Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Bsc2 Econometrics and Economics

Interpreting Cluster Analysis via Prototype Optimisation with Gaussian Mixture Model

Miyu Todani (558023)

Supervisor:	Willemsen, RSH
Second assessor:	Badenbroek, RM
Date final version:	1st July 2024

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Abstract

Assuring high interpretability is important in the validation of cluster analysis. Two models have been previously introduced: a set covering and a partitioning model. These models output optimal prototypes and corresponding true and false positive rates as their interpretability. In addition, the Gaussian Mixture Model is introduced to relax the assumption of data having circular clusters. It accounts for mean and covariance and allows elliptical clusters under normal distribution. Both real-life data and simulated data are applied. The results in the real-life data show that the Gaussian Mixture Model has worse interpretability compared to the set covering model and the partitioning model, possibly due to the non-normally distributed data. However, the simulated data presents that the set covering model has the least interpretability compared to the partitioning model because its interpretability is affected by the covariance of the data. The Gaussian Mixture Model showed greater interpretability than the set covering but does not fully outperform the partitioning model.

1 Introduction

Cluster analysis summarises data into groups of subpopulations that have similar attributes. There are many applications of cluster analysis in real life. For example, when businesses have a large instance of customer information, cluster analysis can segment customers so that similar customers are grouped. This helps them to have clearer targets and marketing goals for their businesses. Another example is clustering biological patterns such as genetics (Liu et al., 2022) (Shi & Huang, 2017). Cluster analysis helps identify subpopulations within a species, which can be crucial for understanding genetic diversity and evolutionary relationships. Other applications include clustering texts (Dransfield et al., 2004), identifying hydrogeological features to aid groundwater interpretation (Ashley & Lloyd, 1978), grouping finance assets (Gibert & Conti, 2014), and analysis on data security (Corral et al., 2009).

Interpretability, which is the ability to explain the cluster, determines whether the observed groupings are accurate. Commonly, there are two ways of interpreting clustering, namely, intrinsic and post-hoc models. Intrinsic models build explanations and clustering simultaneously (Zeng et al., 2011), while the post-hoc approach starts with clustering, and then identifies the explanation (Dronov & Evdokimov, 2018). This thesis focuses on the post-hoc approach, in which the comprehension of each cluster can cause difficulty. One way of interpreting post-hoc clustered models is via prototypes, a set of data points summarising a cluster's characteristics. Hence, selecting prototypes that achieve high interpretability are meaningful additions to the cluster analysis.

In this thesis, two models are studied as a basis: the set covering model and the partitioning model. The formulation of the set covering is based on the Location Analysis problems such as by García & Marín (2019), while the formulation of the partitioning model is inspired by Marín & Pelegrín (2019)'s p-median problems. These two models output prototypes, allowing us to determine the interpretability of a dataset. The clustering is established on the Euclidean distance. Therefore, the closeness of the data points is a measure to find prototypes. The difference between the models lies in the way it groups data points. The set covering model outputs a radius of a circle with a prototype being the centre. If it is within the circle region, the data point belongs to a cluster. Meanwhile, the partitioning model assigns a data point to a cluster if it is the closest prototype available.

However, one of the limitations of the two models is that these only take the average distance into account, as well as assume data as circular clusters when finding prototypes per cluster Ikotun et al. (2023). To overcome this, we introduce the Gaussian Mixture Model (GMM). GMM fits data into probabilistic distribution and finds clusters. GMM is usually performed in a normal distribution. This model includes the expectation-maximisation algorithm (Patel & Kushwaha, 2020), using both the mean and the covariance, as well as detecting ellipsoidal-shaped clusters based on maximum probability density estimations. These parameters of GMM can provide direct information on mean, covariance and weights that define cluster characteristics. GMM outperforms k-means clustering, a popular clustering technique, in complex data as well as soft clustering, when clusters overlap. An example of when it has better performance is clustering in cloud workloads (Patel & Kushwaha, 2020) and high-speed machining (Z. Wang et al., 2019). GMM is often applied to cluster bimodal patterns and environmental factors. For example, Liu et al. (2022) developed a new GMM to maximise information extracted from gene-expression clustering. Optimising battery storage prototypes to improve electrical resilience using GMM is researched by Huang & Gou (2024).

To contribute to the research of prototype optimisation for cluster analysis, the goal is to have greater accuracy in selecting descriptive prototypes for explaining the clusters. While general p-median clustering is based on uniform shapes and focuses on minimising the distance, GMM allows greater flexibility in capturing the elliptical and complex cluster shapes. Hence, this thesis attempts to recover set covering and partitioning models, as well as apply GMM to find a set of prototypes that can optimise interpretability.

The rest of the sections are organised as follows. Section 2 is a literature review of this topic. Section 3 explains the three models: set covering, partitioning and GMM. Section 4 introduces the real-life data and simulated data. Sections 5 and 6 are the following results and conclusions.

2 Literature Review

As Ullmann et al. (2021) outlines, there are many ways of interpreting post-hoc clusters such as internal and external validations, stability analysis and visual inspection. Rousseeuw (1987) presented a graphical display technique that uses tightness and separation to form clusters. Their work uses dissimilarities, which measure how far away the two objects are from each other. This can be represented as a dissimilarity matrix of rows and columns at each data point. Similarly to the Rousseeuw (1987), this thesis also constructs a dissimilarity matrix.

There are methods invented to improve the interpretability of clustering. For trace clustering, which is a sequence of event logs, De Koninck et al. (2016) proposed an algorithm that finds the key attributes of a cluster and moves instances if it does not have those. This provides clear, concise rules that explain why a particular instance is part of a cluster. With clinical data, Balabaeva & Kovalchuk (2020) applied Bayesian inference to compare the prior and posterior distribution of features. This approach works for any clustering algorithm, however, refining features with medical experts is necessary as human interpretation differs from algorithmic interpretation. Furthermore, the set covering model and partitioning model constructed by Carrizosa et al. (2022) are verified that these are explanatory in terms of true positive rate (TPR) and false positive rate (FPR). TPR is the fraction of total individual data points that actually correspond to the classified groups, also known as true positive cases, divided by the number of data in its subpopulations. FPR is the fraction of total individual data points that are incorrectly classified, known as false positive cases, divided by the number of data in its subpopulations. This thesis utilises these two measures for interpretability.

Meanwhile, GMM has been researched in many studies, both in improving the model and application to real data. Implementing GMM to different data has been researched, such as GMM prototype modelling with fruit images Gerstenberger et al. (2023). They introduced a gradient-based GMM layer to detect prototypes when identifying images. GMM is also applied to cluster words in correct groups Chen et al. (2015). For example, the word ”Apple” can be interpreted as a fruit or an electronic company, and GMM is used to correctly cluster words related to it.

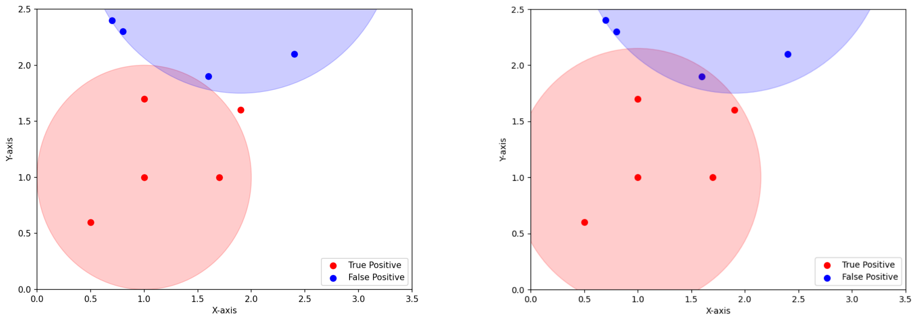
There are a few shortcomings with GMM, such as reliance on a pre-defined number of clusters and sensitivity to initial parameters. Yang et al. (2012) focused on finding the optimal number of clusters by producing robust algorithms that automatically obtain an optimal number of clusters. Nonetheless, due to complexity, a general algorithm along with information criterion is applied, as suggested by Patel & Kushwaha (2020) in this thesis. P. Wang & Wang (2017) introduced a density peak clustering to ensure the cluster captures the global optimum, which identifies noise according to the outlier degree of the point. The result showed that this algorithm is more effective. Patel & Kushwaha (2020) highlighted that capturing global optimum can be possible by initialising multiple times as well, which is incorporated as testing different numbers of clusters in this thesis. Other unique methods introduced are as follows. GMM can also be applied by clustering with rankings over a finite set of predefined labels, as Zhou et al. (2014) built a method that has similar predictive accuracy as other approaches. Combining Gaussian distribution and beta distribution leads to beta-GMM that Dai et al. (2009) invented, where results show that the proposed method has meaningful outcomes compared to separately modelling the two distributions.

Henceforth, the application of GMM in cluster analysis has been acknowledged, however, identifying its interpretability in terms of TPR and FPR is not well-researched, which is why it is intriguing to research for this thesis. If GMM can identify meaningful prototypes that yield good TPR and FPR, greater accuracy and reliability for handling real-life data are assured.

3 Model

This section starts with an explanation of the interpretation of clusters in terms of TPR and FPR, followed by the formulations of the three models: set covering, partitioning and GMM. The first two models are based on the formulation described by the paper Carrizosa et al. (2022). Application to larger instances is described after introducing the two models. The GMM is based on Reynolds (2009) and Wan et al. (2019).

As explained in the previous section, TPR is the fraction of total individual data points that correspond to the classified groups, also known as true positive cases. FPR is the fraction of total individual data points that are incorrectly classified, known as false positive cases. To visualise this, see Figure 1 where the red and blue dots are the data points, corresponding colours represent the actual cluster and the circle represents the allocated clusters.



(a) Clustering option 1

(b) Clustering option 2

Figure 1: Explanation of TPR and FPR in a cluster

In Figure 1a, the TPR of the red cluster is 0.8 since four out of five data points are selected, and the FPR is 0.0 as no blue points are within the red circle. However, in Figure 1b, TPR is 1.0 since all the red points are within the red circle, but FPR is 0.25 as one blue data point is also within the red circle. Hence, there is often a trade-off between TPR and FPR when conducting cluster analysis. The aim is to have as high TPR and as low FPR as possible.

The three models aim to identify prototypes that maximise the TPR minus FPR. Note that an individual is covered by a cluster if it is close enough to a prototype. Later, GMM is explained to define how prototypes are selected based on probability distribution. The difference between the two models and GMM is that the two models use Euclidean distance and its dissimilarities to determine the optimal prototypes, while GMM estimates a probability distribution and accounts for both mean and variance, as explained previously.

Mathematical notation which applies to both set covering and partitioning models is as follows. Predefined sets of clusters C determine how the individuals are allocated into each of $c \in C$. We have an individual $n \in N$, where $N = \bigcup_{c \in C} N_c$. Hence, a set of individuals belonging to a cluster c is defined as N_c. Each prototype i is drawn from a set of prototype candidates $I_c \subseteq N_c$ with $I = \bigcup_{c \in C} I_c$. To determine optimal prototypes, dissimilarities are necessary to quantify differences between the data points. This helps to group similar data points, meaning the combination of points with low dissimilarities. Multiple formulas can be applied, such as Manhattan distance and Cosine dissimilarity. In this thesis, Euclidean distance is selected to calculate the dissimilarity matrix δ_{in}, for every $i \in I$ and $n \in N$. The subsections below explain the three models.

