Structured Mixture of Linear Mappings in High Dimension

Chun-Chen Tu, Florence Forbes, Benjamin Lemasson, Naisyin Wang

Structured Mixture of Linear Mappings in High Dimension †

Chun-Chen Tu
University of Michigan, Ann Arbor, USA
Florence Forbes
INRIA, Montbonnot, France
Benjamin Lemasson
Université Grenoble Alpes, Grenoble, France
and Naisyin Wang
University of Michigan, Ann Arbor, USA

Summary. When analyzing data with complex structures such as high dimensionality and non-linearity, one often needs sophisticated models to capture the intrinsic complexity. However, practical implementation using these models could be difficult. Striking a balance between parsimony and model flexibility is essential to tackle data complexity while maintaining feasibility and satisfactory prediction performances. In this work, we proposed the use of Structured Mixture of Gaussian Locally Linear Mapping (SMoGLLiM) when there is a need to use high-dimensional predictors to predict low-dimensional responses and there is a possibility that the underlying associations could be heterogeneous or non-linear. Besides using mixtures of linear associations to approximate non-linear patterns locally and using inverse regression to mitigate the complications due to high-dimensional predictors, SMoGLLiM also aims at achieving robustness by adopting cluster-size constraints and trimming abnormal samples. Its hierarchical structure enables covariance matrices and latent factors being shared across smaller clusters, which effectively reduce the number of parameters. An Expectation-Maximization (EM) algorithm is devised for parameter estimation and, with analytical solutions; the estimation process is computationally efficient. Numerical results obtained from three real-world datasets demonstrate the flexibility and ability of SMoGLLiM in accommodating complex data structure. They include using high-dimensional face images to predict the parameters under which the images were taken, predicting the sucrose levels by the high-dimensional hyperspectral measurements obtained from different types of orange juice and a magnetic resonance vascular fingerprinting (MRvF) study in which researchers are interested at using the so-called MRv fingerprints at voxel level to predict the microvascular properties in brain. The three datasets bear different features and presents different types of challenges. For example, the size of the MRv fingerprint dataset demands special consideration to reduce computational burden. With the hierarchical structure of SMoGLLiM, we are able to adopt parallel computing techniques to reduce the model building time by 97%. These examples illustrate the wide range of applicability of SMoGLLiM on handling different kinds of complex data structure.

Keywords: Expectation-Maximization, High dimension, Mixture of regressions, Magnetic resonance vascular fingerprinting, Robust regression

† *Address for correspondence*: Chun-Chen Tu, Department of Statistics, University of Michigan, 311 West Hall 1085 South University, Ann Arbor, MI 48109-1107, USA
E-mail: timtu@umich.edu

1. Introduction

Regression is a widely used statistical tool. A large number of applications consists of learning the association between responses and predictors. From such an association, different tasks, including prediction, can then be conducted. To go beyond simple linear models while maintaining tractability, non-linear mappings can be handled through transformation of the data into a so-called feature space using kernel methods (Elisseeff and Weston, 2001; Wu, 2012). Exploration of local linearity presents an alternative to model non-linear association. The non-linear relationship can be captured by a mixture of locally linear regression models. In general, conventional mixture of regressions (De Veaux, 1989; Frhwirth-Schnatter, 2006; Goldfeld and Quandt, 1973) are inadequate for regression because they assume assignment independence (Hennig, 2000). This means that the assignments to each of the regression components are independent of the covariate values. In contrast, when a mixture of locally linear models is considered, one can let the membership indicator of the mixture component depend on the values of the covariates. Consequently, when extended with assignment dependence, models in the family of mixtures of regressions are more likely to be suitable for regression applications. This is the case of the so-called Gaussian Locally Linear Mapping (GLLiM) model (Deleforge et al., 2015) that assumes Gaussian noise models and is, in its unconstrained version, equivalent to a joint Gaussian Mixture Model (GMM) on both responses and covariates. GLLiM includes a number of other models in the literature. It may be viewed as an affine instance of mixture of experts as formulated in (Xu et al., 1994) or as a Gaussian cluster-weighted (CW) model (Gershenfeld, 1997) except that the response variable can be multivariate in GLLiM while only scalar in CW models.

The modeling and computational flexibility of Gaussian mixture models makes it possible to model a rich class of relationships while providing a simple mathematical form for inference. However, when the number of covariates is large, estimating a function defined over a large number of variables is generally difficult because standard regression methods require the estimation of a large number of parameters, which in turns requires an even larger number of observations. With the increasing necessity to handle high-dimensional settings, parsimonious models have gained a lot of attention in recent years. Parsimonious models generally refer to some model instances where the number of parameters is reduced compared to the full parameterization. When dealing with high-dimensional data, the goal is to find a good compromise between model flexibility and parsimony. Complex models cannot be estimated accurately in most real life settings due to lack of data and simple models may not be able to capture the full data structure. In terms of number of parameters, the largest costs usually come from high-dimensional covariance matrices. Diagonal covariance matrices are parsimonious and often tractable in very high-dimension but they cannot capture complex dependence structures. Significant gains are then expected from adopting parsimonious covariance representations. In model-based clustering (Banfield and Raftery, 1993; Fraley and Raftery, 2002), covariance matrices are decomposed into eigenvalues. The number of parameters can be reduced by assuming constraints on the eigen-decomposition, e.g. (Bouveyron et al., 2007). Factor models (McLachlan and Peel, 2000) that induce a decomposition into a diagonal and a low rank matrix also result in a parsimonious parameterization. The concept of sparsity (Städler et al., 2010; Tibshirani, 1996; Yi and Caramanis, 2015) is

