Cluster-Weighted Model Based on TSNE Algorithm for High-Dimensional Data

Kehinde Olobatuyi[1]*, Matthew R. P. Parker[2], and Oludare Ariyo [3]

[1]Department of Mathematics and Statistics, University of Victoria, Canada

[2]Department of Statistics and Actuarial Sciences, Simon Fraser University, Canada

[3] Department of Mathematical Sciences, University of Essex, UK

Abstract

Cluster-weighted models (CWMs) are an important class of machine learning models that are commonly used for modelling complex datasets. However, they are known to suffer from reduced computing efficiency and estimator accuracy when dealing with high-dimensional data. Previous work has proposed a parsimonious technique that can improve CWMs' performance in the high-dimensional data paradigm. However, this method has a setback for very high-dimensional data, where the dimensionality is greater than 100. In this paper, we propose a new hybridised method that incorporates a dimensionality reduction technique called T-distributed stochastic neighbour embedding (TSNE) to enhance the parsimonious CWMs in high-dimensional space. Additionally, we introduce a novel heuristic for detecting the hidden components of the underlying mixture model, which can be used with the popular R package FlexCWM. We evaluated the performance of the proposed method using two real datasets and found that it improves clustering power when compared to both the parsimony methods and the TSNE methods combined with CWMs in the high-dimensional data setting. Our results suggest that the proposed method can improve the efficiency and accuracy of CWMs in dealing with high-dimensional data, making it a valuable tool for data scientists and statisticians.

Keywords: cluster-weighted model, expectation maximisation, FlexCWM, high dimensional data, parsimonious technique, T-distributed Stochastic Neighbor Embedding.

1 Introduction

Cluster-weighted models (CWMs; Gershenfeld, 1997) are a powerful inference algorithm for deriving versatile functional relationships between input and output data using a mixture of expert clusters. CWMs allow for an

*Electronic address: olobatuyikenny@uvic.ca; Corresponding author

output to be an entire probability distribution, which is an essential feature shared by several other modelling approaches, including varieties of CWMs (Ingrassia et al., 2012, 2014, 2015; Punzo, 2014). However, in this article, we focus on the CWM approach.

The CWM algorithm uses expectation-maximisation (EM; Dempster et al., 1977) to find the optimal locations of clusters in the input space and to solve for the parameters of the local model. As such, the CWM algorithm requires interactions between all the data and all the clusters. This means that as the quantity of data increases, the computing efficiency decreases. For example, when using CWMs for recreating the sound of a violin, model training could require a billion data points and a hundred thousand clusters; such an implementation would be impractical from a computing standpoint (Boyden, 1997). Therefore, more efficient computational methods are needed when dealing with high-dimensional data (HDD).

In the HDD setting, efficient dimensionality reduction is often used to uncover the hidden patterns of information in real data. Dimensionality reduction can convert data sets containing millions of functions into smaller, more practicable spaces for effective processing and analysis. Integrating dimensionality reduction techniques with statistical analysis can thus be used to improve computing efficiency when working with HDD (Rehman et al., 2016).

Many dimensionality reduction techniques, such as Principal component analysis (PCA: Alqahtani and Kalantan, 2020) and T-distributed Stochastic Neighbor Embedding (TSNE: Maaten and Hinton, 2008), have been developed in the statistical and artificial intelligence literature. In particular, there has been a recent rise in interest in PCA mixture models. Mixture models provide a framework for complex data using weighted component distributions. Due to these models' high flexibility and statistical efficiency, they are widely used in many fields, including machine learning, image processing, and data mining. Practical considerations have restricted the implementation of mixture models in high-dimensional spaces because the component distributions are formalised as probability density functions. PCA mixture models are a mixture-of-experts technique which models a nonlinear distribution through a combination of local linear sub-models with a sample distribution (Jin et al., 2004). For more detail on PCA mixture models, we recommend Kim et al. (2003); Xu et al. (2014); Kutluk et al. (2016).

One major setback of PCA is that it only creates linear combinations of existing features, so it fails to capture nonlinear aspects of the features. As such, PCA cannot interpret complex polynomial relationships between features. This causes PCA to perform poorly when the true relationship between features is nonlinear. Due to the complexity of HDD, nonlinear features are often present (making PCA inappropriate for many HDD tasks). In many HDD datasets, including one we consider in this paper, a preprocessing step of feature extraction is necessary to reduce the computational burden and time complexity before fitting the CWMs models. This has been a major setback for these CWMs. In this article, we propose to improve the efficiency of CWMs when clustering HDD and reduce the computational burden involved. The contributions of this paper are, therefore, to (i) improve the efficiency of CWMs in clustering HDD, (ii) show that parsimonious techniques on CWMs could lead to local maxima and selection of an incorrect number of clusters for HDD, (iii) propose the use of TSNE as a means of dimension reduction for CWMs; and (iv) show improvements to clustering power when comparing the parsimony methods with the TSNE methods combined with CWMs and HDD.

The plan of the paper is as follows. Section 2 gives an overview of CWMs and Section 3 discusses the T-distributed Stochastic Neighbour Embedding (TSNE) Techniques as related to dimension reduction. We incorporate the TSNE techniques to CWMs for HDD in Section 4; at the same time, we also discuss the EM algorithim applied to CWMs and Geometrically Constrained CWMs in this section. Section 5 discusses the application of the proposed method to real-life datasets. In Section 6, we discussed the major outcomes of our paper. The concluding remarks are given in Section 7.

2 Cluster-Weighted Models

In this article, we present an overview of Cluster-weighted models (CWMs: Gershenfeld, 1997). The CWMs are models that involve a vector of covariates $\boldsymbol{X}$ and a response variable Y, both defined on a domain $\mathcal{D}$ with a joint probability of $(\boldsymbol{X}', Y)'$ represented as a convex combination $p(\boldsymbol{x}, y)$. The input vector $\boldsymbol{X}$ is a d-dimensional vector with values in a subspace $\mathcal{X} \subseteq \mathrm{R}^d$, while the response variable Y has values in a subspace $\mathcal{Y} \subseteq \mathrm{R}$. The set of model parameters is denoted as $\boldsymbol{\Theta} = (\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi)$, where G is the number of groups. The model parameters consist of $\boldsymbol{\beta} \in \mathrm{R}^{d \times G}$, which represents the weight of the local model; $\boldsymbol{\mu} \in \mathrm{R}^{d \times G}$, which represents the location parameter; $\boldsymbol{\Sigma}$, which is the positive definite covariance matrix; and $\boldsymbol{\pi}$, which represents the mixing distribution subject to the constraints $\sum_g \pi_g = 1$ and $\pi_g > 0$. Typically, CWMs are expressed as a sum of clusters,

$$p(\boldsymbol{x}, y) = \sum_{g=1}^{G} p(y|\boldsymbol{x}, \mathcal{D}_g) p(\boldsymbol{x}|\mathcal{D}_g) \pi_g, \tag{1}$$

where g represents the clusters. The total number of clusters G must be selected beforehand and can be determined using information criteria. The cluster weight $\pi_g \in [0, 1]$ denotes the amount of data described by the cluster g. The density $p(\boldsymbol{x}|\mathcal{D}_g)$ describes the domain of influence of cluster g, that is, the distribution of inputs $\boldsymbol{x}$ around the cluster. They are multivariate normal random variables, i.e.