3.1 Set Covering Formulation

This model considers that individuals are covered by cluster c if the dissimilarity is below a threshold value. The threshold value is a radius r_c for cluster c, of which the centre is the corresponding prototype, chosen when the model is optimised from prototype candidates $i \in I_c$. The value of the radius can differ for each cluster. The radius takes a discrete amount of values. This approach can lead to cases where individuals are covered by more than one radius, while some individuals may not be covered at all. Hence, constraints are added to verify that an individual belongs to only one prototype. Note that an extension to more than one prototype is possible. We aim to find optimal sets of cluster radii r_c and the prototypes.

This model is a Mixed Integer Linear Programming (MILP) formulation. There are four decision variables of which the first three are binary: π_{in}, z_i, y_{in}, and r_c. Let us formally introduce each variable. A binary decision variable π_{in} takes a value of 1 only if individual $n \in N$ lies in the ball of radius r_c centred at prototype $i \in I$. z_i is a binary decision variable that becomes 1 if the selected prototype $i \in I_c$ is an optimal prototype, and 0 otherwise. y_{in} is a binary decision variable introduced to avoid bi-linear formulation by setting $y_{in} = \pi_{in} z_i$, inspired by the Fortet transformation Fortet (1960). Throughout the paper, we use bold typesetting to denote the vectors, e.g., $\mathbf{z} = (z_i)_{i \in I}$.

With the variables defined above, the TPR and FPR can be formulated as follows. The number of true positive cases in cluster c is $\sum_{i \in I_c} \sum_{n \in N_c} \pi_{in} z_i$. Hence, the TPR of a cluster c

(TPR$_c$) is

$$\text{TPR}_c = \frac{\sum_{i \in I_c} \sum_{n \in N_c} \pi_{in} z_i}{|N_c|} \tag{1}$$

The number of false positive cases in cluster c is $\sum_{i \in I_c} \sum_{n \in N \setminus N_c} \pi_{in} z_i$ and the FPR can be shown as

$$\text{FPR}_c = \frac{\sum_{i \in I_c} \sum_{n \in N \setminus N_c} \pi_{in} z_i}{|N \setminus N_c|} \tag{2}$$

Below are the objective functions and constraints and each is explained afterwards.

$$\max_{\mathbf{z},\pi,\mathbf{r}} \sum_{c \in C} \sum_{i \in I_c} \sum_{n \in N_c} y_{in} - \theta \sum_{c \in C} \sum_{i \in I_c} \sum_{n \in N \setminus N_c} y_{in} \tag{3}$$

s.t.

$$\sum_{i \in I_c} z_i = 1, \quad \forall c \in C \tag{4}$$

$$r_c \geq \delta_{in} \pi_{in}, \quad \forall (i,n) \in I_c \times N_c, \forall c \in C \tag{5}$$

$$r_c \leq \delta_{in} + (r_c^{\max} - \delta_{in}) \pi_{in}, \quad \forall (i,n) \in I_c \times (N \setminus N_c), \forall c \in C \tag{6}$$

$$\sum_{i \in I_c} \sum_{n \in N_c} \pi_{in} z_i \geq \lceil \lambda_c |N_c| \rceil, \quad \forall c \in C \tag{7}$$

$$\sum_{i \in I_c} \sum_{n \in N \setminus N_c} \pi_{in} z_i \leq \lfloor \mu_c |N \setminus N_c| \rfloor, \quad \forall c \in C \tag{8}$$

$$r_c^{\min} \leq r_c \leq r_c^{\max}, \quad \forall c \in C \tag{9}$$

$$z_i \in \{0,1\}, \quad \forall i \in I_c, , \forall c \in C \tag{10}$$

$$\pi_{in} \in \{0,1\}, \quad \forall (i,n) \in I_c \times N, \forall c \in C \tag{11}$$

$$y_{in} \leq \pi_{in}, \quad \forall (i,n) \in I_c \times N, \forall c \in C \tag{12}$$

$$y_{in} \leq z_i, \quad \forall (i,n) \in I_c \times N, \forall c \in C \tag{13}$$

$$y_{in} \geq \pi_{in} + z_i - 1, \quad \forall (i,n) \in I_c \times N, \forall c \in C \tag{14}$$

$$y_{in} \in \{0,1\}, \quad \forall (i,n) \in I_c \times N, \forall c \in C \tag{15}$$

The objective function is equal to a maximisation of the total number of true positives minus the total number of false positives. The trade-off parameter θ weighs the importance of TPR over FPR with $\theta \geq 0$. Constraint 4 certifies that only one prototype is assigned per cluster. Constraint 5 ensures individuals are assigned to cluster c if the dissimilarity between an individual and a selected prototype is below the radius r_c, avoiding the case where $\pi_{in} = 1$ when $r_c < \delta_{in}$. Constraint 6 makes sure that if the individuals do not fall under the radius $r_c < \delta_{in}$, then they will fall under other clusters. Constraints 7 and 8 are constraints to ensure TPR above the threshold of the parameter λ and FPR below the threshold parameter μ. These two thresholds can take any value on the grid $\lambda \in [0.0, 1.0]$ and $\mu \in [0.0, 1.0]$. Constraints 9, 10, and 11 define the decision variables.

In this thesis, we assign $r_c^{\min}$ as the minimum value of dissimilarity values between two different data points, $r_c = \min\{\delta_{in} \mid i \in I_c, n \in N_c, i \neq n\}$ and $r_c^{\max}$ as the maximum value of dissimilarity values between two different data points, $r_c = \max\{\delta_{in} \mid i \in I_c, n \in N_c, i \neq n\}$. As explained previously, the decision variable y_{in} linearizes the bi-linear terms $\pi_{in} z_i$. This is ensured by Constraints 12-15. Hence, the set covering model with the above constraints is an MILP with $|I| \times |N| + |I|$ binary and $|C|$ continuous decision variables, and $|I| \times |N| + 4|C|$ linear constraints. Note that it is separable on the clusters.

3.2 Partitioning Model

The partitioning model does not have a threshold value and explains the prototype selection based on the closeness. It is also MILP. A new binary variable is introduced, namely ρ_{in}, which is 1 only if prototype i is the closest one to individual n from the chosen ones and 0 otherwise. This variable allocates each individual to prototypes. z_i variable is defined the same as before to select a prototype for each cluster $i \in I_c$. The TPR and FPR can be calculated similarly. The number of true positive cases in cluster c is $\sum_{i \in I_c} \sum_{n \in N_c} \rho_{in}$ which implies TPR is

$$\text{TPR}_c = \frac{\sum_{i \in I_c} \sum_{n \in N_c} \rho_{in}}{|N_c|} \tag{16}$$

Meanwhile, the number of false positive cases in cluster c is $\sum_{i \in I_c} \sum_{n \in N \setminus N_c} \rho_{in}$ and

$$\text{FPR}_c = \frac{\sum_{i \in I_c} \sum_{n \in N \setminus N_c} \rho_{in}}{|N \setminus N_c|} \tag{17}$$

The partitioning model is described below.

$$\max_{\mathbf{z}, \rho} \sum_{c \in C} \sum_{i \in I_c} \sum_{n \in N_c} \rho_{in} - \theta \sum_{c \in C} \sum_{i \in I_c} \sum_{n \in N \setminus N_c} \rho_{in} \tag{18}$$

s.t.

$$\sum_{i \in I_c} z_i = 1, \quad \forall c \in C \tag{19}$$

$$\sum_{j \in I_c : \delta_{jn} \leq \delta_{in}} z_j + \sum_{j \in I : \delta_{jn} > \delta_{in}} \rho_{jn} \leq 1, \quad \forall (i, n) \in I_c \times N, \forall c \in C \tag{20}$$

$$\rho_{in} \leq z_i, \quad \forall (i,n) \in I \times N \tag{21}$$

$$\sum_{i \in I} \rho_{in} = 1, \quad \forall n \in N \tag{22}$$

$$\sum_{i \in I_c} \sum_{n \in N_c} \rho_{in} \geq \lceil \lambda_c |N_c| \rceil, \quad \forall c \in C \tag{23}$$

$$\sum_{i \in I_c} \sum_{n \in N \setminus N_c} \rho_{in} \leq \lfloor \mu_c |N \setminus N_c| \rfloor, \quad \forall c \in C \tag{24}$$

$$z_i \in \{0,1\}, \quad \forall i \in I \tag{25}$$

$$\rho_{in} \in \{0,1\}, \quad \forall (i,n) \in I \times N \tag{26}$$

Similarly, as in set covering, the objective function is equal to a maximisation of the total number of true positives minus the total number of false positives. Again, θ represents the trade-off parameter between TPR and FPR. Constraint 19 ensures that only one prototype is selected per cluster, similar to Constraint 4 in set covering. Constraint 20 is an assignment constraint formed based on (Wagner & Falkson, 1975), to make sure each individual is assigned to the closest prototype and there cannot be another closer prototype. Constraint 21 follows up by certifying that individuals are assigned to prototypes that are selected as optimal. Constraint 22 ensures that exactly one prototype is assigned to each individual. Finally, constraints 23 and 24 are parameters controlling for TPR lower bound and FPR upper bound, followed by constraints 25 and 26 representing binary variables explained previously. Hence, the partitioning model with the above constraints is a MILP with $|I| \times |N| + |I|$ binary and $|C|$ continuous decision variables, and $2|I| \times |N| + 3|C| + |N|$ linear constraints. Note that it is again separable on the clusters.

3.3 Application to larger instances

To solve a large instance with the above two models, we use a reduction technique which has three steps: (1) perform a reduced model, (2) find solutions to larger instances, and (3) assess the quality.

For (1), we form a reduced model based on a sample drawn from the large dataset. To do so, perform hierarchical clustering on N_c, based on dissimilarity drawn from Euclidean distance. Choose a threshold that yields $|\tilde{N}_c|$, where $\tilde{N}_c \subset N_c$. Next, randomly select a point from each cluster, yielding a total of $|\tilde{N}_c|$ points. These $|\tilde{N}_c|$ points are the representative of the cluster with weights $\tilde{w}_n$, which is the number of data points in the cluster. Hence, the randomly selected point becomes an individual in $\tilde{N}_c$. To find a prototype candidate, we follow a similar approach where we perform hierarchical clustering on I_c. Choose a threshold that yields $|\tilde{I}_c|$ where $\tilde{I}_c \subset I_c$. Then, randomly select a point from each of the $\tilde{I}_c$, which becomes the total number of $|\tilde{I}_c|$ prototype candidates. The equations for the reduced model for set covering are shown below.

$$\text{TPR}_c = \frac{\sum_{i \in \tilde{I}_c} \sum_{n \in \tilde{N}_c} \pi_{in} z_i \tilde{w}_n}{\sum_{n \in \tilde{N}_c} \tilde{w}_n} \tag{27}$$

$$\text{FPR}_c = \frac{\sum_{i \in \tilde{I}_c} \sum_{n \in \tilde{N} \setminus \tilde{N}_c} \pi_{in} z_i \tilde{w}_n}{\sum_{n \in \tilde{N} \setminus \tilde{N}_c} \tilde{w}_n} \tag{28}$$

$$\max_{\mathbf{z},\pi,\mathbf{r}} \sum_{c \in C} \sum_{i \in \tilde{I}_c} \sum_{n \in \tilde{N}_c} y_{in} \tilde{w}_n - \theta \sum_{c \in C} \sum_{i \in \tilde{I}_c} \sum_{n \in \tilde{N} \setminus \tilde{N}_c} y_{in} \tilde{w}_n \tag{29}$$

$$\sum_{i \in \tilde{I}_c} \sum_{n \in \tilde{N}_c} \pi_{in} z_i \tilde{w}_n \geq \lceil \lambda_c \sum_{n \in \tilde{N}_c} \tilde{w}_n \rceil, \quad \forall c \in C \tag{30}$$

$$\sum_{i \in \tilde{I}_c} \sum_{n \in \tilde{N} \setminus N_c} \pi_{in} z_i \tilde{w}_n \leq \lfloor \mu_c \sum_{n \in \tilde{N} \setminus \tilde{N}_c} \tilde{w}_n \rfloor, \quad \forall c \in C \tag{31}$$

We solve the model with $\tilde{N}_c$, $\tilde{I}_c$ and assigning weights $\tilde{w}_n$ to replace $|N_c|$ in the TPR and FPR boundary constraints. In other words, change from constraints 7 and 8 to 30 and 31. Furthermore, $\tilde{w}_n$ also replaces the denominator in the TPR formula in equation 1 and FPR formula in equation 2, as shown in equations 27, 28. The weights also need to be multiplied for both TPR and FPR in the objective function as shown above.