another very popular example yet different as parsimonious covariance matrices may not be sparse.

In a mixture of regressions context, the work of (Subedi et al., 2013) uses a factor analyzer approach (CWFA) to enhance CW modeling when the number of covariates is large. The high dimensionality issue is overcome by imposing constraints on the covariance matrix of the high-dimensional variables. GLLiM, in its more general version, referred to as hybrid-GLLiM in Deleforge et al. (2015) also adopts such a factor-model based parameterization to accommodate the high-dimensional covariates (see eq. (20) in Deleforge et al. (2015)) but with a different interpretation. In particular, the high-dimensional variables were postulated as a sum of two components: the one that is linearly related with the low-dimensional responses, and the other which can be projected onto a factor model and then be presented as augmented latent variables. This data-augmentation approach is interesting for solving regression problems in many application scenarios, whenever certain variables are only partially observed or corrupted with irrelevant information. This augmentation step, with added latent variables, acts as a factor analyzer modeling for the noise covariance matrix in the regression model. GLLiM is based on a joint modeling of both the responses and covariates, observed or latent. This joint modeling allows for the use of an inverse regression strategy to handle the high dimensionality of the data. Mixtures are used to approximate non-linear associations. GLLiM groups data with similar regression association together. Within the same cluster, the association can be considered as locally linear, which can then be resolved under the classical linear regression setting. However, when the covariate dimension is much higher than the response dimension, GLLiM may result in erroneous clusters at the low dimension, leading to potentially inaccurate predictions. Specifically, when the clustering is conducted at a high joint dimension, the distance at low dimension between two members of the same cluster (component) could remain large. As a result, a mixture component might contain several sub-clusters and/or outliers, violating the model Gaussian assumption (see examples in Figure 1 and Table 1). This results in a model misspecification effect that can seriously impact prediction performance. A natural way to lessen this effect is to increase the number of components in the mixture making each linear mapping even more local. But this also increases the number of parameters to estimate and therefore requires to be done in a parsimonious manner to avoid over-parameterization.

In this work, we propose a parsimonious approach which we refer to as Structured Mixture of Gaussian Locally Linear Mapping (SMoGLLiM) to solve the aforementioned problems. It follows a two-layer hierarchical clustering structure where local components are grouped into global components sharing the same high-dimensional noise covariance structure, which effectively reduces the number of parameters of the model. SMoGLLiM also includes a pruning algorithm for eliminating outliers as well as determining an appropriate number of clusters. Moreover, the number of clusters and training outliers determined by SMoGLLiM can be further used by GLLiM for improving prediction performance. As an extension, a subsetting and parallelization techniques are discussed for the efficiency concern.

With the goal of investigating the flexibility in accommodating data structure and the ability to protect from influences of outliers, we evaluate our method on three datasets

with different characteristics. The face dataset contains face images, the associated angles of faces and the source of the light. There is no obvious cluster structure at first glance nor the existence of real outliers. We use this dataset to evaluate the ability of SMoGLLiM on modeling regression relationship through local linear approximation. The orange juice dataset contains continuous spectrum predictors and some abnormal observations. Using this dataset, we aim to show that SMoGLLiM is robust and can effectively identify outlying observations. We use these two moderate size datasets to demonstrate how the method works at data with different features and the insensitivity of tuning parameter selection on a wide range of selection domain. Finally, at our last data analysis, we study a problem where researchers are interested in predicting the microvascular properties using the so-called magnetic resonance vascular fingerprinting (MRvF). Hereafter we refer to this dataset as the fingerprint data. In Lemasson et al. (2016), a synthetic fingerprint dictionary is obtained with computer simulations. The assessment on a real world fingerprint sample is done by finding the closest match in the synthetic fingerprint dictionary, which provides a connection of microvascular parameters between the synthetic data and the real world data. Here we focus on predicting measurements of two parameters: blood volume fraction (BVf) and apparent diffusion coefficient (ADC). Although the results of using dictionary matching on the BVf is satisfactory, there is still room for improvement on predicting ADC. In addition, finding appropriate dictionary matches for real-world observations can be computationally costly. SMoGLLiM provides a model-based alternative that finds the intrinsic relation between microvascular parameters and the fingerprints. In contrast, SMoGLLiM takes only 25% of the time required by the dictionary matching approach, which is implemented using MATLAB with parallel computing toolbox, in conducting prediction. On top of that, we explore the use of the hierarchical structure of SMoGLLiM that can adopts parallelization techniques for handling large dataset. This practice further reduces the original processing time by about 97%. Results show that SMoGLLiM can provide comparable prediction performance on BVf and smaller prediction error on ADC.

