theory provides a mathematical tool for reasoning on vague, uncertain or incomplete information. With the rough set approach, concepts are decided by two approximations (upper and lower) instead of one precise concept [77], making such methods invaluable to dealing with uncertain information system. The idea is simple, for each concept X, the greatest definable set contained X is called a lower approximation of X and least definable set containing X called is an upper approximation of X.

By now we have reviewed the general concept of uncertainty. Specifically, we have reviewed the four sources of uncertainty, two uncertainty quantification problems and both probabilistic and non-probabilistic methods for the inverse quantification problem. In the next section, we will apply the uncertainty quantification approach to the specific feature selection and clustering problem.

4.3 Uncertainty for Feature Selection and Clustering

In the clustering application, all four kinds of uncertainty listed above are involved. The first one is the inherent physical variability represented by the vector of random variables X (model inputs). The second one is data uncertainty, including inadequate data, interval data, missing data, and measurement error. The third kind of uncertainty is the model form error or structural uncertainty. For example, the Gaussian Mixture model may not be appropriate to represent the data. Fitting the data to GMM would probably fail when the data is far from the mixture of Gaussian distribution. The last one is the numerical uncertainty. The convergence at a local optimal in the EM algorithm could lead to an approximation error. Thus, the uncertainty exists both in the distribution assumption and the estimation of the parameters.

The feature selection process should include an extra layer of uncertainties from different sources. Since the feature selection procedure keeps changing the estimation of parameters in the distribution, the key metric used to select features would also change. To make the decision easier, a quantification of the selection criteria (also known as Relevancy Index in ESM) is necessary.

In this chapter, we will focus on measuring two sources of uncertainties in the feature selection process for clustering: (1) quantification of feature selection; and (2) data uncertainty.

Quantification of the feature selection criteria determines what set of features would finally be selected. Here we will conduct a validation of feature selection results through visualization. The importance of providing a quantification or validation of the selected features is in two folds. Firstly, identifying the true dominant features can help data collection more efficiently. Secondly, validation of identified insignificant features that confound the model could increase the efficiency of computing.

Another source of uncertainty we will focus on measuring is the data uncertainty. To basic idea is to detect the outliers from the clustering results. In machine learning field, we typically assume the input data are the ground truth. However, in many chances, the data are inherently incomplete and inconsistent. These outliers may due to the measurement error in the lab, leading unnecessary clusters shown in clustering results. Since the clustering results can reveal the structure of the data, it is possible that we can identify the potential outliers of original dataset recessively by visualization.

4.3 Proposed Method

As proposed in chapter 2, the ESM algorithm takes advantage of the responsibility measures in the E step. The responsibilities $\gamma(z_{nk})$ are the probability of assigning the data point x_n to cluster k. The full feature space with D features is denoted as $F = \{f_1, f_2, \dots, f_D\}$, and the feature space excluding feature j is $F_j^- = \{f_1, f_2, \dots, f_D\} \backslash \{f_j\}$. As defined in chapter 1, the responsibility on the full feature space is $\gamma^F(z_{nk})$ and the responsibility on the reduced feature space (excluding feature j) is $\gamma^{F_j^-}(z_{nk})$. From the definition, the responsibility is related to the n^{th} data point and the k^{th} cluster. The Relevancy index (*RI*) used in ESM algorithm is defined as the difference between two responsibilities averaged over N data points and K clustering groups to evaluate the importance of j^{th} feature to the clustering. It is written as:

$$RI(j) = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K} |\gamma^{F}(z_{nk}) - \gamma^{F_j^-}(z_{nk})| \tag{4.1}$$

From the formula, the Relevancy Index only handles the average of all the differences, while all the remaining information were ignored. Specifically, if we denote the responsibility differences $|\gamma^{F}(z_{nk}) - \gamma^{F_j^-}(z_{nk})|$ as $\delta_j(z_{nk})$, the difference on instance x_n caused by removing feature j. Then the difference values shall have a distribution over the instances. The variance information of the difference values can not only help to evaluate the confidence of excluding a feature, but also provides some clues to detect outlier instances.

Based on this idea, an upgraded ESM algorithm called Expectation-Selection-Outlier-Maximization (ESOM) is proposed, which takes variance information into account in S step for feature selection. In addition, some candidate outliers are detected as a side product in the O step. Table 16 summarizes the ESOM algorithm.

Table 16. ESOM Algorithm Pseudo Code.

```
1. Initialize the means $\mu_k$, covariances $\Sigma_k$ and mixing coefficients $\pi_k$, and evaluate the initial
   value of the log likelihood.
2. E step. Evaluate the responsibilities using the current parameter values
       $\gamma^{F}(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k)}$
   and responsibilities after excluding each feature
       $\gamma^{F_j^-}(z_{nk}) = \frac{\pi_k N(x_n^*|\mu_k^*, \Sigma_k^*)}{\sum_{k=1}^{K} \pi_k N(x_n^*|\mu_k^*, \Sigma_k^*)}\ for\ j = 1,2,..D$
```

where x_n^*, μ_k^* and Σ_k^* are the corresponding vector of x_n , μ_k and Σ_k after excluding j_{th} variable.