$$p(\boldsymbol{x}|\mathcal{D}_g) \sim \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g),$$

with mean $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$, effectively describing the location and the range of cluster influence. When working in the high dimensional spaces, it is computationally efficient to reduce these inputs by separable Gaussian, with a diagonal matrix of single variances in each dimension, i.e. $\boldsymbol{\Sigma}_g = \mathrm{diag}\ (\sigma_{g,1}, ..., \sigma_{g,d})$. Similarly, the density $p(y|\boldsymbol{x}, \mathcal{D}_g)$ is the conditional density of the outputs y has given the inputs $\boldsymbol{x}$ around the cluster g. The presence of the conditional distribution allows the input vector $\boldsymbol{x}$ to relate with the target variable y. In general, $p(y|\boldsymbol{x}, \mathcal{D}_g)$ are modelled using Gaussian distributions

$$p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_g) \sim \mathcal{N}(f(\boldsymbol{x}, \boldsymbol{\beta}_g), \sigma_g).$$

The mean $f(\boldsymbol{x}, \boldsymbol{\beta}_g)$ and variances σ_g^2 describe the local models and the error around the cluster g. The vector $\boldsymbol{\beta}_g$ denotes the coefficient of the local model or the weight of contribution associated with the input vector $\boldsymbol{x}$. The cluster functions are chosen based on the type of supervised learning (Regression or classification) we wish to do. Often, the cluster functions are chosen as linear combinations of basis functions $f(\boldsymbol{x}) = \boldsymbol{x}'\beta_g$ for $g = 1, \ldots, G$, where f is a link function. The model output of the CWM is therefore weighted averagely by the local functions $f(\boldsymbol{x}, \boldsymbol{\beta}_g)$. The Gaussian, which is the input densities $p(\boldsymbol{x}|\mathcal{D}_g)$, controls the behaviour of the local functions. Ingrassia et al. (2015) showed that Equation 1 is a general and flexible family of mixture models. In particular, they prove that, under suitable assumptions, if both $p(y|\boldsymbol{x}, \mathcal{D}_g)$ and $p(\boldsymbol{x}|\mathcal{D}_g)$ are Gaussian, then mixture of Gaussian distributions on $(\boldsymbol{X}', Y)'$, mixtures of linear Gaussian regressions, and mixtures of linear Gaussian regressions with concomitant variables, using $\boldsymbol{X}$ as the concomitant variable, can be considered as nested in the linear Gaussian CWM. Moreover, both $p(y|\boldsymbol{x}, \mathcal{D}_g)$ and $p(\boldsymbol{x}|\mathcal{D}_g)$ have been considered to be t-distributed, also considering the mixture of t distributions and mixtures of regression models with t errors as nested in the linear t-**CWM**. Subedi et al. (2013) addressed the problem of applicability of the CWM in high-dimensional $\boldsymbol{X}$-spaces by assuming latent factors for the covariates in each mixture component.

2.1 Geometrically Constrained CWMs

The full multivariate Gaussian for CWMs as discussed poses many problems for the estimation process. Some of these problems are due to high-dimensional space or large d and can cause matrix singularity issues,

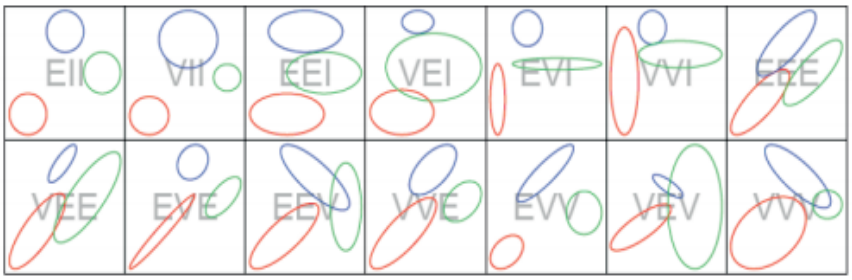

Figure 1: Models Used in CWMs clustering: Example of contours of the bivariates normal component densities for the 14 parameterisation of the covariance matrix. Source: Bouveyron et al. (2019)

leading to non-invertible matrices and failure of the algorithm. For a full covariance matrix, the number of parameters to be estimated is $(G-1)+Gd+G[d(d+1)/2]$. This is a large number of parameters. For example in the Epileptic Seizure data in Section 5.1, with $d=178$ and $G=5$, this is $128,879$ parameters to be estimated, which is too large for any clustering model. Such large numbers of parameters can lead to difficulties in estimation, including lack of precision or even cause the algorithm to degenerate. They also reduce the computational speed of the algorithms. In order to mitigate this problem, Banfield and Raftery (1993) and Celeux and Govaert (1995) introduced the eigenvalue decomposition of the cluster covariance matrix Σ_g, in the form

$$\mathbf{\Sigma}_g = \lambda_g D_g A_g D_g^T,$$

where D_g is the matrix of the eigenvectors of $\mathbf{\Sigma}_g$, $A_g = \text{diag}\{A_{1,g}, ..., A_{d,g}\}$ is a diagonal matrix whose elements are proportional to the eigenvalues of $\mathbf{\Sigma}_g$ arranged in a descending order, and λ_g is the constant associated with the proportionality. Each element in this decomposition corresponds to a particular geometric property of the gth component as follows: the matrix of the eigenvectors D_g determines its orientation in R^d; the diagonal matrix of scaled eigenvalues A_g governs its shape; λ_g determines the volume; and the region where the gth component is densely concentrated can be determined by the maximum number of the shape in the plane. For example, if $A_{1,g} \gg A_{2,g}$, then the gth component is tightly concentrated around a line in R^d. If $A_{1,g} \approx A_{2,g} \gg A_{3,g}$, then the gth component is concentrated in a two-dimensional plane in R^d. If all the values of $A_{j,g}$ are approximately equal, then the gth component is roughly equal. The volume is proportional to $\lambda_g^d |A_g|$ where $|A_g|$ is the determinant of A_g preferably constrained to be equal to 1 to avoid non-invertible or singularity problem.

There are several ways in which parsimony can be imposed in this framework. Using the eigenvalue decomposition, each of volume, shape, and orientation can be individually constrained to be equal across the clusters rather than varying. These constraints serve to reduce the number of parameters which need to be estimated. Additionally, the covariance matrix can be forced to be spherical, i.e. $\Sigma = I$, where I is the Identity matrix. Whenever the covariance matrix is spherical, there are two possible models in the univariate case and 14 possible models in the multivariate case. Figure 1 shows examples of all fourteen possible contours of the component densities for the various models in the two-dimensional case with two mixture components.

Table 1 shows the multivariate models denoted by three-letter identifiers where "E" stands for equal and "V" stands for a variable. If the first letter is "E" it means the volume is equal/constant across the clusters, and "V" if varied across. In the same vein, the second letter "E" represents an equal shape and "V" if not, so that for all $g \in \{1, \ldots, G\}$, the shape matrices $A_g \equiv A$. "I" stands for spherical when the $A_g = I$ for $g \in \{1, \ldots, G\}$. Finally, if "E" is located at the third position, then the D_g of eigenvectors specify the cluster orientations are equal $D_g \equiv D$ for $g = 1, \ldots, G$, "V" if they are not constrained, and "I" if the clusters are spherical such that $D_g = I$ for $g \in \{1, \ldots, G\}$.