Once the model is solved, store the optimal solution of $\mathbf{r}^R$ and $\mathbf{z}^R$ for $i \in \tilde{I}_c$, $c \in C$. In this thesis, $\mathbf{z}^R$ only stored the data point that yields 1.0; in other words, the data points that are selected as the optimal prototypes. This is a partial solution for the original problem with larger instances.

In (2), we apply $\mathbf{r}^O = \mathbf{r}^R$ and $\mathbf{z}^O = \mathbf{z}^R$. Since having greater instances could only reduce the TPR and or increase FPR, $\mathbf{r}^R$ and $\mathbf{z}^R$ already satisfy constraints 7 and 8, acting as an upper bound of TPR and lower bound of FPR. Hence, we can drop these two constraints when conducting the larger instances. Also, note that weights are removed in this model, so the original models described in previous sections are used. Moreover, since the $\mathbf{z}^R$ only contains one value per cluster, all the equations above with z_i are replaced by z_c, δ_{in} as $\delta_{z_c n}$ and π_{in} as $\pi_{z_c n}$. This implies removing any $\sum_{i \in I_c}$ and $\forall i \in I_c$. By default, equation 4 is eliminated.

In (3), recalculation of TPR and FPR are conducted with the corresponding optimal decision variables derived from solving (2). The formulas of TPR and FPR are the same as above, which are Equations 1 and 2.

For the partitioning model, reduction techniques explained in the set covering model can be applied. The difference is that it only stores $\mathbf{z}^R$ for $i \in \tilde{I}_c$, $c \in C$ in (1). This is a partial solution for the original problem with larger instances. In (2), we apply $\mathbf{z}^O = \mathbf{z}^R$ and similar procedures take place for the constraints and variables.

3.4 Gaussian Mixture Models (GMMs)

GMM is a probabilistic model for representing sub-populations within the total population that is normally distributed. A Gaussian distribution is defined by its mean vector μ and covariance matrix Σ. The probability density function (pdf) of a D-dimensional Gaussian distribution is given below.

$$\mathcal{N}(X|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(X-\mu)\right)$$

μ is the mean vector, Σ determines the shape of the distribution and is constructed by the $D \times D$ covariance matrix. $|\Sigma|$ defines the determinant of Σ. GMM in cluster analysis forms ellipsoidal shaped clusters based on probability density estimations, where each cluster is modelled as a Gaussian distribution. Hence, GMM in clustering is a linear combination of the Gaussian probability distribution where K is the number of clusters (or known as components) and π_k, known as a mixing coefficient, is an estimate of each Gaussian component. Hence, each component k is described by consisting of mean μ_k, covariance Σ_k and mixing coefficient π_k.

$$p(X) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

For a given set of N independent and identically distributed observations $\{x_1, x_2, \ldots, x_N\}$ The log-likelihood function can be written as:

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

The Expectation Maximisation (EM) algorithm finds Maximum Likelihood estimates (MLE) for GMM. This algorithm is an iterative method of MLE with latent variables. The steps are described below.

Algorithm 1 EM Algorithm for GMM

```
Initialize the parameters $\theta = (\pi_k, \mu_k, \Sigma_k)$ randomly.
repeat
  E-step: Compute the responsibilities using the current parameter values.
  for each data point $x_i$ do
    for each cluster $k$ do
      $\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j,\Sigma_j)}$
    end for
  end for
  M-step: Update the parameters using the current responsibilities.
  for each cluster $k$ do
    $\pi_k = \frac{1}{N}\sum_{i=1}^{N} \gamma_{ik}$
    $\mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} x_i}{\sum_{i=1}^{N} \gamma_{ik}}$
    $\Sigma_k = \frac{\sum_{i=1}^{N} \gamma_{ik}(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_{i=1}^{N} \gamma_{ik}}$
  end for
until No further changes in cluster assignment
```

It consists of two main steps, the Expectation step (E-step) and the Maximisation step (M-step). In the E-step, the values of the latent variables are estimated, with the values of model parameters fixed. In the M-step, new values for the model parameters are estimated to minimise an error function. Repeat these two steps until a convergence criterion is met. The algorithm is said to converge when there are no further cluster assignment changes.

In this thesis, the GMM is used to find a local optimal set of prototypes by fitting the data into a probability distribution. The idea is as follows. First, the initial parameters μ and Σ are generated from the data. Second, Algorithm 1 is conducted to find the optimal $\mu_{\mathbf{opt}}$ and $\mathbf{\Sigma_{opt}}$ for every cluster. As it is known to be computationally large, we utilise a *sklearn* package in Python. Finally, the data points closest to $\mu_{\mathbf{opt}}$ and $\mathbf{\Sigma_{opt}}$ are selected as optimal prototypes, per cluster. The set of optimal prototypes acts as I_c to the models above to find TPR and FPR.

GMM accounts for both mean and variance, which allows us to define clusters in various shapes, such as ellipses. However, this implies that applying selected prototypes by GMM to set covering formulation is not possible, since the radius is only defined to be spherical. In other words, even if GMM detects a non-spherical shape, individuals are only covered if it is within the sphere, which can worsen TPR and FPR. Therefore, GMM is only applied to the partitioning model in this thesis. Adjusting the models to have non-spherical clusters is for further research.

4 Data

Two datasets are used in this thesis: a real-life data set of Canadian daily average temperature, and a simulated data.

4.1 Canadian Weather

The first data is real-life data of Canadian weather representing the 365 daily average temperatures of Canadian cities. This can be extracted from the "fda" package in R. It is composed of 35 cities and 4 regions. In this case, the predefined clustering of the cities is based on the regions ["Atlantic", "Pacific", "Arctic" and "Continental"] the city is located in. Hence, $N = 35$ and $C = 4$. Figure 2 illustrates the data where the x-axis is the days and the y-axis represents the average temperatures. The cities are coloured according to what cluster they belong to: blue for Atlantic, purple for Continental, red for Pacific, and green for Arctic. Before applying the models, the dissimilarity matrix is calculated. By taking Euclidean distance between each city, a dissimilarity matrix with dimensions 35×35 is created.

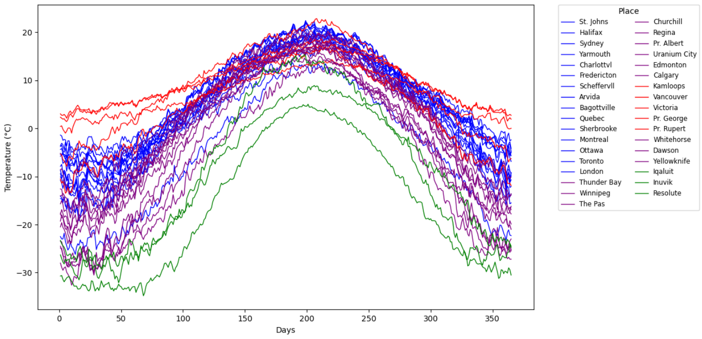


Figure 2: Canadian weather data.

Regarding the Canadian weather data, its distribution is unknown to us. As GMM is designed for normal distribution, testing the dataset is helpful to identify the accuracy of the results. First, the Kolmogorov-Smirnov Test is conducted, which rejects the null hypothesis that it is a normal distribution, see Table 1. The "fitter" package in Python is utilised to verify. The package allows 80 distributions to be fitted. Here, gamma, lognormal, beta, normal, and exponential distributions are fitted to the average daily temperatures. This package outputs a sum of squared residuals (SSR), with lower values indicating a greater fit. The beta distribution is well-fitted compared to the other distributions as a normal distribution.

Table 1: Results of the tests to determine the distribution of Canadian Weather

	SSR of fitted distributions				
Kolmogorov Smirnov Test statistics (p-value)	Normal Distrib.	Exponential Distrib.	Gamma Distrib.	Lognorm Distrib.	Beta Distrib.
0.097(0.002)	0.075	0.073	0.075	0.075	0.027

Although GMM is designed for normally distributed datasets, the daily average temperature of Canadian cities is beta-distributed as shown above. This leads to poor model fitting and inaccuracy because it does not capture the underlying structure of the data. To resolve this, inverse normal transformation is used before applying GMM. Applying for the Beta Mixture Model (BMM) is beyond the scope of the bachelor thesis as it has not been commonly researched and the published previous literature is based on specific data (such as Fu et al. (2010)). Generalising BMM is a further extension to be considered. Furthermore, GMM with beta-distributed data has been discussed by Dai et al. (2009) as mentioned earlier, however, this paper incorporates both Gaussian and beta-distributed data in its methodology which is not applicable here.

4.2 Simulated Data

The second data is simulated data to justify the interpretability with a larger dataset. It is three normally distributed data points with the mean and covariance described below. The predefined clusters are $C = 3$. The demonstration of the scatter plot is in Figure 3. Here, $N = 10000$ are plotted with corresponding clusters which are coloured in red, blue, and green.

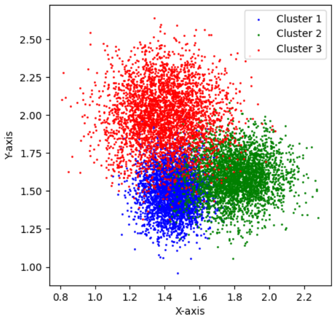


Figure 3: Simulated data

The values of mean and covariance are shown below. Notice that the covariance and the mean differences increase from cluster 1,2,3 respectively.

$$\beta_1 = \begin{pmatrix} 1.45 \\ 1.50 \end{pmatrix}, \quad \beta_2 = \begin{pmatrix} 1.80 \\ 1.60 \end{pmatrix}, \quad \beta_3 = \begin{pmatrix} 1.40 \\ 2.00 \end{pmatrix}$$

$$\Sigma_1 = \begin{pmatrix} 0.01 & 0.00 \\ 0.00 & 0.02 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 0.02 & 0.00 \\ 0.00 & 0.02 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 0.03 & 0.00 \\ 0.00 & 0.04 \end{pmatrix}$$

5 Results

This section is composed of three subsections. First, a general setting of the results, such as the PC I used and adjustments of the initial parameters are described. Second, the shortcoming of the GMM is mentioned. Finally, the results of Canadian weather and simulated data are presented.

5.1 General setting

First, to solve the mathematical optimisation we use the Gurobi package in Python on a PC Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz, 8GB of RAM. Furthermore, $scikit - learn$ package is used to solve the prototypes for GMM. Refer to the Appendix and the replication codes for further details.

Regarding the time limit, Carrizosa et al. (2022) had a maximum of 4.3GHz and set a time limit of 300 seconds. Since the laptop only has 1.8GHz, computational time can be much longer. For this purpose, the time limit is adjusted to 600 seconds when necessary.