3. **S step.** Calculate the differences between responsibilities before and after excluding j_{th} feature at iteration t.

$$\delta(j,n,k)^{(t)} = \left|\gamma^F(z_{nk}) - \gamma^{F_j^-}(z_{nk})\right| \quad for\ n = 1,2,\dots N\ and\ k = 1,2,..K.$$

For each j, calculate the mean and variance of the differences $\delta(j)^{(t)}$ with respect to n and k.

$$\mu(j)^{(t)} = \frac{1}{NK}\sum_{n,k}\delta(j,n,k)^{(t)}$$

$$\sigma(j)^{(t)} = \sqrt{\frac{\sum_{n,k}(\delta(j,n,k)^{(t)} - \mu(j)^{(t)})^2}{NK-1}}$$

If $\left|\mu(j)^{(t+1)} - \mu(j)^{(t)}\right| < \epsilon_1$ (converged) and $\mu(j)^{(t)} + \sigma(j)^{(t)} < \epsilon_2$, then discard the feature with smallest $\mu(j)^{(t)}$ and update the full feature space *F*.

4. **O step.** Keep a record of the instances with largest differences in each iteration and save them into set *O*.

5. **M step.** For reduced data with feature space *F*, re-estimate the parameters using the current responsibilities

$$\mu_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})x_n$$

$$\Sigma_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})'$$

$$\pi_k^{new} = \frac{N_k}{N}$$

6. Evaluate the log likelihood

$$ln\ P(X|\pi,\mu,\Sigma) = \sum_{n=1}^{N} ln\left\{\sum_{k=1}^{K}\pi_k N(x_n|\mu_k,\Sigma_k)\right\}$$

If the parameters or the log likelihood are not converged, go back to step 2.

Illustration Example

To get a basic idea of the responsibility difference values, Table 17 summarizes the mean, standard deviation, minimum and maximum value of the vector $\delta(j,n,k)$ at a certain iteration on ten features. The data is a synthetic dataset with 300 data points and 10 features.

The first two features were the true important features. The detailed experiment settings were in Chapter 2 Table 2.

As shown in the table, the first two features have the highest mean of responsibility difference $\delta(j, n, k)$. For the remaining features, the mean values are close to zero. Even though the fourth feature has a mean value as 0.006, the maximum is 0.388, meaning that some data points rely a lot on that feature. That data point could be an outlier or a key data point that has to be clustered by this feature's information.

Table 17. Delta Values Illustration.

feature	mean	Standard deviation	minimum	maximum
1	0.177502	0.228454	2.55E-05	0.911053
2	0.043622	0.098034	5.24E-12	0.567292
3	0.004453	0.017775	0.00E+00	0.197878
4	0.006441	0.031047	0.00E+00	0.387684
5	0.006979	0.034216	0.00E+00	0.382952
6	0.00443	0.018156	0.00E+00	0.185167
7	0.004065	0.017814	0.00E+00	0.173646
8	0.005816	0.028182	0.00E+00	0.360028
9	0.007329	0.025009	0.00E+00	0.189193
10	0.002365	0.008477	0.00E+00	0.077799

4.4 Experiments

In this section, we first test the clustering performance of the ESOM algorithm on several benchmark datasets in comparison with the original ESM algorithm and other feature selection algorithms. Then the selected features are evaluated through visualization. Moreover, we examine detected outliers by comparing them with the misclassified data points.

Datasets: In order to show that ESOM works well with various datasets, we test the algorithm on both synthetic dataset and three real-world datasets.

Synthetic Dataset: A synthetic dataset having 300 instances and 10 features with f_1 and f_2 being the relevant features for clustering and other eight features are irrelevant features. The relevant features are simulated from two-component mixture of Gaussian distributions with equal number of data points for each component. The irrelevant features are randomly generated from normal distributions. The detailed experiment settings are in Table 2 at Chapter 2.

Vowel: An empirical dataset with 990 rows and 9 context-sensitive features. The problem is to recognize a vowel spoken by an arbitrary speaker. The observations fall in eleven groups (different vowels) [35].

Wine: A chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The wine data contains three types of wines with 178 rows and 27 columns.

Crab: An empirical dataset with 200 rows and 8 columns, among which five columns describe the morphological measurements while the remaining three columns are the color (orange and blue), sex (Female and Male) and index that divide the 200 crabs evenly into four groups by the combination of color and sex.

Clustering Metrics: To evaluate the cluster quality, two evaluation metrics, accuracy (ACC) and adjusted rand index (ARI) were computed. The best mapping between cluster assignments and true labels is computed using the Hungarian algorithm [78]. In addition, we also report the running time.