Table 1: Parameterisations of the covariance matrix $\mathbf{\Sigma}_g$ through Eigenvalue decomposition. A denotes a diagonal matrix

Identifier	**Model**	**Distribution**	**Volume**	**Shape**	**Orientation**
E	–	Univariate	Equal	Not required	Not required
V	–	Univariate	Variable	Not required	Not required
EII	λI	Spherical	Equal	Equal	Not required
VII	$\lambda_g I$	Spherical	Variable	Equal	Not required
EEI	λA	Diagonal	Equal	Equal	Axis-aligned
VEI	$\lambda_g A$	Diagonal	Variable	Equal	Axis-aligned
EVI	λA_g	Diagonal	Equal	Variable	Axis-aligned
VVI	$\lambda_g A_g$	Diagonal	Variable	Variable	Axis-aligned
EEE	Σ	Ellipsoidal	Equal	Equal	Equal
VEE	$\lambda_g DAD^T$	Ellipsoidal	Variable	Equal	Equal
EVE	$\lambda DA_g D^T$	Ellipsoidal	Equal	Variable	Equal
EEV	$\lambda D_g A D_g^T$	Ellipsoidal	Equal	Equal	Variable
VVE	$\lambda_g D A_g D^T$	Ellipsoidal	Variable	Variable	Equal
VEV	$\lambda_g D_g A D_g^T$	Ellipsoidal	Variable	Equal	Variable
EVV	$\lambda D_g A_g D_g^T$	Ellipsoidal	Equal	Variable	Variable
VVV	Σ_g	Ellipsoidal	Variable	Variable	Variable

3 CWMs-TSNE for High-dimensional data

The stochastic neighbour embedding (SNE) was first introduced in a paper by (Hinton and Roweis, 2002). Its aim is to place objects in a low-dimensional space while retaining their neighbouring identity. It can also be extended to allow multiple low-dimensional images of each object. SNE is a dimensionality reduction technique that can construct reasonable visualisations, but its complex cost function makes it difficult to optimise. To address this issue, Maaten and Hinton (2008) introduced a variation of SNE called t-distributed stochastic neighbour embedding (TSNE). TSNE transforms a high-dimensional dataset $\boldsymbol{X} = (x_1, ..., x_n)$ into a low-dimensional data set $\boldsymbol{U} = (u_1, ..., u_n)$, which is much easier to optimise and provides significantly better visualisation by reducing the tendency to crowd points together in the centre of the map. In the context of CWMs, we can consider $\boldsymbol{X}$ as discussed in Section 2. TSNE employs a symmetric version of SNE as an alternative to mitigate the problem of the presence of outliers. In SNE, the pairwise similarities in the high-dimensional space p_{ij} are defined by asymmetric mapping, while in TSNE, a Student t-distribution with a degree of freedom $v = 1$ is applied to assess the similarity between points in the low-dimensional space. The joint probabilities for the low-dimensional map q_{ij} become a function of the distance between points in $\boldsymbol{U}$. The ultimate goal of TSNE is to represent p_{ij} by q_{ij} as accurately as possible, so the cost function C is the Kullback-Leibler divergence between P and Q. The gradient descent method is commonly used to minimise C, and the gradient of C is given by a resultant force pulling u_i in the direction of u_j or pushing it away depending on whether j is observed as a neighbour of i. The gradient descent is initialised by sampling the map point $\boldsymbol{U}^{(0)}$ randomly from a normal distribution. A momentum is added to the gradient descent to speed up the optimisation and avoid being stuck in local optima. The gradient update is calculated recursively, with the solution at each iteration being updated based on the gradient, momentum, and learning rate. The asymmetric mapping used in SNE is given as follows:

$$q_{ij} = \frac{\exp(-||u_i - u_j||^2)}{\sum_{k \neq i} \exp(-||u_k - u_j||^2)},$$

where q_{ij} is the pairwise similarities in the low-dimensional map and the way to define the pairwise similarities in the high-dimensional space p_{ij} is given by

$$p_{ij} = \frac{\exp(-||x_i - x_j||^2)/2\sigma^2}{\sum_{k \neq i} \exp(-||x_k - x_j||^2/2\sigma^2)}.$$

where σ is the scale parameter of the distance between x_i and x_j. These equations are referred to as symmetric because it has $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$ for $\forall i, j$. The joint probabilities for the low-dimensional map q_{ij} become

$$q_{ij} = \frac{\big(1 + ||u_i - u_j||^2\big)^{-1}}{\sum_{k \neq i} \big(1 + ||u_i - u_j||^2\big)^{-1}}.$$

The advantage of employing a Student t-distribution can be found in Maaten and Hinton (2008). The ultimate goal of TSNE is to represent p_{ij} by q_{ij} as accurately as possible, so the cost function C is given by

$$C = \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

The gradient descent method is commonly used to minimise C, and the gradient of C is given by

$$\frac{\partial C}{\partial u_i} = 4 \sum_j (p_{ij} - q_{ij})(u_i - u_j)\big(1 + ||u_i - u_j||^2\big)^{-1}. \tag{2}$$

Equation (2) can be interpreted as the summation of a resultant force pulling u_i in the direction of u_j or pushing it away depending on whether j is observed as a neighbour of i. The gradient descent is initialised by sampling the map point $\mathrm{U}^{(0)} = (u_1, ...u_n)$ randomly from $\mathcal{N}(0, 10^{-4}I)$, and set $\mathrm{U}^{(-1)} = 0$. A momentum is added to the gradient descent to speed up the optimisation and avoid being stuck in local optima. The gradient update is calculated recursively by

$$\boldsymbol{U}^{(t)} = \boldsymbol{U}^{(t-1)} + \zeta \frac{\partial C}{\partial \boldsymbol{U}} + \alpha(t)(\boldsymbol{U}^{(t-1)} - \boldsymbol{U}^{(t-2)}).$$

where $\boldsymbol{U}^{(t)}$ is the solution at iteration t, ζ is the learning rate, and the $\alpha(t)$ is the momentum at iteration t. Then, Equation 1 becomes

$$p(\boldsymbol{u}, y) = \sum_{g=1}^{G} p(y|\boldsymbol{u}, \mathcal{D}_g) p(\boldsymbol{u}|\mathcal{D}_g) \pi_g. \tag{3}$$

CWM models which use the various covariance matrix constraints will have far fewer parameters to estimate when compared with the full multivariate CWMs. An additional benefit of constrained models is that they can yield more precise estimates of the model parameters, accurate out-of-sample predictions, and more readily interpreted parameter estimates. Aside from the covariance matrix parameters, these models also have Gd parameters for the component means $\boldsymbol{\mu}_g$, and $(G-1)$ parameters for the mixture proportions π_g. Parsimony can help reduce the number of parameters to be estimated and thus make the model more computationally tractable. However, it is not always sufficient to achieve a good fit for the data, and in some cases, the number of parameters in a parsimonious model can still be quite large. In these cases, further parsimonious methods may be needed to reduce the number of parameters even further. However, this can be challenging from a computational standpoint, as these methods often require additional computations that can be time-consuming. In the case of the VVV model, the combination of dimensionality reduction and eigenvalue decomposition can provide significant gains in terms of both parsimony and computational

Table 2: Numbers of the parameters needed to specify the covariance matrix for models used CWMs and CWMs-TSNE, d is the data dimension, and G is the number of components.