Due to the low GHz and greater computational time, some models are infeasible within the 600-second time limit. For example, for the reduced partitioning model in simulated data, the optimal solution is only found for one of the clusters under 600 seconds. Meanwhile, all the models are feasible as the time limit increases to 1000 seconds. The Appendix shows the evidence in Table 2.

Furthermore, the model records feasible values if it does not reach the optimal solution after 600 seconds, hence, it is likely that the overall results differ with a PC with high computational power. Hence, the weak computational power is one of the reasons why the results can differ from Carrizosa et al. (2022) in simulated data. Hence, the detailed comparison of this thesis and Carrizosa et al. (2022) is only possible for the Canadian Weather Data as all the results are collected in under 300 seconds.

Throughout this thesis, $\theta = 1$. The results are created based on the *heatmap* package in Python. The white background represents model infeasibility. In the following section, for each dataset, the results of set covering and partitioning models are shown first, followed by GMM.

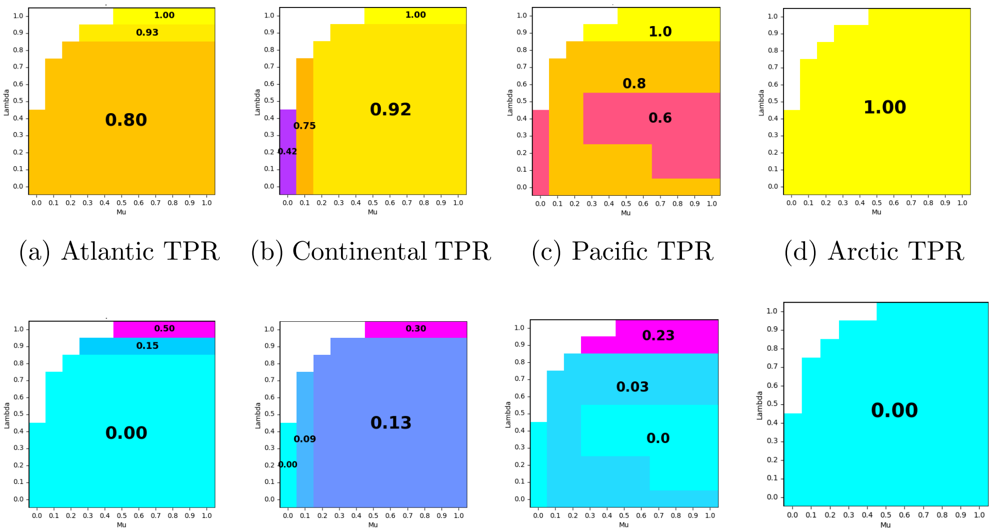
5.2 GMM results interpretation

The shortcomings with interpreting the results from GMM is that it has many infeasible solutions, and the reasoning is as follows. GMM pre-selects an optimal prototype based on the distribution of the data. Hence, z is predefined containing one data point assigned as a prototype per cluster when running the model. Meanwhile, the set covering model and the partitioning model described above are designed to assign optimal prototypes that maximise interpretability. This implies that the prototypes that are selected by GMM may contradict the prototypes that are selected by the models for a given λ and μ. It may result in infeasible solutions for some combinations of these as it may not satisfy the lower bound of TPR and upper bound of FPR constraints. To determine whether GMM optimises prototype selection, only the feasible results will be interpreted and compared in this thesis. Finding a method to implement GMM in the above two models is potential future research to be considered.

5.3 Canadian Weather data

The results of the Canadian Weather data are presented below where λ and μ vary on the grid $[0.0, 1.0] \times [0.0, 1.0]$. Generally, the set covering model has good interpretability, shown in Figure 4. In other words, there exist some trade-off between FPR and TPR depending on the combination of λ and μ. For example, $(\lambda, \mu) = (0.80, 0.20)$ then we have $TPR_{Atlantic} = 0.80$, $TPR_{Pacific} = 0.80$, whereas $FPR_{Atlantic} = 0.00$, $FPR_{Pacific} = 0.03$. However, when we increase the lower bound of TPR, such as $(\lambda, \mu) = (0.90, 0.30)$, then we have greater values in 2 clusters, namely, increase by 0.13 for $TPR_{Atlantic}$ and 0.2 for $TPR_{Pacific}$. On the other hand, the FPR increased for those clusters by 0.15 for $FPR_{Atlantic}$ and 0.20 for $FPR_{Pacific}$. To summarise, the higher the value of lambda which restricts TPR value, the worse the FPR. For the *Arctic*, the values stay the same regardless of the combination of λ and μ.

On the other hand, Carrizosa et al. (2022)'s set covering model does not have exactly the pattern. For example, they find different combinations of λ and μ. as $TPR_{Pasific} = 0.6$ and $FPR_{Pasific} = 0.0$. One possible reason is the use of different versions of the Gurobi package. This thesis uses a newer package which may result in more advanced and accurate outcomes. Another possible cause is the calculation of distances, since Carrizosa et al. (2022) does not mention how they calculated the distance precisely, it may have resulted in a different dissimilarity matrix. The results of the partitioning model is explained below.



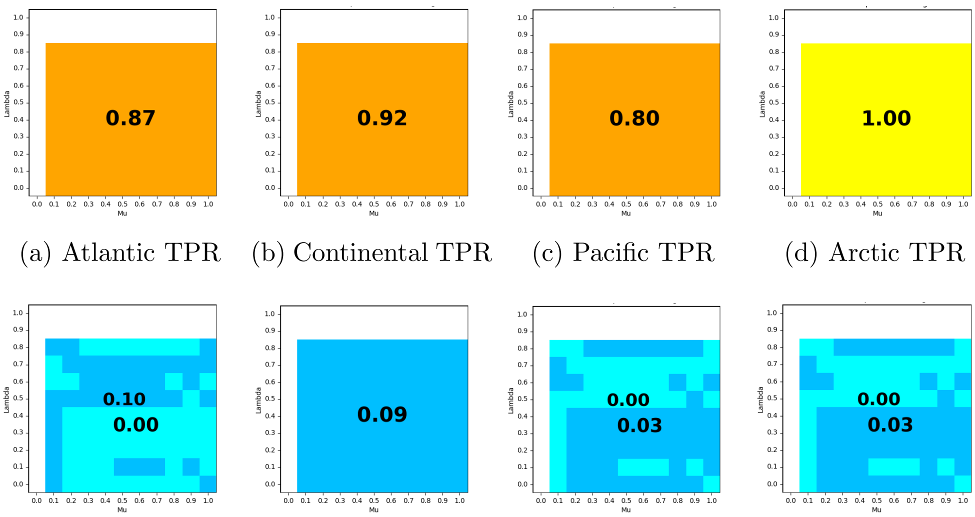
(a) Atlantic TPR (b) Continental TPR (c) Pacific TPR (d) Arctic TPR

(e) Atlantic FPR (f) Continental FPR (g) Pacific FPR (h) Arctic FPR

Figure 4: Set covering for Canadian weather data. λ and μ vary on the grid $[0.0, 0.1] \times [0.0, 1.0]$

The result of the partitioning model shown in Figure 5 is straightforward; there is simply no trade-off with greater TPR for worse FPR since TPR values do not alter regardless of the combination of λ and μ. However, there is a trade-off between what FPR to choose depending on the regional interpretation. Specifically, if we look at the pattern of the FPR, *Pacific* and *Arctic* have the same values, whereas for *Atlantic* the pattern is the same but the FPR values differ. For example, if we choose $(\lambda, \mu) = (0.80, 0.10)$ then the result is $FPR_{Atlantic} = 0.10$, $FPR_{Pacific} = 0.00$, $FPR_{Arctic} = 0.00$. However, when we choose $(\lambda, \mu) = (0.70, 0.10)$ the value of $FPR_{Atlantic}$ decreases by 0.1 while $FPR_{Pacific}$ and FPR_{Arctic} increase by 0.03. Hence, while TPR values do not change throughout, FPR can vary depending on the region to focus on.

On the other hand, Carrizosa et al. (2022)'s partitioning model does not have the same pattern because they found one FPR value per region which does not vary across λ or μ. The same reasoning as the set covering model can be applied, where the versions of the Gurobi package and the calculation of distances may influence the results.



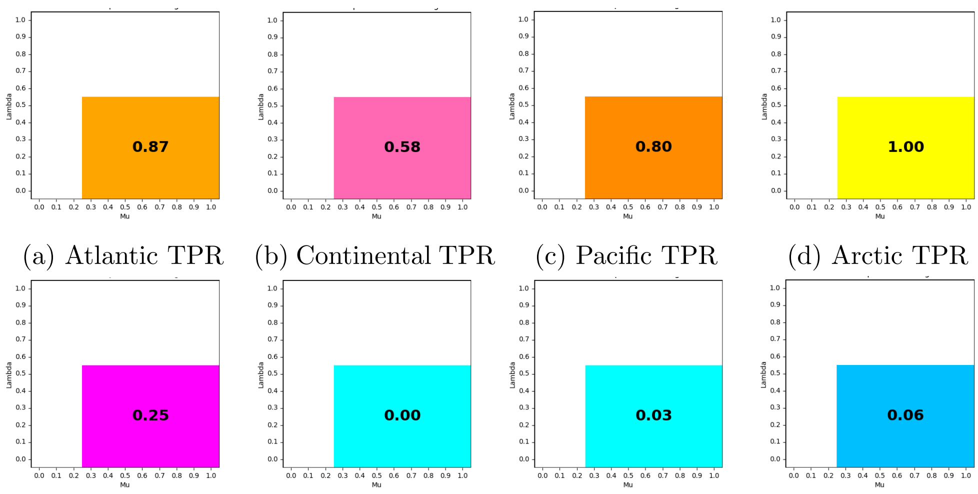
(a) Atlantic TPR (b) Continental TPR (c) Pacific TPR (d) Arctic TPR

(e) Atlantic FPR (f) Continental FPR (g) Pacific FPR (h) Arctic FPR

Figure 5: Partitioning model for Canadian Weather data. λ and μ vary on the grid $[0.0, 0.1] \times [0.0, 1.0]$

Next, the results of GMM applied to the partitioning model are shown. As explained in the previous section, GMM resulted in more infeasible regions, meaning a greater white background in the heatmap. In the following, the outcomes of the two models are shown. First, the results of regular GMM are presented, followed by the results from GMM with inverse normal transformation.

Figure 6 shows the TPR and FPR derived based on the model with the optimal prototypes selected by GMM. The result is worse than the regular partitioning model (Figure 5). For example, FPR is generally higher, especially noticeable with $FPR_{Atlantic} = 0.25$. Except for *Continental* where it improved by 0.09. TPR is the same for all except deterioration by 0.34 in *Continental*. This is expected as Canadian Weather data fitted β-distribution well, while GMM is designed for Normal distribution.