This paper is organized as follows. In Section 2 we explain and illustrate the issue encountered with unstructured GLLiM in high-dimensional settings. In Section 3, we present the structured alternative that we propose. The experiment results on three real datasets are provided in Section 4. For the analysis of the fingerprint dataset, we also illustrate a parallelization/aggregation implementation of SMoGLLiM to further reduce the computation time. Finally, Section 5 concludes with discussion and potential future directions.

2. Unstructured Gaussian Locally Linear Mapping (GLLiM)

To predict low-dimensional data $X \in \mathbb{R}^L$ using high-dimensional data $Y \in \mathbb{R}^D$, GLLiM elegantly copes with several challenging issues simultaneously. The high-to-low mapping difficulty is circumvented by inverse regression. The role of the predictors and responses are exchanged. And then the desired high-to-low relationship could be easily converted from the low-to-high associations under a proper model construction. Non-linearity is approximated by mixtures of linear associations. The parameter estimation can be carried out by an Expectation-Maximization algorithm, which nicely incorporates estimation of

latent variables.

The original GLLiM model is hierarchically defined by the following equations:

$$p(Y = y|X = x, Z = k; \theta) = \mathcal{N}(y; A_k x + b_k, \Sigma_k). \quad (1)$$
$$p(X = x|Z = k; \theta) = \mathcal{N}(x; c_k, \Gamma_k), \quad (2)$$
$$p(Z = k; \theta) = \pi_k.$$

The model parameters are: $\theta = \{c_k, \Gamma_k, \pi_k, A_k, b_k, \Sigma_k\}_{k=1}^K$.

A distinct feature of GLLiM is that X needs not be a completely observable vector. In fact, it is set to have $X^\top = (T^\top,\ W^\top)$, where T contains the observable covariates and W, being latent, absorbs the remaining dependency and variation in Y. The inclusion of W strengthens the chance to reach validity of (1).

The issue with GLLiM is that the high dimensionality of the data may have an unexpected impact on the posterior probability of the cluster assignment. When $D >> L$, this comes from the following observation: in the ith iteration of E-step with $\hat{\theta} = \theta^{(i)}$, the posterior probabilities r_{nk} (Equation (27) in Deleforge et al. (2015)) could be computed as:

$$\tilde{r}_{nk} = p(Z_n = k|x_n, y_n; \theta^{(i)}) = \frac{\pi_k^{(i)} p(y_n, x_n|Z_n = k; \theta^{(i)})}{\sum_{j=1}^K \pi_j^{(i)} p(y_n, x_n|Z_n = j; \theta^{(i)})} \quad (3)$$

for all n and all k, where $p(y_n, x_n|Z_n = k; \theta^{(i)})$ can be computed as $p(y_n|x_n, Z_n = k; \theta^{(i)}) p(x_n|Z_n = k; \theta^{(i)})$. The first term is a density with much higher dimension (D) so that its value could dominate the product. In addition, x_n can be decomposed into two parts: the observed variable t_n and the latent variable w_n. The component w_n reflects the remaining variation in y_n that cannot be explained by y_n's association with t_n. When w_n accounts for explaining most of the variation in y_n, the clustering outcome would highly depend on w_n and weakens the ability in detecting sub-clusters in T.

Therefore, although GLLiM assumes that within each cluster $p(X = x|Z = k; \theta)$ is Gaussian and centered on c_k, in practice, the model groups data according to the high-dimension term and could fail in imposing the Gaussian shape on the t_n's. In other words, the model rather chooses the clusters to satisfy the assumption in Equation (1). And this induces a clustering of the (x_n, y_n)'s into groups within which the same affine transformation holds. Thus, a cluster could contain several sub-clusters and/or outliers since the Gaussian assumption on T, as part of the X, in Equation (2) is sometimes neglected. This may cause a serious impact on the estimation of c_k and Γ_k and consequently on the prediction step.

We illustrate this issue by presenting an example using a face dataset (Tenenbaum et al., 2000). This dataset contains 698 images (of size 64×64 and being further condensed to 32×32). The pose of each image is defined by three variables of T: *Light*, *Pan* and *Tilt*, as shown in Figure 1 (a). We adopt GLLiM to predict these T's (low-dimensional) using the image (high-dimensional). The superiority of GLLiM in prediction, comparing to multiple existing approaches, for this data set was numerically illustrated in Deleforge et al. (2015).