Model	General	$d = 3, G = 5$	$d = 178, G = 5$
E	–	–	–
V	–	–	–
EII	1	1	1
VII	G	5	5
EEI	d	3	178
VEI	$G + (d - 1)$	7	182
EVI	$1 + G(d - 1)$	11	886
VVI	Gd	15	890
EEE	$d(d + 1)/2$	6	15931
VEE	$G + (d + 2)(d - 1)/2$	10	15935
EVE	$1 + (d + 2G)(d - 1)/2$	14	16639
EEV	$1 + (d - 1) + G[d(d - 1)/2]$	18	78943
VVE	$G + (d + 2G)(d - 1)/2$	18	16643
VEV	$G + (d - 1) + G[d(d - 1)/2]$	22	78947
EVV	$1 + G(d + 2)(d - 1)/2$	26	79651
VVV	$G[d(d + 1)/2]$	30	79655

efficiency, while still allowing for a good fit to the data. This is because the covariance matrices can be represented using a much smaller number of parameters, while still capturing the essential structure of the data. Before performing the dimensionality reduction, we note that parsimonious CWM is impracticable. For example, Table 2 shows the number of parameters needed to specify the covariance matrix for each model in the 178-dimensional five-component case, $d = 178, G = 5$, three-dimensional five-component case, $d = 3, G = 5$ gotten from the Epileptic seizure recognition data before dimensionality reduction and after dimensionality reduction, respectively. These results are obtained by noting that for one mixture component, the volume is specified by 1 parameter, the shape by $(d - 1)$ parameters, and the orientation by $d(d - 1)/2$. The potential gain in the combination of parsimony and dimensional reduction is far higher than the gain from only parsimony compared to the full covariance matrix parameters. In the most extreme case in Table 2, in the 178-dimensional case with 5 mixture components, the VVV model requires $79,655$ parameters to represent the covariance matrices, whereas the same VVV requires 30 parameters with the combination of dimensionality reduction and eigenvalue decomposition. Parsimony can offer some gains in a model fit; however, the most parsimonious models do not always fit the data adequately. Moreover, the number of parameters to be estimated in parsimonious models can still be outrageously high. One solution would be to apply further parsimonious methods to the results of the parsimonious model. However, this might not be achievable if computational time is a priority.

This leads us to a new solution for CWMs on high dimensional data: performing dimensionality reduction before using parsimony. Unfortunately, the Eigenvalue decomposition method does what we can call a "local parameter reduction" when the "global feature" remains large; consequently reducing the classification power, slowing the computation speed, and leading to potential misinterpretation of the results. Instead, we use TSNE for dimensionality reduction. Thereafter, we maximise Equation 3 for high-dimensional data sets using the EM algorithm.

4 EM algorithm applied to CWMs-TSNE.

In the case of CWMs, as described in Section 2, the likelihood becomes easier when a latent variable is introduced to model the unobserved data. The latent variable can be imagined as sampling each pair $(\boldsymbol{u}_i, y_i)$ from a single cluster with some probability.

Let $(\boldsymbol{u}_1', y_1)', \ldots, (\boldsymbol{u}_n', y_n)'$ be a sample of n independent observation pairs drawn from the model defined in Equation 3. The corresponding likelihood, for a fixed number of components G, is given by

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} p(\boldsymbol{u}_i, y_i; \boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{g=1}^{G} \pi_g p(y_i|\boldsymbol{u}_i; \boldsymbol{\beta}_g, \sigma_g) \phi(\boldsymbol{u}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g).$$

Define $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iG})'$, with $z_{ig} = 1$ if $(\boldsymbol{u}_i', y_i)'$ comes from $\mathcal{D}_g$, and $z_{ig} = 0$ otherwise, and consider the complete data $\{(\boldsymbol{u}_i', y_i, z_i')'; i = 1, \ldots, n\}$. Then, the complete-data likelihood can be written as

$$L_c(\boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \left[p(y_i|\boldsymbol{u}_i; \boldsymbol{\beta_g}, \sigma_g) \phi(\boldsymbol{u}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \pi_g \right]^{z_{ig}}. \tag{4}$$

The corresponding complete-data log-likelihood, the logarithm of Equation 4, can be written as

$$l_c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \big[\ln p(y_i|\boldsymbol{u}_i; \boldsymbol{\beta}_g, \sigma_g) + \ln \phi(\boldsymbol{u}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) + \ln \pi_g \big]$$

$$= \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \ln p(y_i|\boldsymbol{u}_i; \boldsymbol{\beta}_g, \sigma_g) + \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \ln \phi(\boldsymbol{u}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) + \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \ln \pi_g. \tag{5}$$

Maximisation of $L(\boldsymbol{\theta})$, through $L_c(\boldsymbol{\theta})$, is hereby achieved by the EM algorithm (Dempster et al., 1977). Each iteration of the EM algorithm alternates between two steps, the E-step (expectation) and the M-step (maximisation) below.

4.1 E-step

During the $(k+1)$th iteration of the algorithm, known as the E-step, where k ranges from 0 to some maximum value until convergence, the expectation of $l_c(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ is required. In the expression for $l_c(\boldsymbol{\theta})$, the unobservable data z_{ig} appears in a linear fashion. Thus, to perform the E-step, it is sufficient to compute the conditional expectation of Z_{ig} given the observed sample in the current iteration. Here, Z_{ig} represents the random variable corresponding to z_{ig}. In other words, the E-step requires calculating the expected value of Z_{ig} for each observation in the sample, given the model parameters estimated in the previous iteration. In particular, the expectation of Z_{ig} given the observed sample for $i = 1, \ldots, n$ and $g = 1, \ldots, G$ on the $(k+1)th$ iteration can be expressed as:

$$\mathrm{E}_{\boldsymbol{\theta}^{(k)}}[Z_{ig}|(\boldsymbol{u}_i', y_i)'] = \tau_{ig}^{(k)} = \frac{p\Big(y_i|\boldsymbol{u}_i; \boldsymbol{\beta}_g^{(k)}, \sigma_g^{(k)}\Big) \phi\Big(\boldsymbol{u}_i; \boldsymbol{\mu}_g^{(k)}, \boldsymbol{\Sigma}_g^{(k)}\Big) \pi_g^{(k)}}{p\Big(\boldsymbol{u}_i, y_i; \boldsymbol{\theta}^{(k)}\Big)}. \tag{6}$$

Here, $\tau_{ig}^{(k)}$ represents the posterior probability that the unlabeled observation $(\boldsymbol{u}_i, y_i)'$ belongs to the gth component of the mixture using the current fit $\boldsymbol{\theta}^{(k)}$ for $\boldsymbol{\theta}$. The numerator consists of the product of the conditional probability of the response variable y_i given the covariate vector $\boldsymbol{u}_i$ and the parameters $\boldsymbol{\beta}_g^{(k)}$ and $\sigma_g^{(k)}$ of the gth cluster, the probability density function of the covariate vector $\boldsymbol{u}_i$ given the parameters $\boldsymbol{\mu}_g^{(k)}$ and $\boldsymbol{\Sigma}_g^{(k)}$ of the gth cluster, and the mixing weight $\pi_g^{(k)}$ of the gth cluster. The denominator is the marginal probability of $(\boldsymbol{u}_i, y_i)'$ given the current fit $\boldsymbol{\theta}^{(k)}$.

4.2 M-step

During the M-step of the algorithm, at the $(k+1)$-th iteration, we maximise the conditional expectation of $l_c(\boldsymbol{\theta})$ given the observed data, denoted as $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)})$, with respect to the parameters $\boldsymbol{\theta}$. To obtain the maximum likelihood estimates of the parameters $\boldsymbol{\theta}$, the value z_{ig} in Equation 5 is replaced with their current expectations $\tau_{ig}^{(k)}$, which were obtained in Equation 6. This leads to the following expression for $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) = \sum_{i=1}^{n}\sum_{g=1}^{G} \tau_{ig}^{(k)} \ln \pi_g + \sum_{i=1}^{n}\sum_{g=1}^{G} \tau_{ig}^{(k)} \ln p(y_i|\boldsymbol{u}_i; \boldsymbol{\beta}g, \sigma_g) + \sum_{i=1}^{n}\sum_{g=1}^{G} \tau_{ig}^{(k)} \ln \phi(\boldsymbol{u}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g). \tag{7}$$

The three terms on the right-hand side of Equation 7 have zero cross-derivatives and can be maximised separately. Let us define the parameter vectors $\boldsymbol{\pi} = \{\pi_g; g = 1, \ldots, G\}, \boldsymbol{\beta} = \{\boldsymbol{\beta}_g; g = 1, \ldots, G\}$, and $\boldsymbol{\sigma} = \{\sigma_g; g = 1, \ldots, G\}$. We will maximise each term with respect to the corresponding parameter vector while holding the other parameters fixed.