Alternative Models: We compare the proposed algorithm with other existing algorithms. To get a fair comparison, we focus on the model-based feature selection algorithms. The survey paper [27] summarized six GMMs variable selection R packages named as sparcl[28], clustvarsel [29], VarSelLCM [30], vscc [31], SelvarMix [32], bclust [33]. Among the six methods, 'sparcl' mainly performs sparse hierarchical and sparse K-means clustering; the 'bclust' package requires a deliciated initial transformed model. Thus, we exclude these two methods and compare the proposed method with the other four methods.

4.4.1 Clustering Performance Comparison

The results are summarized in Table 18. From the results of the experiments, we found that, on this set of benchmark datasets, the ESOM algorithm outperforms the original ESM method and has comparable accuracy with other algorithms.

Table 18. Performance Results on Benchmark Datasets.

DATASET	ALGORITHM	ACC	ARI	TIME(SEC)
SYNTHETHIC	clustvarsel	0.977	0.909	0.929
	VarSelCluster	0.973	0.896	1.656
	vscc	0.970	0.883	**0.332**
	selvarclustLasso	0.977	0.909	0.450

	ESM	0.977	0.909	0.617
	ESOM	**0.977**	**0.909**	0.694
VOWEL	clustvarsel	0.302	0.118	143.445
	VarSelCluster	0.377	**0.211**	64.623
	vscc	0.371	0.181	7.979
	selvarclustLasso	0.339	0.151	16.094
	ESM	0.358	0.169	**3.973**
	ESOM	**0.384**	0.204	26.404
WINE	clustvarsel	0.910	0.739	13.910
	VarSelCluster	0.944	0.830	4.086
	vscc	**0.978**	**0.931**	8.316
	selvarclustLasso	0.843	0.585	**2.380**
	ESM	0.916	0.755	2.953
	ESOM	0.955	0.864	6.707
CRAB	clustvarsel	**0.935**	**0.840**	1.982
	VarSelCluster	0.355	0.038	1.723
	vscc	0.620	0.428	**0.196**
	selvarclustLasso	0.595	0.354	0.914
	ESM	0.910	0.783	1.255
	ESOM	0.910	0.783	1.771

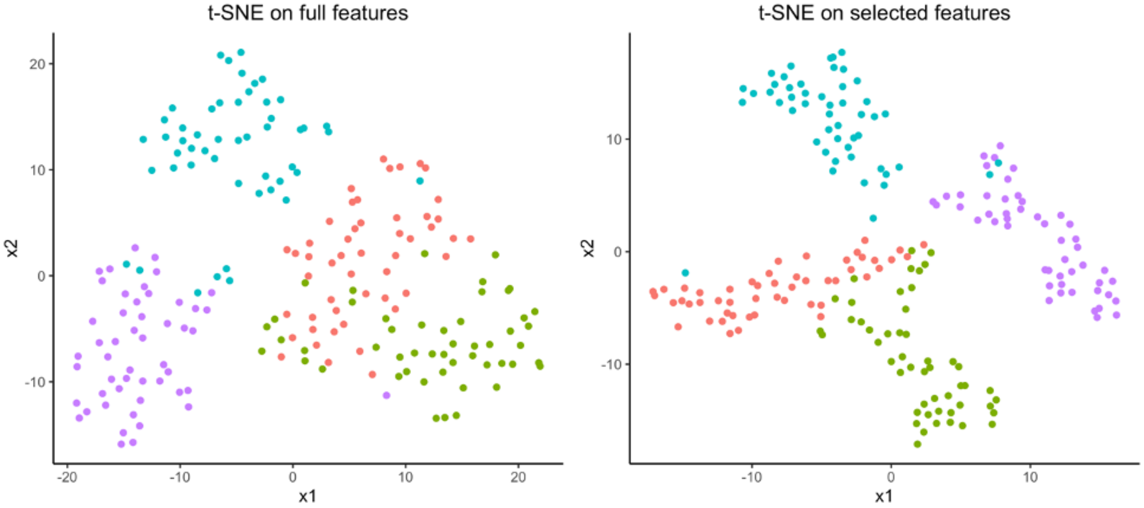
Another observation is that the clustering performance depends a lot on the dataset. One algorithm can perform well one dataset while badly on the other. For example, ‘VarSelCluster shows the best ARI (second-best accuracy) on ‘Vowel’ dataset while worst on the ‘Crab’ dataset. ‘vscc’ shows the best accuracy on ‘Wine’ dataset but wicked accuracy on ‘Crab’ dataset. Nevertheless, the ESOM algorithm has competitive accuracy on all benchmark datasets here. The facts show the potential robustness of the ESOM algorithm on different datasets.

4.4.2 Evaluation of Selected Features

For simulated datasets with known selected features, it is easy to check the feature selection results by comparing the selected features with the ground truth. For example, in the two synthetic datasets above, we can verify that the ESOM algorithm selected the correct features. While for a real-world dataset, there is no knowledge about which features are correct and which features are redundant. One way to check whether the features are valid is through visualization.