Figure 1(b) shows the scatter plot of T within Clusters 7 and 13, grouped by GLLiM. By visual inspection, both clusters seem to consists of two or more sub-clusters. In GLLiM, samples within the same cluster are assumed to follow Gaussian distributions. This sub-cluster structure, however, violates the assumption and potentially increases the prediction errors. We demonstrate the difference of prediction performance before and after accounting for the sub-cluster structure in Table 1. We use prediction Sum of Squared Error (SSE) for testing data pre- and post cluster-division. We observe that the prediction errors are mostly reduced if we account for the sub-cluster structure.

Dividing samples at low dimension is an effective and straightforward solution for this issue. However, we could obtain small sub-clusters after division and the prediction variance of these small sub-clusters could be large. In Table 1, Images 114 and 223 were assigned to small and/or tight local clusters and the prediction of T for these two images become worse after cluster-division. Conceptually, small clusters could damage the prediction performances for several reasons: the small number of observations in such a cluster lead to estimates with large variation; a small cluster with a small covariance matrix determinant (volume) could lead to instability of the whole likelihood-based algorithm, and a small/tight cluster could consider a close-by testing sample unfit and force it to be predicted by another less suitable cluster with a larger within-cluster covariance. The last consideration is not relevant to model building but plays an important role in prediction precision.

This observation motivates us to look into enhancing prediction stability and accuracy by eliminating small clusters in the training samples. Another factor that would influence stability is the presence of outliers. We further explore both issues in Section 4.

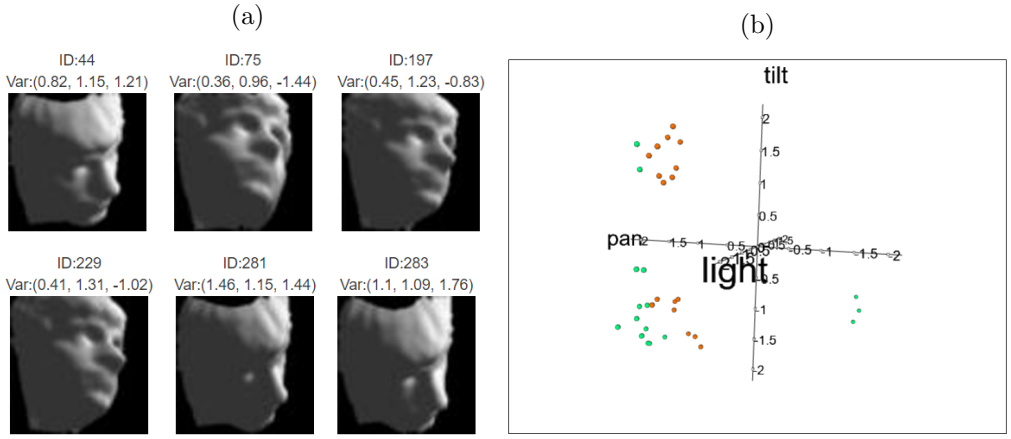


Fig. 1: The clustering results of the face dataset obtained from GLLiM: (a) six face images from Cluster 7; (b) scatter plot of T for points within Cluster 7 and 13 clustered by GLLiM. Data with orange color are from Cluster 7, and data with green color are from Cluster 13. Three variables are (*Light*, *Pan*, *Tilt*).

Table 1: The comparison of original and post cluster-division SSE. The improved ratio is calculated as the ratio of difference of SSE from pre- to post cluster-division over the original SSE. The value is positive if the procedure reduces the SSE, and negative, otherwise.

Image ID	GLLiM cluster	Original SSE	Post-Division SSE	Improved ratio
56	7	0.306	0.043	86.03%
223	7	0.016	0.180	-1039.83%
293	7	0.060	0.023	61.27%
302	7	0.087	0.003	96.99%
114	13	0.114	0.118	-2.93%
204	13	0.307	0.073	76.19%
294	13	3.119	0.120	96.15%

3. Structured Mixture of Gaussian Locally Linear Mapping (SMoGLLiM)

In our proposed work, we intend to strike a balance between model flexibility and variation reduction in the estimated predictive model, with the goal of predicting the low-dimensional observable variables, T, using the high-dimensional Y. These predictive models need not be the true model but they could be effective in prediction. To present the fundamental concepts with clarity, we will first describe the model structure when $X = T$, with minimum required notations. The scenario of X containing W is easily extended in Section 3.2.