4.2.1 Mixture Weights

To maximise $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)})$ with respect to $\boldsymbol{\pi}$ while satisfying the parameter constraints, we optimise the augmented function:

$$\sum_{i=1}^{n}\sum_{g=1}^{G} \tau_{ig}^{(k)} \ln \pi_g - \psi\left(\sum_{g=1}^{G} \pi_g - 1\right), \tag{8}$$

where ψ is a Lagrangian multiplier. Solving for π_g requires taking the derivative of Equation 8 with respect to π_g, setting it to zero, and solving for π_g, which yields:

$$\pi_g^{(k+1)} = \psi_g^{(k)} \Big/ n,$$

where $\psi_g^{(k)} = \sum_{i=1}^{n} \tau_{ig}^{(k)}$.

4.2.2 Parameters Related to U

Maximising Equation 7 with respect to $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g = \lambda_g D_g A_g D_g'$, with $g = 1, \ldots, G$, is equivalent to independently maximising each of the G expressions

$$\sum_{i=1}^{n} \tau_{ig}^{(k)} \ln \phi(\boldsymbol{u}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$$

In particular, we obtain

$$\boldsymbol{\mu}_g^{(k+1)} = \frac{1}{n_g^{(k)}} \sum_{i=1}^{n} \tau_{ig}^{(k)} \boldsymbol{u}_i$$

and

$$\boldsymbol{\Sigma}_g^{(k+1)} = \frac{1}{n_g^{(k)}} \sum_{i=1}^{n} \tau_{ig}^{(k)} \left(\boldsymbol{u}_i - \boldsymbol{\mu}_i^{(k+1)}\right)\left(\boldsymbol{u}_i - \boldsymbol{\mu}_i^{(k+1)}\right).$$

4.2.3 Parameters Related to Y

Maximising Equation 7 with respect to $\boldsymbol{\beta}_g$ and σ_g is equivalent to independently maximising each of the G expressions

$$\sum_{i=1}^{n} \tau_{ig}^{(k)} \ln p(y_i|\boldsymbol{u}_i; \boldsymbol{\beta}_g, \sigma_g). \tag{9}$$

Maximising Equation 9 is equivalent to the maximisation problem of the complete data. Maximisation of Equation 9 with respect to $\boldsymbol{\beta}_g$ can be carried out numerically.

5 Application to real-life datasets

This section illustrates some real data applications of the linear CWMs defined above with a substantive high dimensionality. The analysis is performed using the **R** package for CWMs called **FlexCWM** (Mazza et al., 2018).

5.1 Epileptic Seizure Recognition

In this paper, we present the methodology by analysing the Epileptic Seizure recognition data obtained from the UCI data set. The dataset corresponds to a brain activity recording spanning a duration of 23.5 seconds. To capture the temporal information, the corresponding time series is sampled into 4094 data points. Each data point represents the value of the EEG recording at a specific time instance. The dataset consists of a total of 500 individuals, and for each individual, there are 4094 data points collected over 23.5 seconds. To facilitate further analysis, these 4094 data points are divided into 23 chunks, with each chunk containing 178 data points. Thus, each chunk represents a 1-second segment of the recording. The division is done by shuffling the data points to ensure randomness. Consequently, we obtain a tabular representation of the data with 11500 rows and 178 columns. Each row corresponds to a chunk of 178 data points, and the final column represents the class label. The class labels are assigned values from the set 1, 2, 3, 4, 5, indicating different classes or categories associated with the epileptic seizure recognition task.

The objective of this case study is to identify the underlying components within the data. Previous studies have approached this data by treating it as a binary classification problem. In this context, class 1 indicates the presence of a seizure in a patient, while classes 2, 3, 4, and 5 represent the absence of a seizure.

The CWMs employ the Ordinary Least Squares (OLS) method for the maximisation step of the Expectation-Maximisation (EM) algorithm. However, OLS is not suitable for fitting a categorical dependent variable. As an alternative approach, we transform the label class by taking the logarithm and adding a small constant value, such as 0.5, to convert it into a continuous variable. Furthermore, we perform dimensionality reduction on the independent variable of order 178 using TSNE. It's important to note that TSNE is not primarily intended for clustering purposes. In Figure 2, we visualise the high-dimensional data on a 2D plane using TSNE with a perplexity value of 15 (representing the number of nearest neighbours), 1000 iterations, and a theta value of 0.5 (determining the speed/accuracy trade-off). The plot depicted in Figure 2 reveals a linear pattern discovered by TSNE only. We note in Figure 2 that when the perplexity ranges from 9 to 15 and theta is set to 0.5, TSNE may provide insufficient information about the low-dimensional data, resulting in what is known as a "crowd point". However, due to the high volume of data, TSNE tends to be slower when applied to datasets with significantly high dimensionality. There is a trade-off between speed and accuracy in the TSNE process, preserving the hidden structure of the high-dimensional data in the low-dimensional space. Nevertheless, clustering the epileptic seizure data is challenging due to substantial overlap. It is worth noting that one limitation of the TSNE output in Figure 2 is that the information criteria tend to favour the number of label classes, which contradicts previous works that focused on binary classification, where

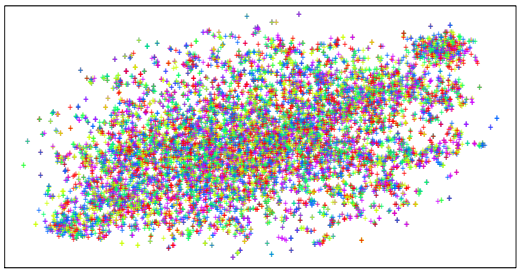

Figure 2: The TSNE for dimensionality reduction of the Epileptic Seizure data for 1000 iterations, perplexity = 15 and theta = 0.5.

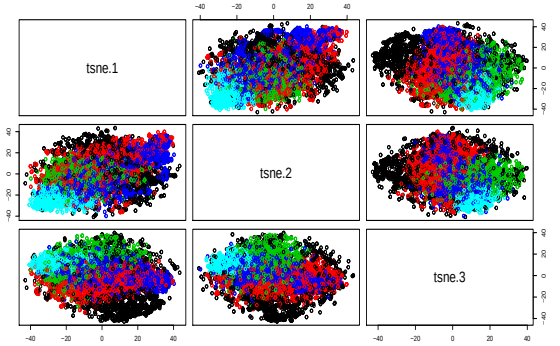

Figure 3: The CWM-TSNE plot for clustering the low-dimensional data produced by TSNE for Seizure recognition data with five categories.

class 1 represented the presence of epileptic seizures in patients against their absence. Figure 3 displays a five-component structure obtained from the CWMs model on the low-dimensional data filtered through the CWMs model. The majority of the information criteria favour a model with five mixture components. Despite visualising the high-dimensional data, the clusters are not well separated.