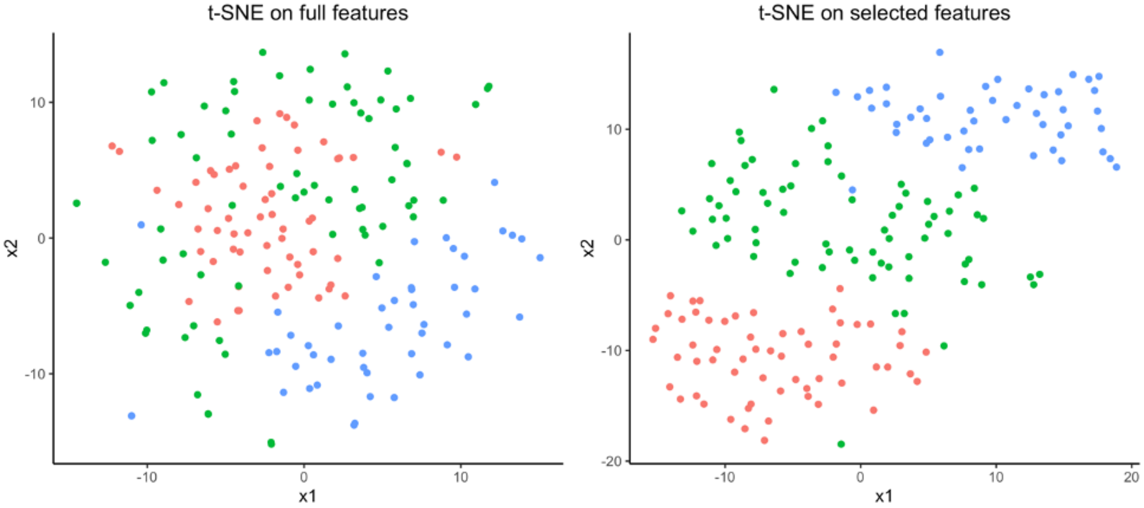
To visualize the dataset, mapping the original dimension to two-dimensional space is necessary. Here we use the t-Distributed Stochastic Neighbor Embedding (t-SNE) method for the visualization. The t-SNE is a dimension reduction technique that maps the original feature space to two or three-dimensional space for visualization [79].

The following plot, Figure 16, compares the t-SNE plot of the Crab dataset mapped from full feature space and selected feature space. Precisely, the left plot maps from the original feature space, which includes all the features in the dataset. The right one maps from the selected feature space by the ESOM algorithm. The color represents the class or label information. In the left plot, the green and red classes are mixed and hard to separate. In the right one, the classes are well separated, except for a few data points. The observations also hold for the Wine dataset, as shown in Figure 17. From the comparison, we conclude that excluding some noisy features could help to cluster.


t-SNE on full features

t-SNE on selected features

Figure 16. Compare Full Feature Versus Selected Features on Crab Dataset.


t-SNE on full features

t-SNE on selected features

Figure 17. Compare Full Feature Versus Selected Features on Wine Dataset.

4.4.3 Examine Detected Outliers

One way to examine the detected outliers is to compare them with misclassified data points. Note that clustering is unsupervised learning. The models were trained without any label information, and thus the outliers were detected without labels. The outliers are either boundary points or probably misclassified. When we know the ground truth of labels, we can evaluate the performance of outliers by comparing them with misclassified data points. If the overlap is significant, then we can conclude that the outlier detection is robust.

Figure 18 compares the detected outliers versus misclassified data points on the 'Wine' dataset. The dots with bigger sizes are the detected outliers in the left plot and misclassified in the right plot. In this wine dataset, there are 176 samples in total, of which eight instances are misclassified after applying the ESOM algorithm with accuracy 95.5%. From the 'Misclassified' visualization plot, all of the misclassified samples belong to green class but clustered either in red or blue. The left plot shows the detected outliers from the ESOM algorithm. Of the eight detected outliers, four of them (No.62, 84, 95, and 99) are actually misclassified. No. 60 and No.96 are on the margin and may form a new cluster. The remaining two data points No.75 and No.82 are on the boundary between the green and red class. The three forms of "outliers" do make sense on the visualization plot based on full feature space.