3.1. Model description

The joint probability of high-dimensional data Y and low-dimensional data X can be written as:

$$p(Y = y, X = x; \theta) = \sum_{k=1}^{K}\sum_{l=1}^{M} p(Y = y|X = x, Z = k, U = l; \theta)p(X = x|Z = k, U = l; \theta)p(Z = k, U = l; \theta),$$

where θ denotes the vector of parameters; Z and U are, respectively, latent indicators of global and local cluster assignment. The linear relationship between X and Y is given by the model below:

$$Y = \sum_{k=1}^{K}\sum_{l=1}^{M} \mathbb{I}(Z = k, U = l)(A_{kl}X + b_{kl} + E_k),$$

where $\mathbb{I}$ is the indicator function, $A_{kl} \in \mathbb{R}^{D\times L}$ and $b_{kl} \in \mathbb{R}^{D}$ maps X onto Y, and $E_k \in \mathbb{R}^{D\times D}$ is the error term that absorbs the remaining uncertainty. Here, we let the local cluster size $M(k) \equiv M$ for notation simplicity only. We assume, within a global cluster, all local clusters share the same error structure which follows zero-mean Gaussian with covariance matrix Σ_k. We obtain:

$$p(Y = y|X = x, Z = k, U = l; \theta) = \mathcal{N}(y; A_{kl}x + b_{kl}, \Sigma_k).$$

The model is completed by assuming X follows a mixture of Gaussian and defining a prior for clustering assignment:

$$p(X = x|Z = k, U = l, \theta) = \mathcal{N}(x; c_{kl}, \Gamma_{kl}),$$
$$p(Z = k, U = l, \theta) = \rho_{kl},$$

where $c_{kl} \in \mathbb{R}^L$, $\Gamma_{kl} \in \mathbb{R}^{L\times L}$ and $\sum_{k=1}^{K}\sum_{l=1}^{M}\rho_{kl} = 1$. It follows that the parameters in the inverse regression model are:

$$\theta = \{c_{kl}, \Gamma_{kl}, \rho_{kl}, A_{kl}, b_{kl}, \Sigma_k\}_{k=1,l=1}^{K,M}.$$

The inverse conditional density can be written as:

$$p(Y = y|X = x; \theta) = \sum_{k=1}^{K}\sum_{l=1}^{M}\frac{\rho_{kl}\mathcal{N}(x; c_{kl}, \Gamma_{kl})}{\sum_{i=1}^{K}\sum_{j=1}^{M}\rho_{ij}\mathcal{N}(x; c_{ij}, \Gamma_{ij})}\mathcal{N}(y; A_{kl}x + b_{kl}, \Sigma_k),$$

and the conditional density in the forward regression model is expressed as:

$$p(X = x|Y = y; \theta^*) = \sum_{k=1}^{K}\sum_{l=1}^{M}\frac{\rho^*_{kl}\mathcal{N}(y; c^*_{kl}, \Gamma^*_{kl})}{\sum_{i=1}^{K}\sum_{j=1}^{M}\rho^*_{ij}\mathcal{N}(y; c^*_{ij}, \Gamma^*_{ij})}\mathcal{N}(x; A^*_{kl}y + b^*_{kl}, \Sigma^*_{kl}),$$

where θ^* denotes the parameter vector in the forward regression model, as:

$$\theta^* = \{c^*_{kl}, \Gamma^*_{kl}, \rho^*_{kl}, A^*_{kl}, b^*_{kl}, \Sigma^*_{kl}\}_{k=1,l=1}^{K,M}.$$

Note that θ^* has closed-form expressions as functions of θ, which makes it computationally efficient. The relation is obtained analytically with:

$$\begin{aligned}
c^*_{kl} &= A_{kl}c_{kl} + b_{kl},\\
\Gamma^*_{kl} &= \Sigma_k + A_{kl}\Gamma_{kl}A_{kl}^\top,\\
\rho^*_{kl} &= \rho_{kl},\\
A^*_{kl} &= \Sigma^*_{kl}A_{kl}^\top\Sigma_k^{-1},\\
b^*_{kl} &= \Sigma^*_{kl}(\Gamma_{kl}^{-1}c_{kl} - A_{kl}^\top\Sigma_k^{-1}b_{kl}),\\
\text{and } \Sigma^*_{kl} &= (\Gamma_{kl}^{-1} + A_{kl}^\top\Sigma_k^{-1}A_{kl})^{-1}.
\end{aligned}$$

The prediction can be done by taking expectation over the forward conditional density:

$$\mathbb{E}[X|Y = y] = \sum_{k=1}^{K}\sum_{l=1}^{M}\frac{\rho^*_{kl}\mathcal{N}(y; c^*_{kl}, \Gamma^*_{kl})}{\sum_{i=1}^{K}\sum_{j=1}^{M}\rho^*_{ij}\mathcal{N}(y; c^*_{ij}, \Gamma^*_{ij})}(A^*_{kl}y + b^*_{kl}) \tag{4}$$

3.2. SMoGLLiM model with partially-latent responses

Recall that the low-dimensional data X contains a latent component W. Namely,

$$X = \begin{bmatrix} T \\ W \end{bmatrix},$$

where $T \in \mathbb{R}^{L_t}$ is observed and $W \in \mathbb{R}^{L_w}$ is latent and $L = L_t + L_w$. It is assumed that T and W are independent given Z, and so are W and U. According to the decomposition of X, the corresponding mean (c_{kl}), variance (Γ_{kl}) and regression parameters (A_{kl}) of X, at the local-cluster level, are given as:

$$c_{kl} = \begin{bmatrix} c_{kl}^t \\ c_k^w \end{bmatrix}, \ \Gamma_{kl} = \begin{bmatrix} \Gamma_{kl}^t & 0 \\ 0 & \Gamma_k^w \end{bmatrix}, \ \text{ and } A_{kl} = [A_{kl}^t \ A_k^w]. \tag{5}$$