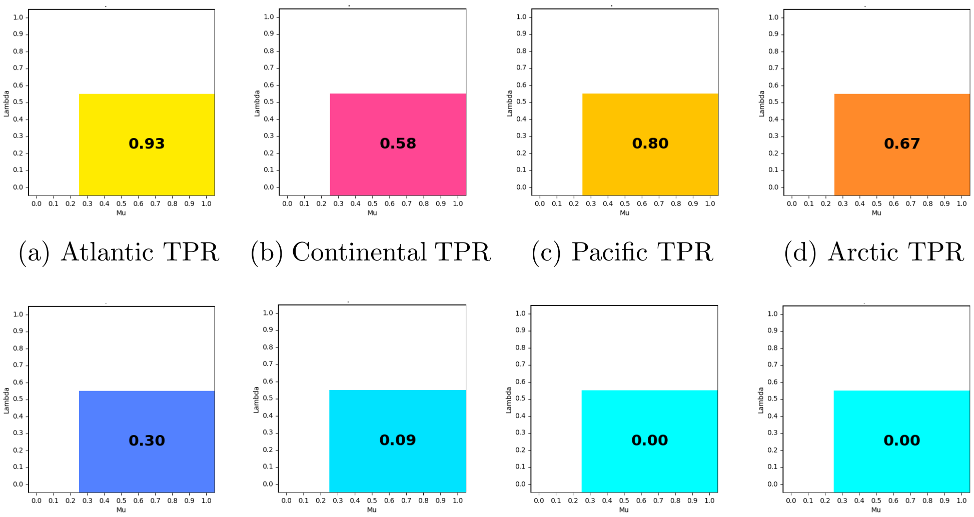


(a) Atlantic TPR (b) Continental TPR (c) Pacific TPR (d) Arctic TPR

(e) Atlantic FPR (f) Continental FPR (g) Pacific FPR (h) Arctic FPR

Figure 6: Gaussian Mixture Model with Canadian Weather data. λ and μ vary on the grid $[0.0, 0.1] \times [0.0, 1.0]$

We focus on Figure 7 where the data is inversely normally transformed to apply GMM. Compared to the original partitioning model in Figure 5, Figure 7 three models worsened by approximately 0.3 for $TPR_{Continental}$, TPR_{Arctic} and $FPR_{Atlantic}$. However, some models improved, in comparison to the results obtained by Carrizosa et al. (2022) and partitioning model (Figure 5), which are $TPR_{Atlantic}$, $FPR_{Pacific}$ and FPR_{Arctic}, by 0.03 or 0.06. Compared to the previous model which is GMM without inverse normal transformation (Figure 6), it has favourable results for $TPR_{Atlantic}$ and smaller FPR for $FPR_{Pacific}$ and FPR_{Arctic}. Nonetheless, it has worse TPR_{Arctic}, $FPR_{Atlantic}$ and $FPR_{Continental}$. Therefore, if we want greater improvement for the specific regions, GMM with inverse normal transformation is preferred, while the importance is equal for all, the original model is preferred. In other words, there is a trade-off between putting more importance on $TPR_{Atlantic}$, $FPR_{Pacific}$ and FPR_{Arctic}, or $FPR_{Continental}$, TPR_{Arctic} and $FPR_{Atlantic}$. Overall, the inversely normally transformed data gave mixed results that caused some of the FPR to improve while worsening TPR and visa versa.



(a) Atlantic TPR (b) Continental TPR (c) Pacific TPR (d) Arctic TPR

(e) Atlantic FPR (f) Continental FPR (g) Pacific FPR (h) Arctic FPR

Figure 7: Gaussian Mixture Model with Inverse Normal Transformed Canadian Weather data. λ and μ vary on the grid $[0.0, 0.1] \times [0.0, 1.0]$

Before moving on to Simulated data, the summary of this section is given. The results of set covering and partitioning models are similar to Carrizosa et al. (2022). However, the results of a few combinations of λ and μ do alter, despite using the same data. This is likely due to differences in Gurobi packages and ambiguity in calculating Euclidean distance and constructing dissimilarity matrix.

Next, we summarise the implementation of GMM. First, simple GMM resulted in a worse than regular partitioning model, despite an improvement in $FPR_{Continental}$. Second, GMM with Inverse normal transformation had mixed results. The values improved in $TPR_{Atlantic}$, $FPR_{Pacific}$ and FPR_{Arctic}, however, worsened in $TPR_{Continental}$, TPR_{Arctic} and $FPR_{Atlantic}$. Hence, despite some improvements, the regular partitioning model and set covering model (Figure 4 and 5) are favoured seeing the overall TPR and FPR values, however, if the focus is on particular regions, GMM with inverse normal transformation could be beneficial. Application of GMM to non-normal distributed data is a possible further extension for the future.

5.4 Simulated data

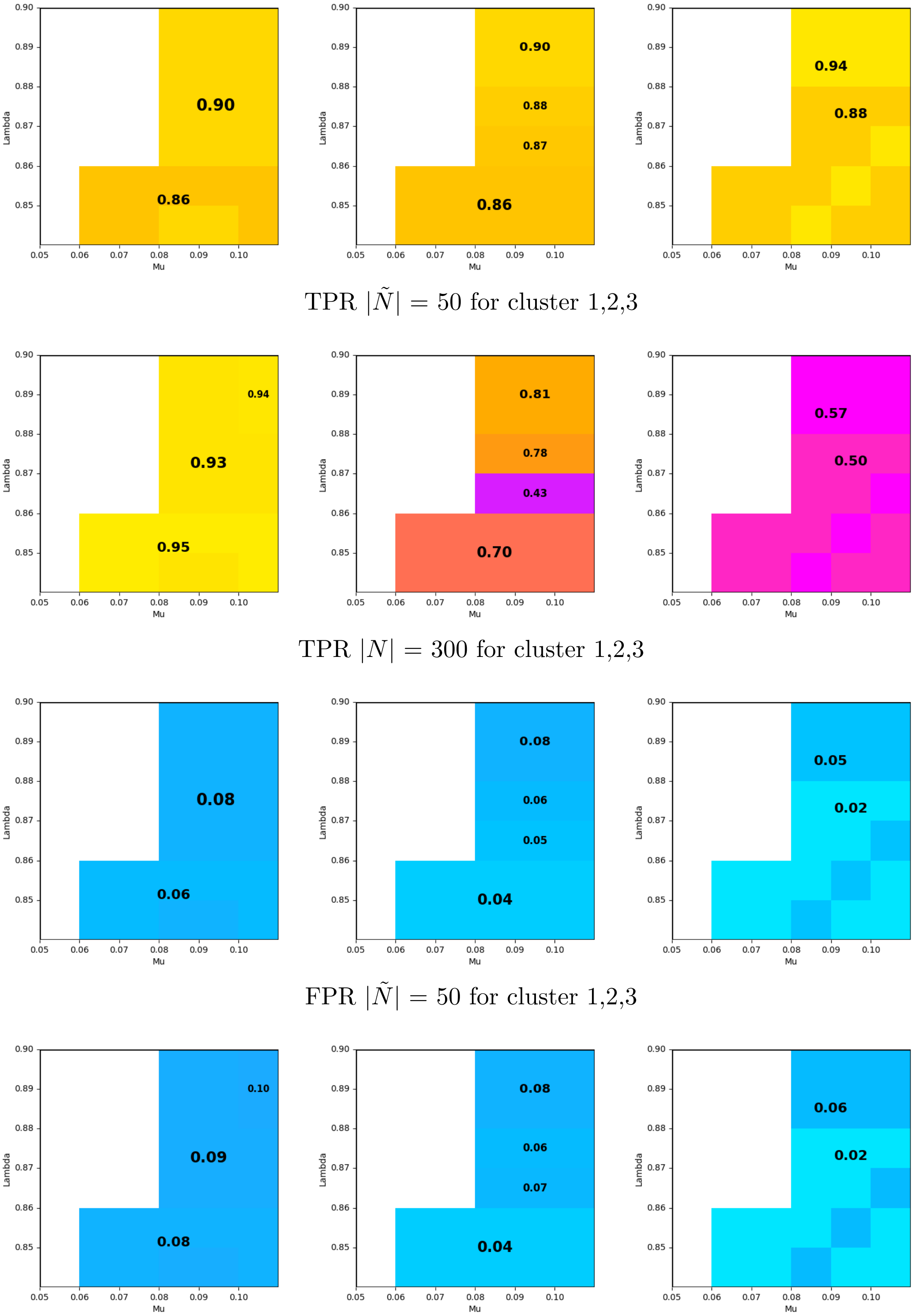
As explained in the previous section, two steps are involved in simulated data. First, the reduction technique is conducted to get a reduced model, and second, large instances are applied to this model.

Carrizosa et al. (2022) applied this technique to $|N| \in 10^4, 10^5, 10^6$ with $|\tilde{N}_c| = 125$, $|\tilde{I}_c| = 25$. However, it is not possible with the current PC, due to the weak computational power that led to hours to get one optimal solution, and most importantly, a memory error occurs when creating a dissimilarity matrix for larger instances from $|N| \in 10^4$. Hence, to present that the methodology is valid, the reduction technique is conducted with a smaller dataset of $|N| = 300$ with $|\tilde{N}_c| = 50$, $|\tilde{I}_c| = 15$. (See Appendix A.2 for the attempt on reduced model with $|N| = 10^4$ with $|\tilde{N}_c| = 125$, $|\tilde{I}_c| = 25$.) The values of λ and μ vary on the grid $[0.05, 0.1] \times [0.85, 0.9]$. We now discuss the results of the set covering model followed by the partitioning model, then GMM.

The set covering model with a sample of 300 datasets is shown in Figure 8. Seeing $|\tilde{N}| = 50$, the values vary in each cluster in TPR and FPR, especially in cluster 1. This is possibly due to the smaller dataset causing larger distances between each cluster, resulting in different sets of prototypes per combination of λ and μ. It is also clear here that the trade-off between high TPR and low FPR exists, as a higher lower bound on TPR (implying higher λ) implies greater TPR but also with larger FPR.

In general, it is clear that $|N| = 300$ leads to worse outcomes compared to the reduced model. Specifically, cluster 3 has TPR of 0.5 and 0.57 which is considerably low compared to Carrizosa et al. (2022), where the lowest TPR value recorded is 0.85. Meanwhile, FPR values do increase but by a small amount. For example, cluster 3 does not change except from 0.05 to 0.06, and part of cluster 1 changed from 0.08 to 0.09 and 0.10. This implies that having a small dataset does not affect the FPR as much as TPR. Most likely this is because of the higher covariance and larger mean differences in clusters 2 and 3, resulting in more scattered data points with low concentration around the mean (see Data Section).

Three factors could have affected the results of the $|N| = 300$. First, a small dataset implies more scattered data points. The weights do account for the selected data points in the reduced model, however, the selection of a point for each of the 50 clusters as well as for 15 prototypes might have led to the undesirable summarisation of the data leading to lower TPR and high FPR, compared to Carrizosa et al. (2022) where they used $|N| = 10^4$. Second, the randomisation of selecting data points from each of the 50 hierarchical clusters as well as 15 prototypes could highly have led to different optimisation than the Carrizosa et al. (2022) which makes it difficult to compare. Finally, the elimination of the lower bound of TPR and the upper bound of FPR constraints for the $|N| = 300$, as explained in the Model Section. This could have resulted in a much lower TPR and higher FPR than expected.