To mitigate the presence of crowd points in Figure 2, we conducted further comprehensive dimensionality reduction using different parameters for TSNE. We set the perplexity value to 250, theta to 15, and increased the number of iterations to 10,000. The output after 10,000 iterations is depicted in Figure 4. The plot in Figure 4 reveals the underlying structure after the extended iterations, although TSNE alone lacks the ability to effectively cluster the label class into two distinct classes. To address this, we employed CWM-TSNE, which utilised the output of TSNE to uncover the hidden categorical structure within the seizure data. The resulting plot of CWM-TSNE is presented in Figure 5. Since no single information criterion is perfect, we compared the Bayesian Information Criterion (BIC) and the Integrated Completed Likelihood (ICL) to evaluate the selected number of components. Generally, BIC and ICL agreed on the selection of the same number of components. The plot shown in Figure 5 corresponds to the model EEE, chosen by ICL. CWM-TSNE successfully identifies two distinct classes, albeit with some misclassifications. In Figure 5, class 1 represents the presence of an epileptic seizure, while class 2 represents the absence of an epileptic seizure. Regarding the number of components, BIC and ICL consistently selected the same number, except for one model where ICL chose 2 components instead of 3 for the EEE model. The comparison between BIC and ICL on the number of mixture components is shown in Figure 6 and Figure 7. The selection criteria values are provided in Table 3, with BIC on the left and ICL on the right. Additionally, Table 4 presents the Adjusted Rand Index (ARI) and its variants. The EVE model achieves the highest ARI values. However, the EEE model demonstrates a classification accuracy of 73%. Overall, the extended dimensionality reduction using modified TSNE parameters combined with CWM successfully revealed the underlying structure in the seizure data, providing insights into the classification of epileptic seizures.

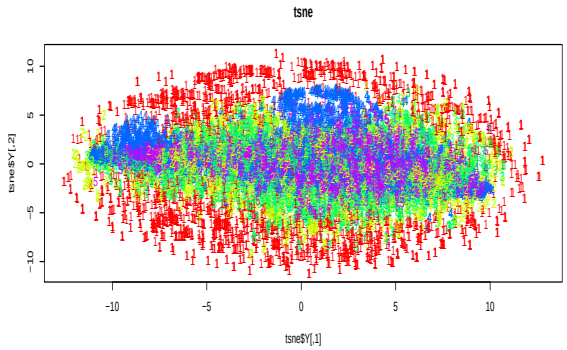

Figure 4: The TSNE for dimensionality reduction of the Epileptic Seizure data for 10,000 iterations, perplexity = 250, and theta = 0.5.

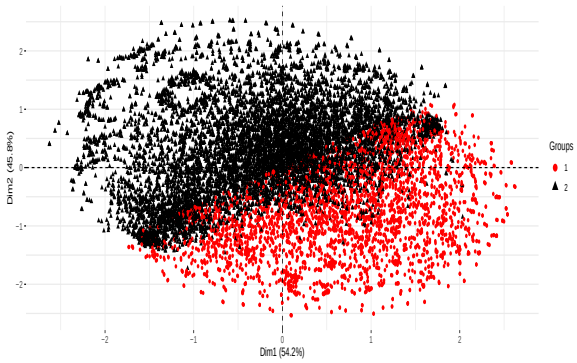

Figure 5: The CWM-TSNE plot for clustering the low-dimensional representation of Seizure recognition data produced by TSNE with EEE model.

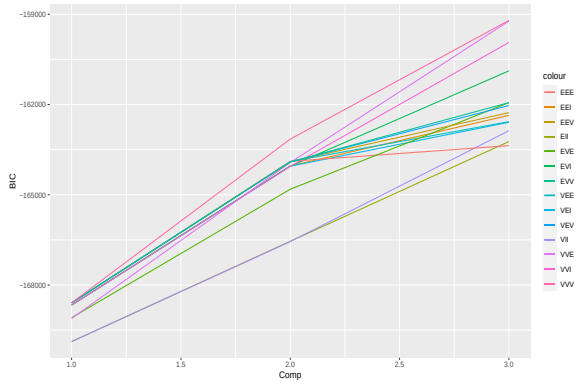

Figure 6: The Model selection of BIC for Seizure data among the fourteen parsimonious models; BIC selected the wrong number of mixture components when the true component according to the label is two-categorical.

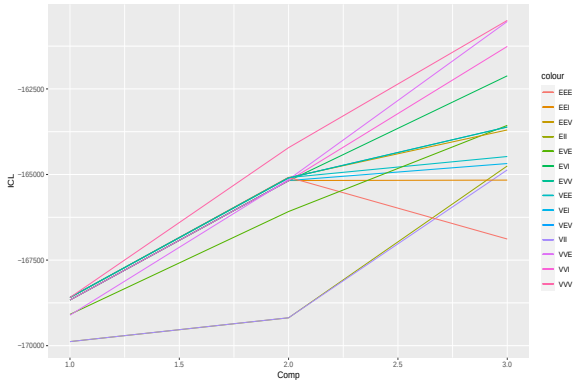

Figure 7: The Model selection of ICL for Seizure data among the fourteen parsimonious model; ICL selected EEE with the correct number of mixture component when the true component is two categories.

Table 3: The comparison of the BIC (on the left) and ICL (on the right) produced by the fourteen parsimonious models after performing the dimensionality reduction.

Model	comp1	comp2	comp3	comp1	comp2	comp3
EII	-169881	-166546	-163231	-169881	-169186	-164750
VII	-169881	-166556	-162870	-169881	-169185	-164866
EEI	-168667	-164041	-162360	-168667	-165176	-165164
VEI	-168667	-164050	-162591	-168667	-165176	-164680
EVI	-168667	-164050	-160881	-168667	-165185	-162118
VVI	-168667	-164059	-159935	-168667	-165183	-161259
EEE	-168593	-163889	-163367	-168593	-165081	-166885
VEE	-168593	-163898	-162569	-168593	-165090	-164475
EVE	-169086	-164816	-161938	-169086	-166081	-163565
EEV	-168593	-163898	-162271	-168593	-165090	-163699
VVE	-169109	-163927	-159226	-169109	-165145	-160531
VEV	-168593	-163908	-162034	-168593	-165099	-163612
EVV	-168593	-163908	-161937	-168593	-165103	-163613
VVV	-168593	-163149	-159204	-168593	-164215	-160499

Table 4: Adjustment Rand Index and its variants of the fourteen parsimonious models to select the hidden structure or cluster in the protein data

Model	Rand	HA	MA	FM	Jaccard
EII	0.557	0.153	0.153	0.618	0.432
VII	0.477	0.052	0.053	0.520	0.328
EEI	0.503	0.083	0.083	0.555	0.363
VEI	0.474	0.053	0.053	0.516	0.323
EVI	0.708	0.428	0.428	0.759	0.596
VVI	0.486	0.071	0.071	0.529	0.336
EEE	0.601	0.152	0.152	0.686	0.520
VEE	0.478	0.056	0.056	0.522	0.329
EVE	0.712	0.434	0.434	0.763	0.601
EEV	0.539	0.152	0.153	0.589	0.394
VVE	0.486	0.069	0.069	0.529	0.336
VEV	0.509	0.106	0.107	0.556	0.360
EVV	0.705	0.421	0.421	0.756	0.592
VVV	0.485	0.068	0.068	0.529	0.336

5.2 Protein data

The second application aims to cluster the proteins' localisation site. The protein data was created by Paul and Kenta (1996) and is available in the UCI database. The data consist of seven input variables and a class variable. There are $N = 336$ observations and attributes information is as follows;
Sequence Name: Accession number for the SWISS-PORT database, mcg: McGeoh's method for signal sequence recognition, gvh: Von Heijne's method for signal sequence recognition, lip: von Heijne's signal Peptidase II consensus sequence score, chg: Presence of charge on N-terminus of predicted lipoproteins, aac: Score of discriminant analysis of the amino acid content of outer membrane, alm1: Score of the ALOM membrane-spanning region prediction program, alm2: Score of ALOM program after excluding putative cleavable signal regions from the sequence. To fit the framework of CWMs, we transformed the multi-class localised site variable from categorical to continuous by adding 0.5 and taking the logarithm. We treat the true multi-class clustering as unknown apriori and fit many parsimonious models. We considered up to $G_{max} = 8$ components for each of the 14 possible covariance models; that is 8×14 competing models. The best fourteen parsimonious models were selected using BIC, one for each of the covariance models. Table 5 lists the values of the BIC for the fourteen models each with eight mixture components. We identified the smallest BIC across the parsimonious models (row values). Although VEV has the smallest BIC value of 6129.2, we chose model VVE as the best model then we chose the smallest BIC among the bold-faced numbers. Among the top 14 models considered, EII is the worst model (BIC -6309.6).