That is, when $Z = k$, $U = l$, at the local-cluster level, $T \sim N(c_{kl}^t, \Gamma_{kl}^t)$; when $Z = k$, at the global-cluster level, $W \sim N(c_k^w, \Gamma_k^w)$. It follows that locally, the association function between the high-dimensional Y and low-dimensional X can be written as:

$$Y = \sum_{k=1}^{K} \mathbb{I}(Z = k) \left\{ \sum_{l=1}^{M} \mathbb{I}(U = l)(A_{kl}^t T + b_{kl}) + A_k^w W + E_k \right\}, \tag{6}$$

Finally, the parameter vector θ in the inverse regression model is rewritten as: $\theta = \{\rho_{kl}, c_{kl}^t, \Gamma_{kl}^t, A_{kl}^t, b_{kl}, c_k^w, \Gamma_k^w, A_k^w, \Sigma_k\}_{k=1,l=1}^{K,M}$.

It follows that (6) rewrites equivalently as

$$Y = \sum_{k=1}^{K} \mathbb{I}(Z = k) \left\{ \sum_{l=1}^{M} \mathbb{I}(U = l)(A_{kl}^t T + b_{kl}) + A_k^w c_k^w + E_k' \right\}, \tag{7}$$

where the error vector E_k' is modeled by a zero-centered Gaussian variable with a $D \times D$ covariance matrix given by

$$\Sigma_k' = \Sigma_k + A_k^w \Gamma_k^w A_k^{w\top}. \tag{8}$$

Considering realizations of variables T and Y, one may view this as a supervised model in which the noise covariance has a specific structure, namely (8), where $A_k^w \Gamma_k^w A_k^{w\top}$ is at most a rank-L_w matrix. When Σ_k is diagonal, this structure is that of factor analysis with at most L_w factors, and represents a flexible compromise between a full covariance with $O(D^2)$ parameters on one side, and a diagonal covariance with $O(D)$ parameters on the other.

The model parameters, θ, can be estimated using the Expectation-Maximization (EM) algorithm (see Appendix A.1). We also discover that the model stability plays an important role in the prediction performance. SMoGLLiM controls the cluster size and removes abnormal observations to guard against disturbances caused by outliers or unstable clusters using a refined alternative whose details can be found in Appendix A.2.

4. Numerical Investigation

We analyze three datasets and use the outcomes to illustrate the usage of the proposed method. Key features of each data, thus the type of data they represent, are reported in the corresponding subsections. Our primary focus is on the analysis of fingerprint data but the other two smaller datasets illustrate the wide usage of the method and the use of tuning procedure. Throughout, unless otherwise noted, we use squared error to

evaluate prediction merit for each data point. The squared error (E^2) for each testing sample i is calculated using:

$$E_i^2 = ||t_i^{pred} - t_i||_2^2,$$

where t_i^{pred} is the prediction of testing data i and t_i is the true value. We also calculate the prediction mean squared error (MSE) among all testing samples with $MSE = \sum_{i=1}^{N_{test}} E_i^2/N_{test}$, where N_{test} is the total number of testing samples.

We calculate and compare the MSE or the quantiles of E^2 over several methods:

(a) SMoGLLiM: This is the proposed method. The user-defined parameters K and L_w are set to the value using the method described in Appendix A.3. The number of local clusters M is set to 5 to reflect the possible sub-cluster structure. In each global cluster, the number of local clusters varies and depends on the structure of the dataset. Some of the local clusters would be dissolved so the number of local clusters could be less than M. The initial cluster assignment is done by dividing the GLLiM clustering outcomes at the low dimension using R package *mclust*. As stated before, the refined version of the EM algorithm is used throughout the version. For every experiment, we follow the suggestion described in the previous section. We set $minSize = 5$ for all three analyses and post-analysis checks at the neighborhood of 5 suggest this is an appropriate choice. The prediction MSE using different values of $dropThreshold$ would be calculated and compared.

(b) GLLiM: The original GLLiM algorithm. GLLiM is compared to other methods under the same settings of K and L_w. The initial cluster assignment is done by applying Gaussian Mixture Model to a dataset that combines the low- and high-dimensional T and Y together.

(c) GLLiM Structure: This method adopts the number of mixtures learned structurally by SMoGLLiM. In addition, outliers identified by SMoGLLiM are removed from the training dataset. We adopts the same tuning parameters as GLLiM and the initial conditions are obtained from the outcomes of SMoGLLiM. The key difference between GLLiM Structure and SMoGLLiM is that GLLiM Structure uses local estimated covariance, which may be more appropriate for a large and more data set. Its effectiveness also suggests an additional usage of SMoGLLiM, in terms of structure learning and identification of outliers.