TPR $|\tilde{N}| = 50$ for cluster 1,2,3

TPR $|N| = 300$ for cluster 1,2,3

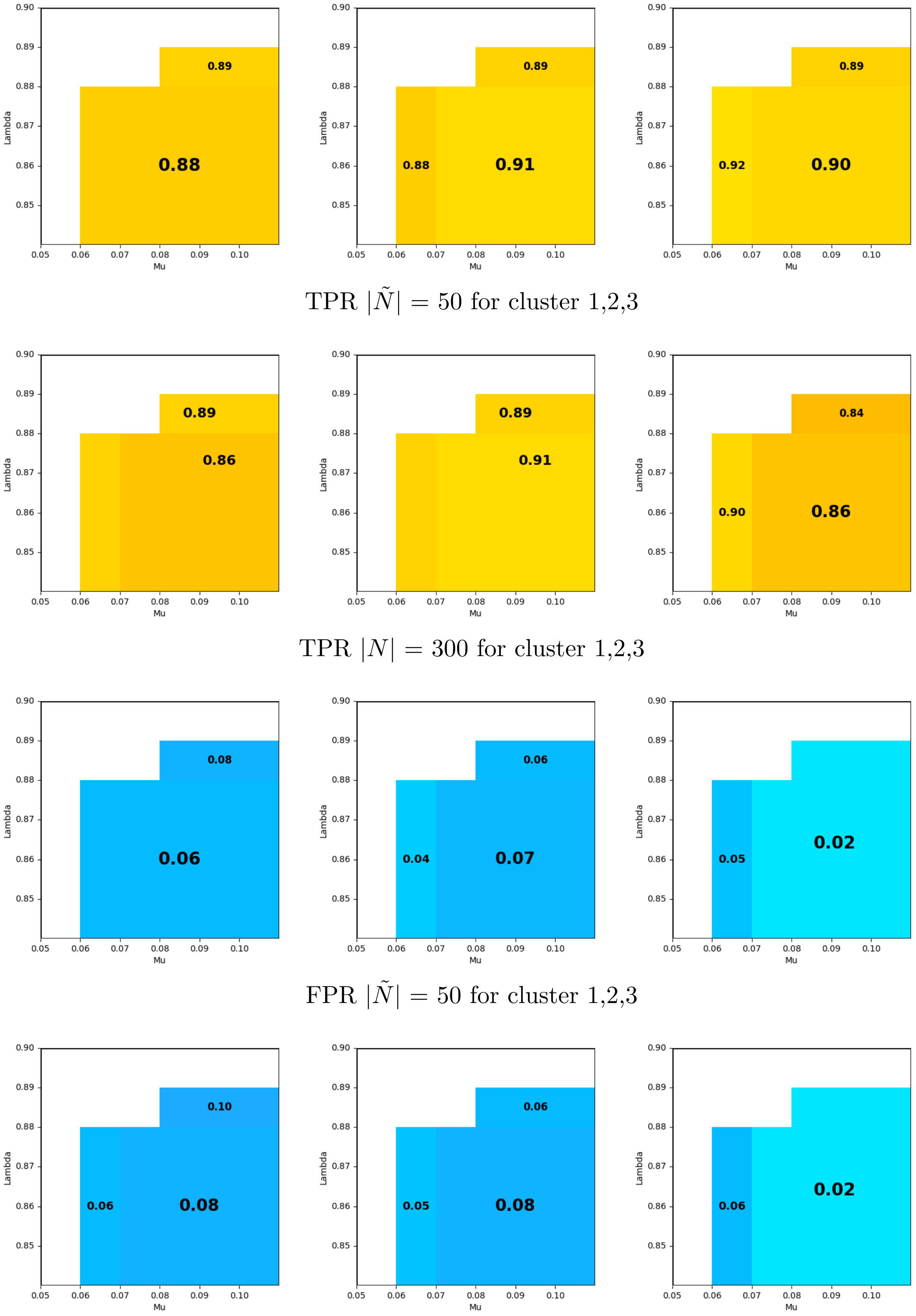
FPR $|\tilde{N}| = 50$ for cluster 1,2,3

FPR $|N| = 300$ for cluster 1,2,3

Figure 8: Set covering model with reduction technique. True Positive Rate and False Positive Rate in the set covering the increasing data size. λ and μ vary on the grid $[0.05, 0.1] \times [0.85, 0.9]$

The results of the partitioning model are shown below in Figure 9. Seeing $|\tilde{N}| = 50$, the TPR value reduces when λ increases in clusters 2 and 3. Meanwhile, cluster 1 still follows the same trend as the results in the set covering model, where a rise in λ leads to greater TPR along with a worse FPR. With $|N| = 300$, the results worsened or stayed the same for most of the values. For example, FPR values of cluster 1 rise by 0.02 in the larger dataset for some areas of 0.06 and 0.08 in the reduced model. On the other hand, some combinations of λ and μ led to a strangely favourable outcome, such as the rise in TPR from 0.88 to 0.89 for clusters 1 and 2 $\mu = 0.06$ region. The possible reason for this irregular pattern is due to the same reasoning as the first factor in the previous paragraph, where summarising clusters with weights might have led to unrealistic data points.

Compared with the set covering model, the overall results seem stable with the partitioning model when conducting the reduction technique and then applying it to larger instances. Specifically, as λ rises, the results of the partitioning model are not as elevated as the set covering model and only two values are recorded in the feasible region. Furthermore, the TPR in set covering does not reduce to below 0.86 in clusters 2 and 3 when applying to $|N| = 300$. Some regions in FPR are more favourable in the set covering than the partitioning models, though it depends on the values of λ and μ.



TPR $|\tilde{N}| = 50$ for cluster 1,2,3

TPR $|N| = 300$ for cluster 1,2,3

FPR $|\tilde{N}| = 50$ for cluster 1,2,3

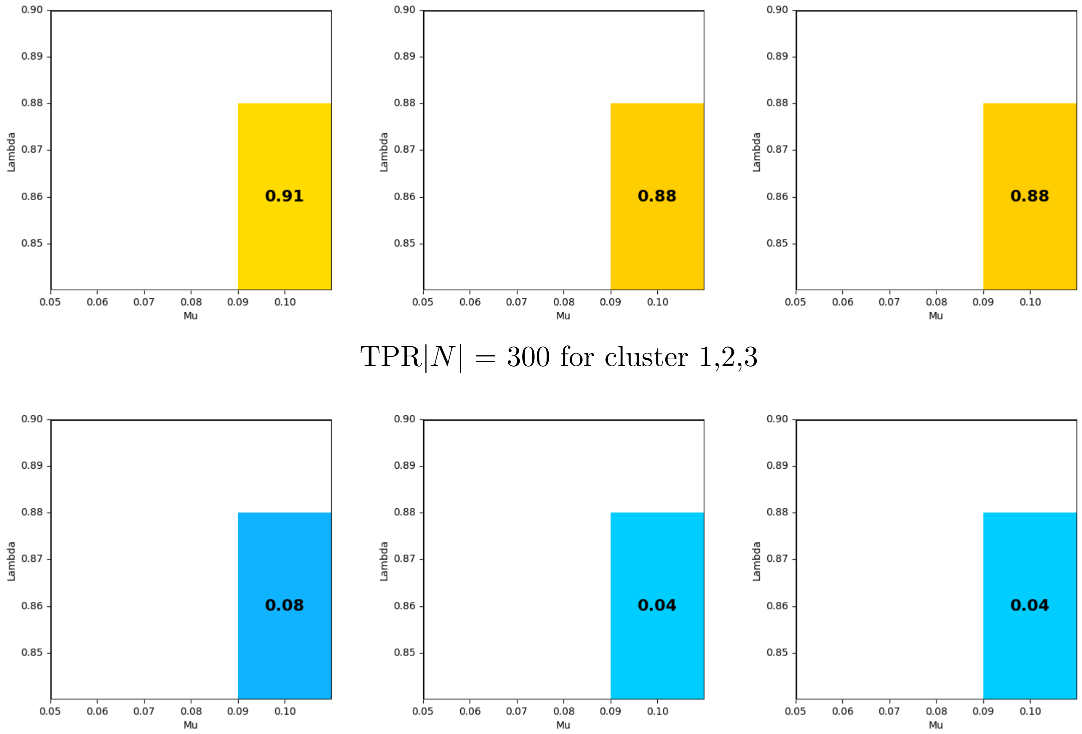
FPR $|N| = 300$ for cluster 1,2,3

Figure 9: Partitioning model with reduction technique. True Positive Rate and False Positive Rate with the increasing data size. λ and μ vary on the grid $[0.05, 0.1] \times [0.85, 0.9]$

The results of GMM are shown below in Figure 10. There are many infeasible regions as expected, due to applying only one set of prototypes to the entire combination of λ and μ. Overall results are analysed by comparing to the previous two models.

Unlike the results in $|N| = 300$ in the set covering model (Figure 8), the GMM has a single value throughout its feasible region. Overall, it showed a larger TPR compared to the set covering model. To be precise, the TPR values of clusters 2 and 3 are greater by at least 0.1 and 0.31 but worse by 0.04 for cluster 1 in GMM compared to the set covering model. GMM has a slightly mixed outcome for FPR, as it improved by 0.04 in cluster 2 but worsened by 0.02 in cluster 3. However, the general improvement in TPR is much larger than the changes in FPR.

Moving on to comparing the results in $|N| = 300$ of the partitioning model(Figure 9) against GMM, the interpretability does improve for two clusters but is slightly worse for one of them. Especially the TPR values improved for clusters 1 by 0.05 and 3 by 0.02 under the same λ and μ combinations. However, it worsened by 0.03 for cluster 2. The values of FPR do not change for cluster 1, improved by 0.04 in cluster 2, worsened by 0.02 in cluster 3. Hence, the GMM does not have a clear improvement when compared to the partitioning model.



TPR$|N| = 300$ for cluster 1,2,3

FPR $|N| = 300$ for cluster 1,2,3

Figure 10: GMM with simulated data. True Positive Rate and False Positive Rate. λ and μ vary on the grid $[0.05, 0.1] \times [0.85, 0.9]$

To summarise the simulated data section, when the reduction technique is conducted with $|\tilde{N}| = 50$ and then with $|N| = 300$, the set covering results in much worse TPR outcomes when the data is more dispersed (as cluster 3 has larger covariance compared to the other two). On the other hand, FPR values are not affected or are affected by a small amount compared to the changes in TPR. Hence, especially with the small and dispersed dataset, the set covering model has the least favourable performance. Meanwhile, the partitioning model has relatively stable results when the reduced model is applied to the larger instances ($|N| = 300$), as its TPR does not reduce below 0.86. The recorded FPR values are similar, hence, the more favourable model depends on the values of λ and μ. Generally, GMM showed a greater improvement in TPR values compared to the set covering model, but with slightly worsened regions in FPR. Finally, comparing GMM with the partitioning model shows mixed results, where for some clusters the TPR or FPR are better, but for others, it is the same or worse.

6 Conclusion

In this thesis, the interpretation of post-hoc clustered models is analysed in terms of TPR and FPR. Two models are studied as a basis: the set covering model and the partitioning model. These two models output prototypes, allowing us to determine the interpretability of a dataset. The goal is to select prototypes that achieve high interpretability, which is high TPR and low FPR as possible. The clustering is established on the Euclidean distance. Therefore, the closeness of the data points is a measure to find prototypes. To tackle the limitation of the two models that these only take the average distance into account, as well as assume data as circular clusters, the GMM is introduced, which uses both the mean and the covariance, as well as detecting ellipsoidal-shaped clusters based on maximum probability density estimations. As GMM is designed for normally distributed data, the inverse normal transformation is applied to beta-distributed real-life data.

The results of set covering and partitioning models in the real-life (Canadian weather) data are similar to Carrizosa et al. (2022). However, due to differences in Gurobi packages and ambiguity in calculating Euclidean distance, the results were not exactly the same.

Implementing GMM in real-life data does not improve all the results, even with the inverse normal transformation. Some of the regions had better results than the partitioning and the set covering models, which resulted in a trade-off between which regions to give more importance to. Therefore, GMM with inverse normal transformation is only better than the regular two models if the importance lies on particular regions.