We compare the predicted classes for the 14 parsimonious models against the known classes in Table 6. The comparison was done using ARI and several varieties of ARI. The model VVE shows the highest values of ARI among all the models. According to the selection of the component produced by the CWM-TSNE model, Figure 8 shows the classification for the VVI selected model with respect to the number of clusters produced by the CWMs-TSNE. The protein data has been analysed by Paul and Kenta (1996), in which their model achieved 81% classification accuracy. Similar accuracy has also been achieved for Binary Decision trees and Bayesian Classification methods. Moreover, the classification accuracy for VVE is 87%.

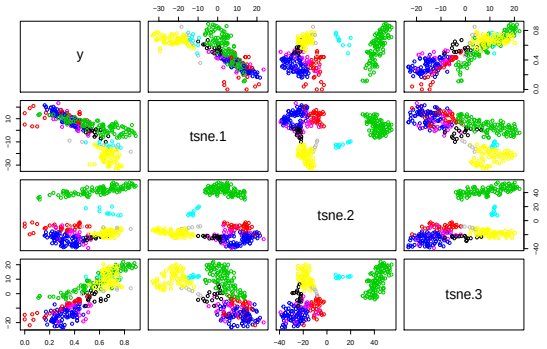

Figure 8: The plot produced by CWMs after dimension reduction via TSNE. CWMs selected eight components which align to the true class of the localisation site of protein.

Table 5: The comparison of the BIC produced by the fourteen parsimonious models after performing the dimensionality reduction

Model	comp1	comp2	comp3	comp4	comp5	comp6	comp7	comp8
EII	-8291.1	-7270.3	-7217.3	-6433.0	-6479.2	**-6309.6**	-6937.0	-6815.8
VII	-7960.9	-7223.6	-6565.9	-6474.5	-6336.2	-6394.1	-6337.5	**-6183.0**
EEI	-7960.9	-7422.0	-6577.6	-6437.5	-6389.8	-6453.8	**-6210.9**	-6213.5
VEI	-7960.9	-7086.6	-7262.7	-6411.8	-6404.0	-6304.2	**-6213.3**	-6346.9
EVI	-7960.9	-7196.4	-6566.5	-6644.1	-6440.1	-6361.3	**-6214.6**	-6230.0
VVI	-7960.9	-7223.6	-6565.9	-6474.5	-6336.2	-6394.1	-6337.5	**-6183.0**
EEE	-7412.5	-7005.6	-6460.8	-6327.7	-6284.3	**-6201.5**	-6295.4	-6214.2
VEE	-7412.5	-7143.0	-6457.2	-6413.9	-6219.8	-6222.3	-6215.4	**-6155.1**
EVE	-8182.5	-6997.2	-6452.2	-6192.0	**-6147.5**	-6168.1	-6186.7	-6227.5
EEV	-7412.5	-6745.1	-6429.6	-6431.5	-6180.8	-6182.5	**-6150.6**	-6247.5
VVE	-8189.6	-6994.1	-6954.8	-6176.8	**-6132.6**	-6222.1	-6185.7	-6198.5
VEV	-7412.5	-6941.4	-6503.3	-6199.0	-6174.0	-6174.7	-6324.7	**-6129.2**
EVV	-7412.5	-6733.9	-6473.3	-6229.9	-6242.5	-6241.3	**-6184.3**	-6299.9
VVV	-7412.5	-6822.0	-6412.8	-6363.1	**-6154.2**	-6183.6	-6241.4	-6192.4

Table 6: Adjusted Rand Index and its variants of the fourteen parsimonious models to select the hidden structure or cluster in the protein data

Model	Rand	HA	MA	FM	Jaccard
EII	0.821	0.478	0.483	0.603	0.411
VII	0.794	0.429	0.435	0.567	0.389
EEI	0.799	0.413	0.419	0.549	0.360
VEI	0.800	0.428	0.434	0.562	0.378
EVI	0.777	0.336	0.343	0.484	0.299
VVI	0.798	0.414	0.420	0.550	0.364
EEE	0.799	0.412	0.419	0.549	0.359
VEE	0.848	0.587	0.591	0.690	0.522
EVE	0.816	0.468	0.474	0.594	0.405
EEV	0.788	0.409	0.415	0.551	0.373
VVE	**0.866**	**0.646**	**0.649**	**0.737**	**0.581**
VEV	0.803	0.446	0.452	0.578	0.396
EVV	0.788	0.406	0.411	0.547	0.368
VVV	0.783	0.372	0.379	0.516	0.334

6 Discussion

This paper introduces a novel method that improves the performance of CWMs in classifying moderately high-dimensional and extremely high-dimensional data. Following the theoretical background of the CWMs (Ingrassia et al., 2014), we illustrate how CWMs metamorphosed from a finite mixture model (FMM). However, FMM is limited as it assumes independence, i.e. the assignment of the data points to the cluster has to be independent of the covariates (Hennig, 2000). On the contrary, CWMs assume random covariates with a parametric specification, allowing for assignment dependence. CWMs are helpful in model classifications; however, it has some limitations, which is the main motivation of this paper. One of the limitations of CWMs is the effect of the "curse of dimensionality". The dimensionality of the data hampers the clustering performance of CWMs. As such, the eigenvalue decomposition only slightly improves in the face of substantial high-dimensional data. For example, the seizure data of 178 dimensions has $128,879$ parameters to estimate. This may be impractical to attain in real-time when using CWMs, unlike RandomForest, which performs internal feature selection. However, using eigenvalue decomposition only partially solves the problems by slightly reducing the number of parameters to be estimated. In the presence of high-dimensional data with CMWs, degeneracies are inevitable; misinterpretation is bound to occur, and the computation time increases proportionally with the dimensionality of the data and low classification performance. An original CWMs fails to cluster image data with 178 dimensions in the example considered in this paper.