4.1. The face dataset

The face dataset was analyzed in the original GLLiM paper (Deleforge et al., 2015). For this dataset, we are interested in predicting the pose parameters ($L_t = 3$) using the image information. The size of each image is condensed to 32×32, and thus $D = 1024$. In addition, T is standardized so that they are equally weighted in each low-dimensional direction. The histograms of the three T variables bear no clustering structure. Consequently, the mixture modeling serves the purpose of local linear approximation and the inverse regression enables dimension reduction.

For each run, we select 100 testing samples and 598 training samples. We repeated this procedure 20 times to establish 2000 testing samples. According to the approach

Table 2: The prediction MSE and the average cluster number of the face dataset when $dropThreshold = 0.5$.

	K=10		K=15		K=20		K=25	
	MSE	#Cluster	MSE	#Cluster	MSE	#Cluster	MSE	#Cluster
GLLiM	0.0711	10.00	0.0441	15.00	0.0369	20.00	0.0321	25.00
SMoGLLiM	0.0314	43.90	0.0318	51.35	0.0294	53.75	0.0295	53.45
GLLiM Structure	0.0307	43.90	0.0301	51.35	0.0291	53.75	0.0288	53.45

described in Appendix A.3, we run cross-validation on K from 10 to 25, L_w from 1 to 10. The range of L_w is extended to 15 to demonstrate the trend as L_w increases. The cross-validation results in Figure 2(a) suggest that $K = 20$, $L_w = 9$. It is also observed that the change of prediction error is relatively small when L_w exceeds a certain value. Therefore, we fix the number of latent factors and compare the prediction performance under $K = 10$, $K = 15$ and $K = 25$.

(a) (b)

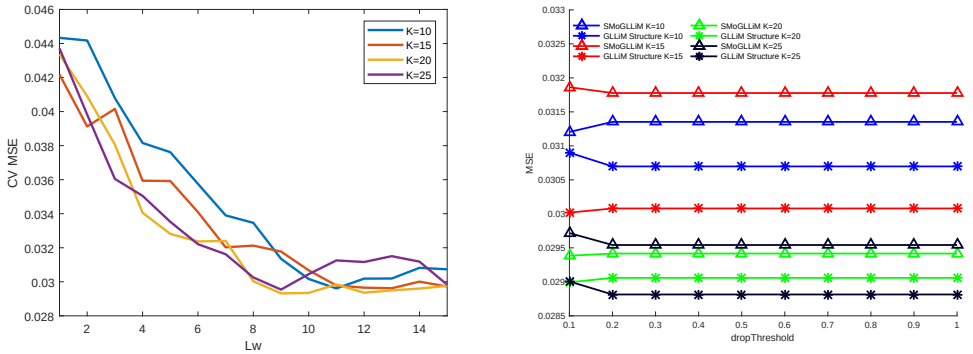


Fig. 2: Results for different user-defined parameters of the face dataset. (a) The SMoGLLiM cross-validation results for different K and L_w. (b) The prediction MSE of different K and different methods against different $dropThreshold$.

Figure 2(b) shows prediction outcomes under different values of $dropThreshold$. We observe that for different methods and different K, the prediction MSEs are not sensitive to the values of $dropThreshold$. Thus, we compare the prediction MSE of SMoGLLiM, GLLiM Structure when $dropThreshold = 0.5$ to GLLiM in Table 2. The prediction MSE for GLLiM decreases as K increases, which indicates that more clusters could be helpful to capture the non-linear relationship between Y and T. For SMoGLLiM, we observe that the prediction MSE is not sensitive to the choice of K. In addition, the numbers of clusters are similar under different choices of K. This indicates that SMoGLLiM could adjust itself to reach the number of clusters suitable to its setting. As for GLLiM Structure, the prediction MSE's are slightly smaller than those of SMoGLLiM. This is because GLLiM Structure estimates all parameters using local clusters, local covariances, and the prediction would be less biased when the local structures sufficiently differ. In the face dataset, there is no obvious cluster structure and, as a result, clustering

only serves the purpose of improving local approximation. Thus, the prediction MSE for GLLiM Structure would be smaller. However, the differences of prediction MSE's between SMoGLLiM and GLLiM Structure are small, which implies that the settings learned from SMoGLLiM are appropriate, even though SMoGLLiM imposes a global-cluster structure when there is none. Overall, the prediction performance for SMoGLLiM is similar when $K = 20$ and $K = 25$. As for GLLiM Structure, the MSE is smaller when $K = 25$ but the difference is negligible.

4.2. The orange juice dataset

The orange juice dataset contains the spectra measured on different kinds of orange juice ($N = 218$). The goal is to use the spectra to predict the level of sucrose ($L_t = 1$). We follow the step described in Perthame et al. (2018) and decompose the spectra on a spline basis with ($D = 134$) to make $D \approx N$. This dataset is known for the presence of outliers; the realization of Y and T is given in Figure 3.