In the simulated data, the reduction technique is conducted with $|\tilde{N}| = 50$ and then with $|N| = 300$. This is due to the memory error when constructing the large dissimilarity matrix with $|N| = 10000$ for the reduced model as done by the original author Carrizosa et al. (2022). In general, the results of the set covering result in much lower TPR outcomes when the data is more dispersed (as cluster 3 has a larger covariance compared to the other two). Especially with the small and dispersed dataset, the set covering model has the least favourable performance. Meanwhile, the partitioning model has relatively stable results when the reduced model is applied to the larger instances ($|N| = 300$), as its TPR does not reduce below 0.86. Generally, GMM showed a greater improvement in TPR values compared to the set covering model but compared to the partitioning model it shows mixed results.

The three possible areas of further research are as follows. Firstly, find a method to directly implement GMM in the above two models, since there is only one set of prototypes selected by GMM, it may contradict the prototypes that the models selected. It resulted in many infeasible solutions as it does not satisfy the lower bound of TPR and upper bound of FPR constraints.

Secondly, research into fitting Canadian data into the Beta Mixture Model. Generalising this model may improve the interpretability of the Canadian data, and extend towards research in the Beta Mixture Models. Finally, extend the set covering model that allows adjusting to the non-spherical radius, for example, by allowing two radii of different sizes. This is beneficial as it will capture the cluster characteristics.

References

Ashley, R. & Lloyd, J. (1978, Dec). An example of the use of factor analysis and cluster analysis in groundwater chemistry interpretation. *Journal of Hydrology*, *39*(3–4), 355–364. doi: 10.1016/0022-1694(78)90011-2

Balabaeva, K. & Kovalchuk, S. (2020). Post-hoc interpretation of clinical pathways clustering using bayesian inference. *Procedia Computer Science*, *178*, 264–273. doi: 10.1016/j.procs.2020.11.028

Carrizosa, E., Kurishchenko, K., Marín, A. & Morales, D. R. (2022). Interpreting clusters via prototype optimization. *Omega*, *107*, 102543.

Chen, X., Qiu, X., Jiang, J. & Huang, X. (2015). Gaussian mixture embeddings for multiple word prototypes. *CoRR*, *abs/1511.06246*. Retrieved from `http://arxiv.org/abs/1511.06246`

Corral, G., Armengol, E., Fornells, A. & Golobardes, E. (2009, Aug). Explanations of unsupervised learning clustering applied to data security analysis. *Neurocomputing*, *72*(13–15), 2754–2762. doi: 10.1016/j.neucom.2008.09.021

Dai, X., Erkkilä, T., Yli-Harja, O. & Lähdesmäki, H. (2009, May). A joint finite mixture model for clustering genes from independent gaussian and beta distributed data. *BMC Bioinformatics*, *10*(1). doi: 10.1186/1471-2105-10-165

De Koninck, P., De Weerdt, J. & vanden Broucke, S. K. (2016, Dec). Explaining clusterings of process instances. *Data Mining and Knowledge Discovery*, *31*(3), 774–808. doi: 10.1007/s10618-016-0488-4

Dransfield, E., Morrot, G., Martin, J.-F. & Ngapo, T. (2004, Jul). The application of a text clustering statistical analysis to aid the interpretation of focus group interviews. *Food Quality and Preference*, *15*(5), 477–488. doi: 10.1016/j.foodqual.2003.08.004

Dronov, S. V. & Evdokimov, E. A. (2018, May). Post-hoc cluster analysis of connection between the forming characteristics. *Model Assisted Statistics and Applications*, *13*(2), 183–195. doi: 10.3233/mas-180429

Fortet, R. (1960). Applications de l'algebre de boole en recherche opérationelle. *Revue Française de Recherche Opérationelle*, *4*(14), 17–26.

Fu, R., Dey, D. K. & Holsinger, K. E. (2010, Nov). A beta-mixture model for assessing genetic population structure. *Biometrics*, *67*(3), 1073–1082. doi: 10.1111/j.1541-0420.2010.01506.x

García, S. & Marín, A. (2019). Covering location problems. *Location Science*, 99–119. doi: 10.1007/978-3-030-32177-2_5

Gerstenberger, M., Maaß, S., Eisert, P. & Bosse, S. (2023, Oct). A differentiable gaussian prototype layer for explainable fruit segmentation. *2023 IEEE International Conference on Image Processing (ICIP)*. doi: 10.1109/icip49359.2023.10222905

Gibert, K. & Conti, D. (2014, May). On the understanding of profiles by means of post-processing techniques: An application to financial assets. *International Journal of Computer Mathematics*, *93*(5), 807–820. doi: 10.1080/00207160.2014.898065

Huang, Z. & Gou, Z. (2024, Jun). Optimizing battery energy storage prototypes for improved resilience in commercial buildings: Gaussian mixture modeling and hierarchical analysis of energy storage potential. *Energy and Buildings*, *312*, 114187. doi: 10.1016/j.enbuild.2024.114187

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B. & Heming, J. (2023, Apr). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, *622*, 178–210. doi: 10.1016/j.ins.2022.11.139

Liu, T.-C., Kalugin, P. N., Wilding, J. L. & Bodmer, W. F. (2022, Nov). Gmmchi: Gene expression clustering using gaussian mixture modeling. *BMC Bioinformatics*, *23*(1). doi: 10.1186/s12859-022-05006-0

Marín, A. & Pelegrín, M. (2019). P-median problems. *Location Science*, 25–50. doi: 10.1007/978-3-030-32177-2_2

Patel, E. & Kushwaha, D. S. (2020). Clustering cloud workloads: K-means vs gaussian mixture model. *Procedia Computer Science*, *171*, 158–167. doi: 10.1016/j.procs.2020.04.017

Reynolds, D. (2009). *Gaussian mixture models.* Department of Defense under Air Force Contract FA8721-05-C-0002. Retrieved from `http://leap.ee.iisc.ac.in/sriram/teaching/MLSP_16/refs/GMM_Tutorial_Reynolds.pdf`

Rousseeuw, P. J. (1987, Nov). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. doi: 10.1016/0377-0427(87)90125-7

Shi, F. & Huang, H. (2017, Jul). Identifying cell subpopulations and their genetic drivers from single-cell rna-seq data using a biclustering approach. *Journal of Computational Biology*, *24*(7), 663–674. doi: 10.1089/cmb.2017.0049

Ullmann, T., Hennig, C. & Boulesteix, A. (2021, Dec). Validation of cluster analysis results on validation data: A systematic framework. *WIREs Data Mining and Knowledge Discovery*, *12*(3). doi: 10.1002/widm.1444

Wagner, J. L. & Falkson, L. M. (1975, Jan). The optimal nodal location of public facilities with price-sensitive demand. *Geographical Analysis*, *7*(1), 69–83. doi: 10.1111/j.1538-4632.1975.tb01024.x

Wan, H., Wang, H., Scotney, B. & Liu, J. (2019). A novel gaussian mixture model for classification. , 3298–3303.

Wang, P. & Wang, J. (2017). A clustering algorithm based on find density peaks. *Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering*, 81–85. doi: 10.18178/wcse.2017.06.013

Wang, Z., Da Cunha, C., Ritou, M. & Furet, B. (2019). Comparison of k-means and gmm methods for contextual clustering in hsm. *Procedia Manufacturing*, *28*, 154–159. doi: 10.1016/j.promfg.2018.12.025

Yang, M.-S., Lai, C.-Y. & Lin, C.-Y. (2012, Nov). A robust em clustering algorithm for gaussian mixture models. *Pattern Recognition*, *45*(11), 3950–3961. doi: 10.1016/j.patcog.2012.04.031

Zeng, Y.-L., Xu, H.-B., Wu, G.-W. & BaI, S. (2011, Jan). Clustering algorithm based on the distributions of intrinsic clusters. *Journal of Software*, *21*(11), 2802–2813. doi: 10.3724/sp.j.1001.2010.03677

Zhou, Y., Liu, Y., Gao, X.-Z. & Qiu, G. (2014, Dec). A label ranking method based on gaussian mixture model. *Knowledge-Based Systems*, *72*, 108–113. doi: 10.1016/j.knosys.2014.08.029

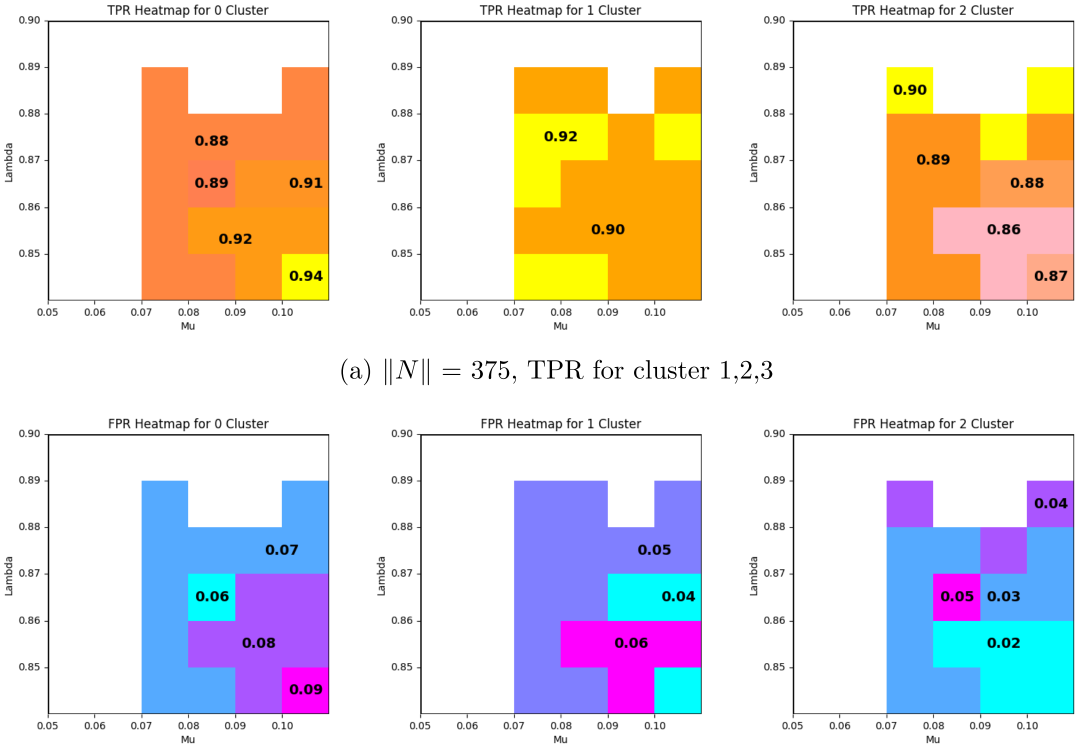
A Appendix

A.1 Evidence of time limit leading to infeasibility

Table 2: The TPR and FPR of each cluster when $\lambda = 0.89$ and $\mu = 0.08$ with time limit of 1000 seconds

Cluster	TPR	FPR	Selected prototype index in seed(150)
1	0.89	0.07	1046,6316,8761
2	0.90	0.05	1046,6316,8761
3	0.90	0.04	1046,6316,8761

As presented above, the TPR and FPR are present for all clusters when the time limit is set as 1000 seconds. Compared to the Figure below, the $\lambda = 0.89$ and $\mu = 0.08$ are missing in Cluster 1 and 3 when setting the time limit as 600 seconds.