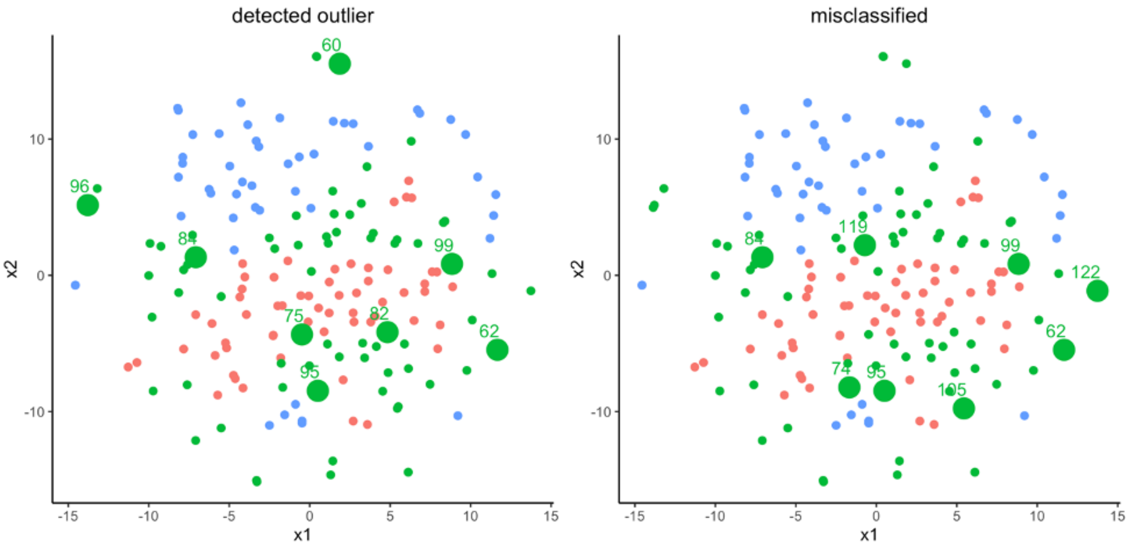


Figure 18. Compare Detected Outliers with Misclassified Data Points on Full Feature Space.

4.5 Conclusion

In this chapter, we proposed an improved version of the ESM algorithm, which takes the variance information of the feature selection criteria into consideration while conducting the feature selection for clustering. Specifically, the proposed ESOM algorithm evaluates the distribution of the feature selection criteria, the responsibility difference values on instance x_n caused by removing feature j, to quantify the confidence of selecting a certain feature. Besides, the variance information is also used to detect outliers to quantify data uncertainty. To evaluate the performance of the ESOM algorithm, we conducted experiments on four benchmark datasets. The experiments show that (1) the selected features are promising as they form a new space that's easier to cluster, compared with the original feature space; (2) the improved ESOM algorithm improves the clusteirng accuracy;

(3) The new algorithm can also detect candidate outliers which have a big overlap with misclassified instances. The promisng results show the potential of applying uncertainty quantification to the general unsupervised feature selection problem as future work.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Mixture model-based clustering approaches, as a soft assignment clustering methodology, have demonstrated their superior performance in many fields since most real-world problems are uncertain by nature. However, there still exist some challenges when applying mixture models on noisy datasets. The inclusion of the redundant features will confuse the model to identify the true structure of the clusters. To address the issue, I propose feature selection as a solution in this dissertation.

In the first topic, a novel feature selection algorithm termed ESM is proposed based on the EM algorithm for Gaussian mixture model, which can handle the continuous dataset. Specifically, the traditional EM algorithm for GMM modeling parameter estimation is extended with an S step using *RI* for feature selection, where *RI* is the relevancy index to measure the importance of each feature. The ESM algorithm preserves the good properties of the EM algorithm, such as guaranteed convergence and optimum determination of the clustering number. The experiments on synthetic datasets show that ESM can identify the relevant features and improved clustering accuracy comparing to EM. The experiments on four benchmark datasets show that ESM has a competitive performance on accuracy and running time compared with existing algorithms. Other than improved clustering results, the experiment on AD indicates that ESM may potentially identify the patient subtypes, which is crucial for patient treatment planning.

The second topic extends the feature selection work from continuous only datasets to mixed type datasets. To achieve that, the mixture model is extended from GMM to Latent Class Model. A new metric termed Feature Index (*FI*) is introduced to measure the importance of each feature. The extended algorithm is evaluated on one synthetic data set, one benchmark dataset and the Alzheimer's Disease (AD) dataset with categorical variables included. The experiments show that the proposed *FI* is able to identify the relevant features and improved clustering accuracy comparing to classical Latent Class model without feature selection.

The third topic proposes an improved version of the original ESM algorithm termed ESOM, which aims to quantify the uncertainty of the feature selection results. The ESOM algorithm takes the variance information of the feature selection criteria into consideration while conducting the feature selection for clustering. The variance information is also used to detect outliers to quantify the input data uncertainty. The experiments on four benchmark datasets show that the selected features can form a new space that's easier to cluster compared with the original feature space. Also, the improved ESOM algorithm improves the clusteirng accuracy. Finally, the new algorithm can detect candidate outliers which have a big overlap with misclassified instances.

For the future work, I would like to consider an extension of the feature selection approach to higher dimensional datasets. The current ESM algorithm is still limited in handling real high dimensional dataset due to the exponentially increased parameters in the mixture model. More work could be done to speed up the algorithm and increase the robustness for datasets without a mixture property. Also, the uncertainty quantification idea can be extended to more machine learning models to test the robustness of the model in the wild as future work.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[2] J. Yu, “Fault detection using principal components-based gaussian mixture model for semiconductor manufacturing processes,” *IEEE Trans. Semicond. Manuf.*, vol. 24, no. 3, pp. 432–444, 2011.