To mitigate these limitations in CWMs, we introduce CWMs based on TSNE (Maaten and Hinton, 2008) for high-dimensional data. We first performed a dimensionality reduction based on different parameters of **Rtsne** package in **R**. Our novel approach called CWMs-TSNE is applied to real high-dimensional Epileptic Seizure recognition data. The goal is primarily to detect the hidden mixture component different from the class labels. We investigated different perplexities and selected the one with a satisfactory low-dimensional output. At first, perplexities between 9 and 15 gave an unsatisfactory representation with "crowd points" presented in Figure 2. We further increased the perplexity to 250. This contradicts the authors' suggestion. However, the output gave a clear structure. Unfortunately, the output fails to reveal the hidden cluster of

epileptic patients even after $10,000$ iterations [Figure 4]. Afterwards, the output with the perplexity = 250 was filtered into the CWMs model. At this junction, we applied the 14 parsimonious models and observed a varying computation time due to their varying model complexities. The model selection was performed through eight different information criteria. We observed that the number of mixture components selected by BIC did not agree with ICL. While the BIC selected the models with the wrong number of components, ICL selected the model EEE with the correct number of hidden components. The output is provided in Figure (5). However, the overlap reduced drastically compared to Figure (4). The data we have used in this paper are categorical data with class labels of more than two classes. All the class labels are first transformed to be continuous variables. This is necessary because the linear Gaussian CWMs models use OLS for the maximisation step and can only efficiently handle a continuous dependent variable. The possible future direction should be to create self-sufficient CWMs by embedding a dimensionality reduction technique into the **CWMs** package in **R**. This will allow the package to handle high-dimensional data. Another useful exercise is to tackle the limitations of the family of CWMs and mitigate the effect of the 'curse of dimensionality' on CWMs by developing an appropriate model suitable for categorical data in high-dimensional space. This work is currently under review in other papers by the authors.

7 Conclusions

In this paper, a new hybridised method is proposed to enhance the performance of CWMs in classifying high-dimensional data. We describe the limitations of the existing CWMs model, including the curse of dimensionality, which hinders the clustering performance of CWMs. To address these limitations, we introduce CWMs-TSNE, a novel approach based on TSNE for high-dimensional data. We performed dimensionality reduction using different parameters and selected the one that yielded satisfactory results. CWMs-TSNE was applied to real high-dimensional Epileptic Seizure recognition data to detect the hidden mixture component different from the class labels. we observed a varying computation time due to the varying complexities of the 14 parsimonious models used. The model selection was performed through eight different information criteria, and we found that the number of mixture components selected by BIC did not agree with ICL. Finally, we suggest future directions for research, including embedding a dimensionality reduction technique into the CWMs package and developing an appropriate model suitable for categorical data in high-dimensional space. Our study is not without limitations, which should be taken into consideration when interpreting the findings. One limitation pertains to the methods employed in the analysis, specifically their applicability to categorical data. In our case studies, we encountered the need for additional data transformation to accommodate categorical variables, which introduces complexities and potential biases. Although the authors are actively addressing this limitation in parallel research endeavors, it remains an area of improvement for our current study.

Another limitation concerns the computational efficiency of t-distributed stochastic neighbor embedding (TSNE), a technique used for dimension reduction. We observed that TSNE can be computationally intensive, particularly when applied to high-dimensional datasets (HDD). This issue becomes more prominent in situations where there are no significant non-linear patterns or artifacts in the data. In such cases, an alternative approach like principal component analysis (PCA) may be more appropriate for dimension reduction, as it offers faster computation without sacrificing meaningful insights. Consideration of these limitations enhances the understanding and contextualization of our study's outcomes.

Acknowledgements

Kehinde Olobatuyi would like to acknowledge the Aspiration 2030 for providing Postdoctoral support for KO and Natural Sciences and Engineering Research Council of Canada (NSERC) for providing PGS-D support

for MP [funding reference number 569754]. Also, the authors would like to acknowledge the editor and anonymous reviewer(s) for their helpful feedback which greatly improved this manuscript.

Authors' Contribution

KO conceived and developed the theoretical formalism, performed the simulation studies, and the analysis of real-life datasets. Both MP and OA contributed to the research methodology and the final version of the manuscript. OA supervised the project. All authors read and agreed on the content of the manuscript.

Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this article. They affirm that the research was conducted in an unbiased manner and that the findings and conclusions presented in the manuscript are solely based on the scientific merit of the study. The authors have no financial or personal relationships that could potentially influence the interpretation or reporting of the research.

Data Availability Statement

The datasets used and/or analyzed during the current study are publicly available.

References

Alqahtani, N. A. and Kalantan, Z. I. (2020). Gaussian Mixture Models Based on Principal Components and Applications. *Mathematical Problems in Engineering*, 2020:e1202307. Publisher: Hindawi.

Banfield, J. D. and Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, 49:803–821.

Bouveyron, C., Celeux, G., Murphy, T. B., and Raftery, A. E. (2019). Model-based clustering and classification for data science with application in r. *Cambridge University Press, United Kingdom*, page 21.

Boyden, E. S. (1997). Tree-based Cluster Weighted Modeling: Towards A Massively Parallel Real-Time Digital Stradivarius. *Cambridge, MA: MIT Media Lab.*

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(25, 76, 171, 237, 248):781–793.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B.*

Gershenfeld, N. (1997). Nonlinear inference and cluster-weighted modeling. *Annals of the New York Academy of Sciences,*, 808(1):18–24.

Hennig, C. (2000). Identifiability of models for clusterwise linear regression. *Journal of classification*, 17(1):237–296.

Hinton, G. E. and Roweis, S. T. (2002). Stochastic neighbor embedding. *In Advances in Neural Information Processing Systems*, 15(2/3):833–840.

Ingrassia, S., Minotti, S., and Punzo, A. (2014). Model-based clustering via linear cluster-weighted models. *Computational Statistics and Data Analysis,*, 71(1):159–182.

Ingrassia, S., Minotti, S. C., and Vittadini, G. (2012). Local statistical modeling via a cluster-weighted approach with elliptical distributions. *Journal of Classification*, 39:363–401.

Ingrassia, S., Punzo, A., Vittadini, G., and Minotti, S. C. (2015). The generalized linear mixed cluster-weighted model. *Journal of Classification*, 32:85–113.

Jin, Z., Davoine, F., and Lou, Z. (2004). An effective em algorithm for pca mixture model. *in Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Springer, Lisbon, Portugal*, pages 626–634.

Kim, H. C., Kim, D., and Bang, S. Y. (2003). An efficient model order selection for pca mixture model. *Pattern Recognition Letters*, 24(9–10):1385–1393.

Kutluk, S., Kayabol, K., and Akan, A. (2016). Classification of hyperspectral images using mixture of probabilistic pca models. *in Proceedings of the 2016 24th European Signal Processing Conference (EUSIPCO), IEEE, Budapest, Hungary*, pages 1568–1572.

Maaten, L. V. and Hinton, G. E. (2008). Visualizing data using t-stochastic neighbor embedding. *Journal of Machine Learning Research*, 9:2579–2605.

Mazza, A., Punzo, A., and Ingrassia, S. (2018). flexcwm: A flexible framework for cluster-weighted models. journal of statistical. *Journal of Statistical Software*, 86(2):1–30.

Paul, H. and Kenta, N. (1996). A probablistic classification system for predicting the cellular localization sites of proteins. *Intelligent Systems in Molecular Biology, St. Louis, USA*, pages 109–115.

Punzo, A. (2014). Flexible Mixture Modeling with the Polynomial Gaussian Cluster-Weighted Model. *Statistical Modelling*, pages 203–231.

Rehman, M. H., Liew, C. S., Abbas, A., Jayaraman, P. P., Wah, T. Y., and Khan, S. U. (2016). Big data reduction methods: a survey. *Data Science and Engineering*, 1(4):265–284.

Subedi, S., Punzo, A., Ingrassia, S., and MCNicholas, P. D. (2013). Clustering and Classification Via Cluster-Weighted Factor Analyzers. *Advances in Data Analysis and Classification*, 7(1):5–49.

Xu, X., Xie, L., and Wang, S. (2014). Multimode process monitoring with pca mixture model. *Computers & Electrical Engineering*, 40(7):2101–2112.