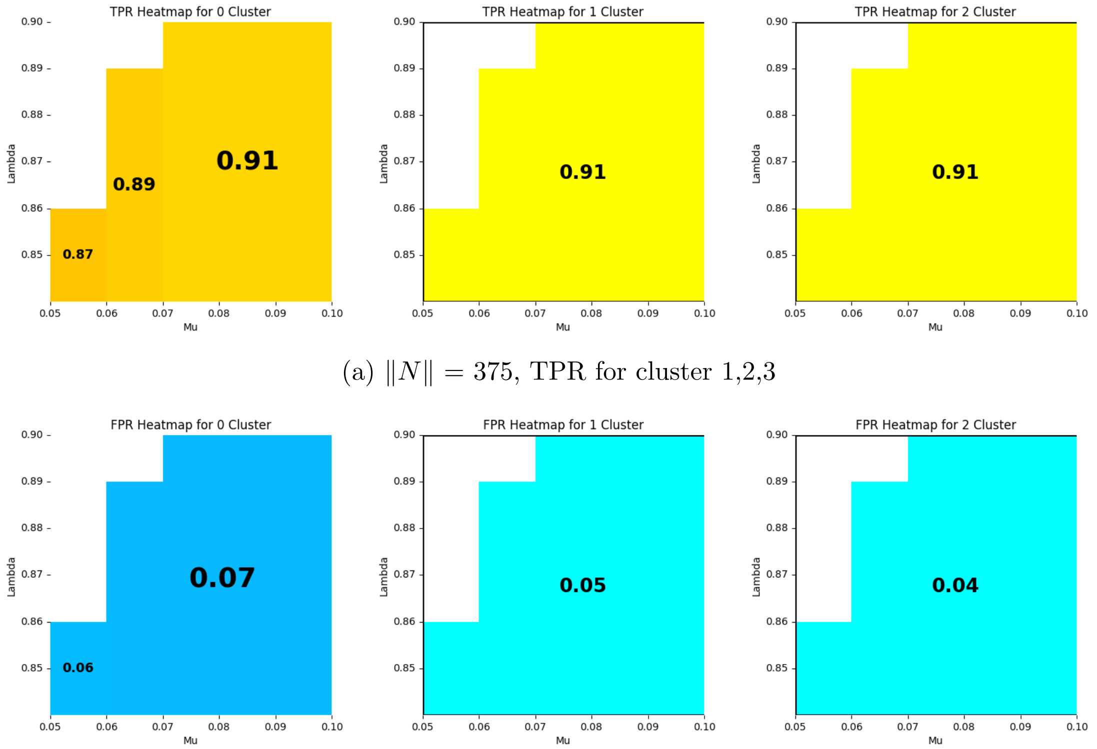
(a) $\|N\| = 375$, TPR for cluster 1,2,3

(b) $\|N\| = 375$, FPR for cluster 1,2,3

(c) Partitioning model with reduction technique. True Positive Rate and False Positive Rate in set covering with the increasing data size. λ and μ vary on the grid $[0.05, 0.1] \times [0.85, 0.9]$

A.2 Reduced model with $|N| = 10^4$

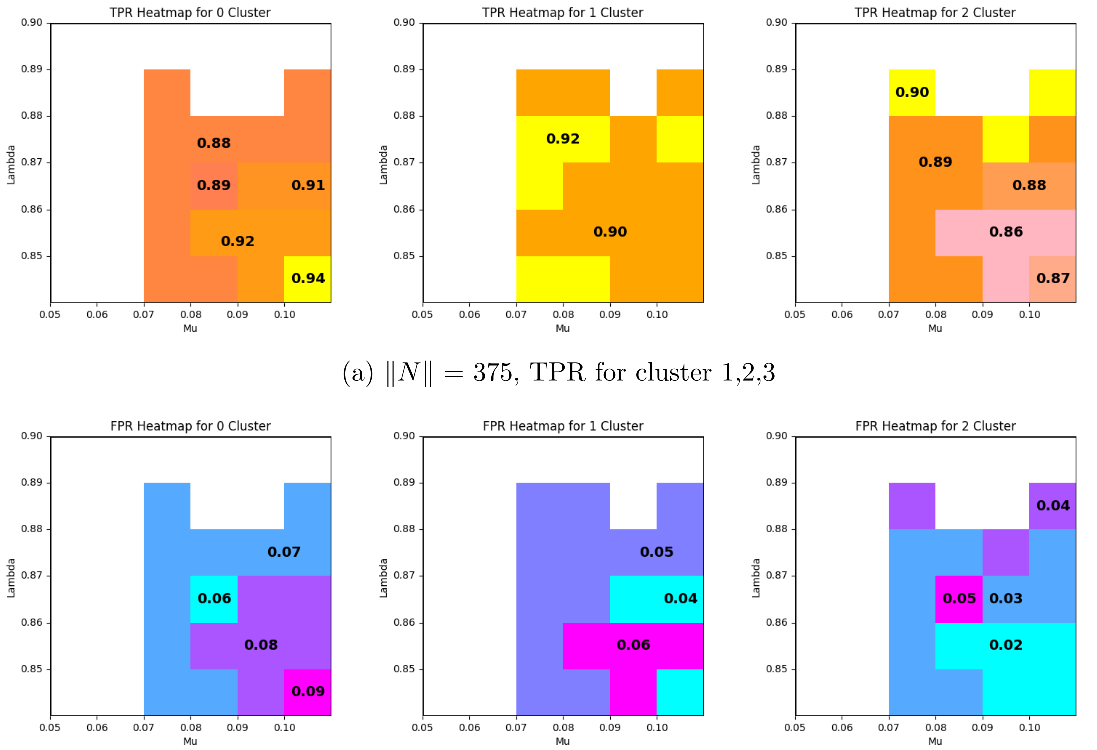
The attempt to solve a reduced model as described in Carrizosa et al. (2022) is presented below.



(a) $\|N\| = 375$, TPR for cluster 1,2,3

(b) $\|N\| = 375$, FPR for cluster 1,2,3

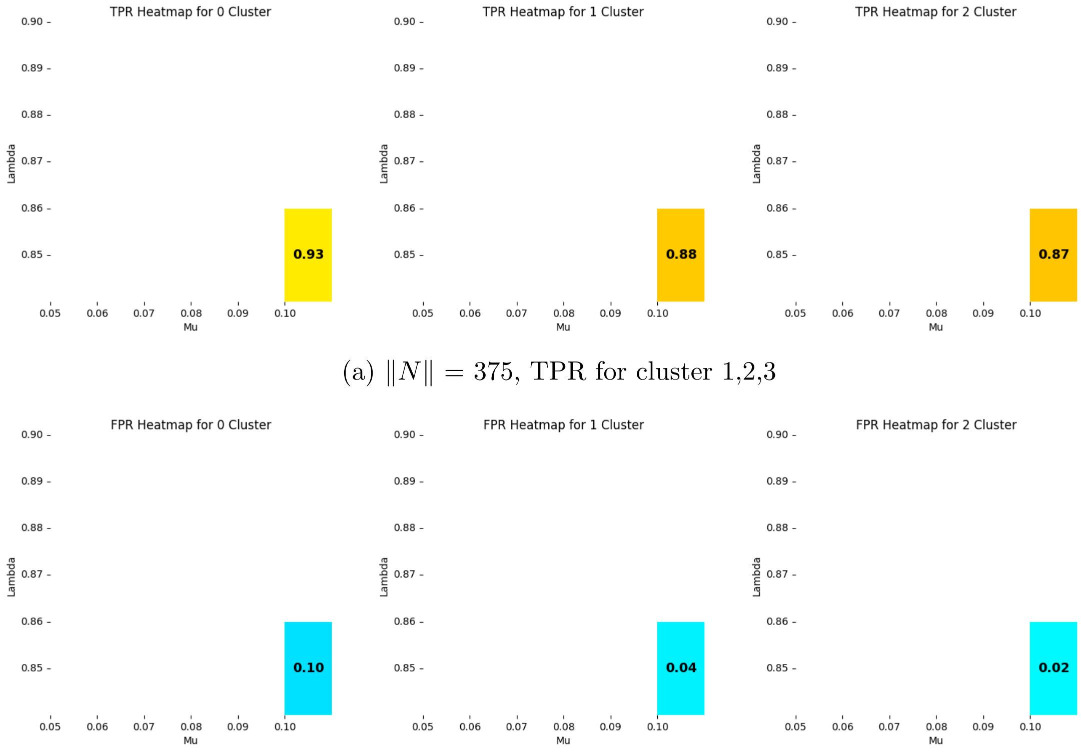
(c) Set covering model with reduction technique. True Positive Rate and False Positive Rate in set covering with the increasing data size. λ and μ vary on the grid $[0.05, 0.1] \times [0.85, 0.9]$



(a) $\|N\| = 375$, TPR for cluster 1,2,3

(b) $\|N\| = 375$, FPR for cluster 1,2,3

(c) Partitioning model with reduction technique. True Positive Rate and False Positive Rate in set covering with the increasing data size. λ and μ vary on the grid $[0.05, 0.1] \times [0.85, 0.9]$



(a) $\|N\| = 375$, TPR for cluster 1,2,3

(b) $\|N\| = 375$, FPR for cluster 1,2,3

(c) Gaussian Mixture Model. True Positive Rate and False Positive Rate in partitioning model with the increasing data size. λ and μ vary on the grid $[0.05, 0.1] \times [0.85, 0.9]$

B Programming code

Hardware information: PC Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz, 8GB of RAM.

Step 1: Download Pycharm version 2024.1.4 from `https://www.jetbrains.com/pycharm/download/?section=windows`.

Step 2: Download R version 4.3.3 from `https://cran.rstudio.com/`

Step 3: Download the IDE compatible with R version 4.3.3 called Rstudio from `https://posit.co/download/rstudio-desktop/`

Step 4: Download Rtools43 from `https://cran.rstudio.com/bin/windows/Rtools/rtools43/rtools.html`

Step 5: Run the code

Description of the code:

- **thesis(Rcode)**: extracts Weather data in R.
- **thesis_fig1.py**: produces Figure 1 to explain TPR and FPR
- **thesis_figure2.py**: produces Figure 2
- **heatmap.py**: produces heatmaps for all the csv filed results from Canadian Weather Data
- **heatmap_simulated.py**:produces heatmaps for all the csv filed results from simulated data

- **extension-GMM_partitioning_weather.py**: produces GMM without inverse normal transformation in weather data.
- **extension-GMM_partitioning_weather_with_Inv_Norm.py**: produces GMM with inverse normal transformation in weather data.
- **extension-GMM_with_small_dataset.py**: produces GMM results with $|N| = 300$
- **extension-simulation_partitioning.py**: produces GMM with $|N| = 10000$
- **extension_dissimilarity_matrix.py** verifies the memory error of 10000 × 10000 dissimilarity matrix
- **thesis_dissim_matrix.py**: produces dissimilarity matrix for each region in Canadian Weather data
- **thesis_simulated_datacollection.py**: produces simulated data scatter plot
- **trial_partitioning_simulation.py**: produces the output of reduced technique for partitioning model with $|N| = 10000$
- **trial_partitioning_simulation_with_small_dataset.py**: produces the output of reduced technique for set covering model with $|N| = 300$
- **trial_partitioning_simulation_with_small_dataset_reduced_model.py**: produces output with reduced technique for partitioning model with $|N| = 300$ and $|\tilde{N}| = 50$.
- **trial_partitioning_weather.py**: produces the output for weather data for partitioning model
- **trial_set_covering_simulation.py**: produces the output for simulated data for the set covering model for $|N| = 10000$
- **trial_set_covering_simulation_with_small_dataset.py**: produces the output of reduced technique for partitioning model with $|N| = 300$
- **trial_set_covering_simulation_with_small_dataset_reduced_model.py**:produces output with reduced technique for set covering model with $|N| = 300$ and $|\tilde{N}| = 50$.
- **trial_set_covering_weather.py**: produces the output for weather data for the set covering model
- **weather_data_analysis**: produces output of the Kolmogorov Smirnov test and fitter package in Python