[3] B. Mazoyer *et al.*, “Gaussian Mixture Modeling of Hemispheric Lateralization for Language in a Large Sample of Healthy Individuals Balanced for Handedness,” *PLoS One*, vol. 9, no. 6, p. e101165, Jun. 2014.

[4] A. Lindemann, C. L. Dunis, and P. Lisboa, “Probability distributions, trading strategies and leverage: An application of Gaussian mixture models,” *J. Forecast.*, vol. 23, no. 8, pp. 559–585, 2004.

[5] K. J. Lee, L. Guillemot, Y. L. Yue, M. Kramer, and D. J. Champion, “Application of the Gaussian mixture model in pulsar astronomy - pulsar classification and candidates ranking for the Fermi 2FGL catalogue,” *Mon. Not. R. Astron. Soc.*, vol. 424, no. 4, pp. 2832–2840, 2012.

[6] M. Steinbach, G. Karypis, and V. Kumar, “A Comparison of Document Clustering Techniques,” *KDD Work. text Min.*, vol. 400, pp. 1–2, 2000.

[7] S. Ben-David, “A framework for statistical clustering with constant time approximation algorithms for K-median and K-means clustering,” *Mach. Learn.*, vol. 66, no. 2–3, pp. 243–257, Mar. 2007.

[8] C. Fraley and a E. Raftery, “How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis,” *Comput. J.*, vol. 41, no. 8, pp. 578–588, 1998.

[9] D. B. Dahl, “Model-Based Clustering for Expression Data via a Dirichlet Process Mixture Model,” *Bayesian inference gene Expr. proteomics*, pp. 201–218, 2006.

[10] I. Morlini, “A latent variables approach for clustering mixed binary and continuous variables within a Gaussian mixture model,” *Adv. Data Anal. Classif.*, vol. 6, no. 1, pp. 5–28, 2012.

[11] G. C. & G. Govaert, “A classification EM algorithm for clustering and two stochastic versions,” vol. 14, 1992.

[12] B. Thiesson, C. Meek, and D. Heckerman, “Accelerating EM for large databases,”

Mach. Learn., vol. 45, no. 3, pp. 279–299, 2001.

[13] C. Maugis, G. Celeux, and M. L. Martin-Magniette, “Variable Selection for Clustering with Gaussian Mixture Models,” *Biometrics*, vol. 65, no. 3, pp. 701–709, 2009.

[14] H. Permuter, J. Francos, and I. Jermyn, “A study of Gaussian mixture models of color and texture features for image classification and segmentation,” *Pattern Recognit.*, vol. 39, no. 4, pp. 695–706, 2006.

[15] D. a. Reynolds, “Automatic speaker recognition using Gaussian mixture speaker models,” *Lincoln Lab. J.*, vol. 8, no. 2, pp. 173–192, 1995.

[16] S. Adams and P. A. Beling, “A survey of feature selection methods for Gaussian mixture models and hidden Markov models,” *Artif. Intell. Rev.*, pp. 1–41, 2017.

[17] S. Krishan, K. Samudravijaya, and P. V. S. Rao, “Feature Selection for Pattern Recognition with Gaussian Mixture Models: A New Objective Criterion,” *Pattern Recognit. Lett.*, vol. 17, no. 7, pp. 803–809, Jul. 1996.

[18] A. E. Raftery and N. Dean, “Variable Selection for Model-Based Clustering,” *J. Am. Stat. Assoc.*, vol. 101, no. 473, pp. 168–178, 2006.

[19] L. Scrucca, “Genetic algorithms for subset selection in model-based clustering,” in *Unsupervised Learning Algorithms*, Cham: Springer International Publishing, 2016, pp. 55–70.

[20] W. Pan and X. Shen, “Penalized Model-Based Clustering with Application to Variable Selection,” *J. Mach. Learn. Res.*, vol. 8, pp. 1145–1164, 2007.

[21] S. Wang and J. Zhu, “Variable Selection for Model-Based High-Dimensional Clustering and Its Application to Microarray Data,” *Biometrics*, vol. 64, no. 2, pp. 440–448, Jun. 2008.

[22] B. Xie, W. Pan, and X. Shen, “Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables,” *Electron. J. Stat.*, vol. 2, pp. 168–212, 2008.

[23] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, “Simultaneous feature selection and clustering using mixture models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1154–1166, 2004.

[24] P. Carbonetto, N. de Freitas, P. Gustafson, and N. Thompson, “Bayesian Feature Weighting for Unsupervised Learning with Application to Object Recognition,”

Society for Artificial Intelligence and Statistics, Jan. 2002.

[25] C. M. Bishop and N. Nasrabadi, “Pattern Recognition and Machine Learning,” *Pattern Recognit.*, vol. 4, no. 4, p. 738, 2006.

[26] P. D. Hoff, “Model-based subspace clustering,” *Bayesian Anal.*, vol. 1, no. 2, pp. 321–344, 2006.

[27] M. Fop and T. B. Murphy, “Variable Selection Methods for Model-based Clustering,” *Stat. Surv.*, vol. 12, no. 0, pp. 18–65, 2017.

[28] A. M. Daniela Witten and R. Tibshirani Maintainer Daniela Witten, “Package ‘sparcl’ Type Package Title Perform Sparse Hierarchical Clustering and Sparse K-Means Clustering,” 2018.

[29] L. Scrucca and A. E. Raftery, “**clustvarsel**: A Package Implementing Variable Selection for Gaussian Model-Based Clustering in *R*,” *J. Stat. Softw.*, vol. 84, no. 1, pp. 1–28, Apr. 2018.

[30] M. Marbac, M. S. Maintainer, and M. Sedki, “Package ‘VarSelLCM’ Title Variable Selection for Model-Based Clustering of Mixed-Type Data Set with Missing Values,” 2018.

[31] J. L. Andrews and P. D. Mcnicholas, “Package ‘vscc’ Title Variable selection for clustering and classification,” 2015.

[32] M. Sedki, G. Celeux, C. Maugis, and R. Maintainer, “Package ‘SelvarMix’ Type Package Title Regularization for Variable Selection in Model-Based Clustering and Discriminant Analysis,” 2017.

[33] P. Nia and A. C. Davison, “Package ‘bclust’ Type Package Title Bayesian Hierarchical Clustering Using Spike and Slab Models,” 2015.

[34] P. Fränti, R. Mariescu-Istodor, and C. Zhong, “XNN graph,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 10029 LNCS, pp. 207–217.

[35] P. D. Turney, “Exploiting context when leaming to classify,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1993, vol. 667 LNAI, pp. 402–407.

[36] “2008 Alzheimer’s disease facts and figures,” *Alzheimer’s and Dementia*, Mar-2008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1552526008000332. [Accessed: 18-

Aug-2017].

[37] S. Gauthier *et al.*, “Mild cognitive impairment,” *Lancet*, vol. 367, no. 9518. pp. 1262–1270, 2006.

[38] M. Castro and G. E. Smith, “Mild cognitive impairment and Alzheimer’s disease.,” in *APA handbook of clinical geropsychology, Vol. 2: Assessment, treatment, and issues of later life.*, Washington: American Psychological Association, 2015, pp. 173–207.

[39] Michael W. Weiner MD, “ADNI Alzheimer’s Disease Neuroimaging Initiative,” 2013. [Online]. Available: http://www.adni-info.org/. [Accessed: 18-Aug-2017].

[40] C. Hennig and T. F. Liao, “How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification,” *J. R. Stat. Soc. Ser. C Appl. Stat.*, vol. 62, no. 3, pp. 309–369, 2013.

[41] A. Foss, M. Markatou, B. Ray, and A. Heching, “A semiparametric method for clustering mixed data,” *Mach. Learn.*, vol. 105, no. 3, pp. 419–458, Dec. 2016.

[42] A. Ahmad and L. Dey, “A k-mean clustering algorithm for mixed numeric and categorical data,” *Data Knowl. Eng.*, vol. 63, no. 2, pp. 503–527, 2007.

[43] J. C. Gower, “A General Coefficient of Similarity and Some of Its Properties,” *Biometrics*, vol. 27, no. 4, pp. 857–871, 1971.

[44] Z. He, X. Xu, and S. Deng, “Clustering Mixed Numeric and Categorical Data: A Cluster Ensemble Approach,” p. 14, 2005.

[45] I. Moustaki and I. Papageorgiou, “Latent class models for mixed variables with applications in Archaeometry,” *Comput. Stat. Data Anal.*, vol. 48, no. 3, pp. 659–675, 2005.

[46] C. J. Lawrence and W. J. Krzanowski, “Mixture separation for mixed-mode data,” *Stat. Comput.*, vol. 6, no. 1, pp. 85–92, Mar. 1996.

[47] L. A. H. Murray A. Jorgensen, “Mixture Model Clustering of Data Sets with Categorical and Continuous Variables,” pp. 278–283, 1996.

[48] G. J. McLachlan, R. W. Bean, and D. Peel, “A mixture model-based approach to the clustering of microarray expression data,” *Bioinformatics*, vol. 18, no. 3, pp. 413–422, Mar. 2002.

[49] R. P. Browne and P. D. McNicholas, “Model-based clustering, classification, and

discriminant analysis of data with mixed type," *J. Stat. Plan. Inference*, vol. 142, no. 11, pp. 2976–2984, 2012.

[50] L. Hunt and M. Jorgensen, "Clustering mixed data," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 4, pp. 352–361, 2011.

[51] W. J. Krzanowski, "The location model for mixtures of categorical and continuous variables," *J. Classif.*, vol. 10, no. 1, pp. 25–49, Jan. 1993.

[52] J. M. Joyce, "Kullback-Leibler Divergence," in *International Encyclopedia of Statistical Science*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 720–722.

[53] D. Lam, M. Wei, and D. Wunsch, "Clustering Data of Mixed Categorical and Numerical Type With Unsupervised Feature Learning," *IEEE Access*, vol. 3, pp. 1605–1616, 2015.

[54] A. Association, "2008 Alzheimer's disease facts and figures," *Alzheimer's Dement.*, vol. 4, no. 2, pp. 110–133, 2008.

[55] R. H. Hariri, E. M. Fredericks, and K. M. Bowers, "Uncertainty in big data analytics: survey, opportunities, and challenges," *J. Big Data*, vol. 6, no. 1, 2019.

[56] D. J. Stracuzzi, M. C. Darling, M. G. Peterson, and M. G. Chen, "Quantifying Uncertainty to Improve Decision Making in Machine Learning; Quantifying Uncertainty to Improve Decision Making in Machine Learning," 2018.

[57] X. Liu, K. K. Lin, B. Andersen, and M. Rattray, "Including probe-level uncertainty in model-based gene expression clustering," *BMC Bioinformatics*, vol. 8, 2007.

[58] K. Hayes, "Uncertainty and uncertainty analysis methods," 2011.

[59] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?," *Struct. Saf.*, vol. 31, no. 2, pp. 105–112, 2009.

[60] H. M. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural Network-Based Uncertainty Quantification: A Survey of Methodologies and Applications," *IEEE Access*, vol. 6. Institute of Electrical and Electronics Engineers Inc., pp. 36218–36234, 03-Jun-2018.

[61] R. H. Johnstone *et al.*, "Uncertainty and variability in models of the cardiac action potential: Can we build trustworthy models?," *Journal of Molecular and Cellular Cardiology*, vol. 96. pp. 49–62, 2016.

[62] S. H. Lee and W. Chen, “A comparative study of uncertainty propagation methods for black-box-type problems,” *Structural and Multidisciplinary Optimization*, vol. 37, no. 3. pp. 239–253, 2009.

[63] A. Litvinenko and H. G. Matthies, “Inverse problems and uncertainty quantification,” Dec. 2013.

[64] C. G. Morgan, “Many valued probability theory,” *Proc. Int. Symp. Mult. Log.*, pp. 294–299, 2004.

[65] X. Chen, A. Molina-Cristóbal, M. D. Guenov, V. C. Datta, and A. Riaz, “A Novel Method for Inverse Uncertainty Propagation,” in *Computational Methods in Applied Sciences*, vol. 48, 2019, pp. 353–370.

[66] J. Vejnarová and V. Kratochvíl, “Belief Functions: Theory and Applications,” 2016.

[67] R. Willink and R. White, “Disentangling Classical and Bayesian Approaches to Uncertainty Analysis,” pp. 1–19, 2011.

[68] W. Tian *et al.*, “A review of uncertainty analysis in building energy assessment,” *Renewable and Sustainable Energy Reviews*, vol. 93. pp. 285–301, 2018.

[69] X. Wang and Y. He, “Learning from Uncertainty for Big Data,” *Ieee Syst. Man Cybern. Mag.*, no. August, 2016.

[70] B. R. Gaines, “Fuzzy and probability uncertainty logics,” *Inf. Control*, vol. 38, no. 2, pp. 154–169, 1978.

[71] F. Qi, “Fuzziness vs. probability in a data mining application for soil classification,” *Proc. - 2010 7th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2010*, vol. 6, no. Fskd, pp. 2614–2618, 2010.

[72] Q. Hu, D. Yu, Z. Xie, and J. Liu, “Fuzzy probabilistic approximation spaces and their information measures,” *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 2, pp. 191–201, 2006.

[73] L. A. Zadeh, “Toward a generalized theory of uncertainty (GTU)- An outline,” *Inf. Sci. (Ny).*, vol. 172, no. 1–2, pp. 1–40, 2005.

[74] L. A. Zadeh, “Toward a perception-based theory of probabilistic reasoning with imprecise probabilities,” in *Intelligent Systems for Information Processing: From Representation to Applications*, vol. 105, 2003, pp. 3–34.

[75] S. Vajapeyam, “Understanding Shannon’s Entropy metric for Information,” 2014.

[76] A. Lesne, “Shannon entropy: A rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics,” *Math. Struct. Comput. Sci.*, vol. 24, no. 3, 2014.

[77] L. Pawlak, L. Grzvmala-Busse, R. Slowinski, and W. Ziarko, “Rough Sets,” *Commun. ACM*, vol. 38, no. 11, pp. 88–95, 1995.

[78] R. Jonker and T. Volgenant, “Improving the Hungarian assignment algorithm,” *Oper. Res. Lett.*, vol. 5, no. 4, pp. 171–175, 1986.

[79] D. Graham-Rowe, “Visualizing Data using t-SNE,” *New Sci.*, vol. 164, no. 2210, p. 10, 2008.