We setup the following prediction evaluation procedure. In each run, we randomly select 20 testing samples from the main population (excluding outliers). The remaining 198 samples (including outliers, unless otherwise specified) are used for training. These outliers were identified through Leave One Out Cross Validation (LOOCV) using GLLiM, with $K = 10$ and $L_w = 2$. Although the set of outliers may differ for different selection of K, L_w, the severe outliers are always selected and they are included here. We identify 11 points, which are the observations with top 5% of the prediction E^2s (above 4.8) among all data points, as outliers. Removing outliers from testing data prevents the summarized outcomes being overwhelmed by prediction results of few points, which potentially make the differences among methods less obvious. All methods were evaluated using the same settings.

Figure 4(a) shows the cross-validation results, which suggests the use of $K = 5$, $L_w = 8$. For comparison purpose, we also provide MSE results for $K = 10$ and $K = 15$. The rest of the settings are the same as the experiment settings used for the face dataset.

To evaluate the influence of outliers on GLLiM, we conduct an analysis in which we use the same cluster number as in GLLiM Structure but without removing training outliers. This method is referred to as GLLiM Outlier. In addition, we consider SLLiM in Perthame et al. (2018) provided by the R package *xLLiM*. SLLiM is a counterpart of GLLiM that accommodates abnormal samples using Student's t-distribution. Precisely, instead of normal, the high-dimensional Y is now modeled by a mixture of K generalized multivariate Student's t-distributions, using the structure given in Section 5.5 (p.94) of Kotz and Nadarajah (2004). We also compare SLLiM performances by using the same cluster number learned structurally by SMoGLLiM. We refer to the resulting procedure as "SLLiM Structure". We use the default settings in *xLLiM* for the remaining SLLiM configurations.

Figure 4(b) shows the prediction MSE for different *dropThreshold*. The prediction MSEs vary, mainly reflect the high variation in this data set, partially due to outliers. For a small *dropThreshold*, the number of identified training outliers is more than expected. This reduces the training data size and makes the prediction unreliable . As *dropThreshold* reaches a reasonable value, the prediction performance becomes better. However, more and more abnormal training samples would be included in the

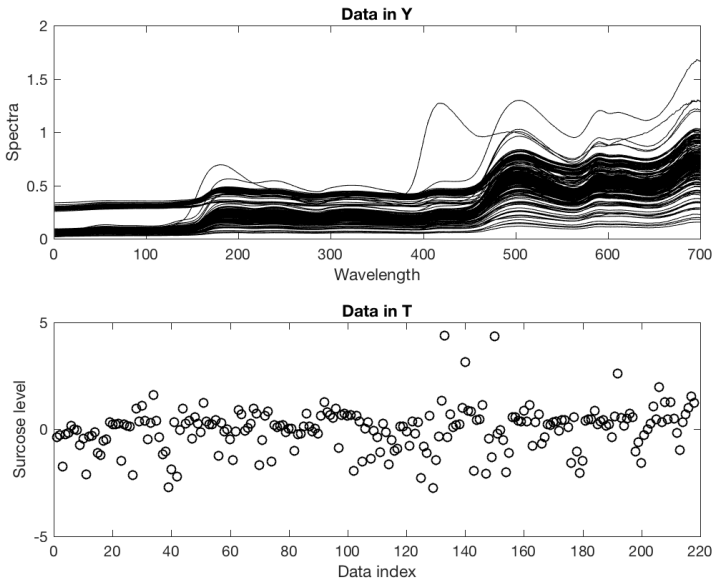


Fig. 3: The orange Juice dataset. The upper panel shows the high-dimensional data (Y) and the lower one shows the low-dimensional data (T).

(a) (b)

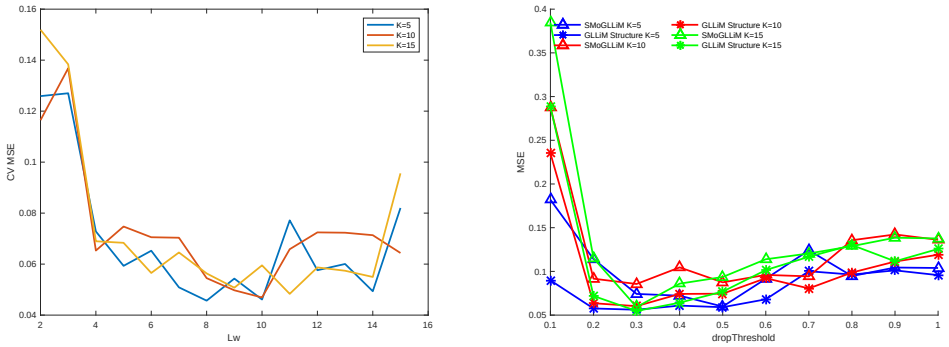


Fig. 4: Results for the user-defined parameters of the orange juice dataset. (a) The SMoGLLiM cross-validation results for different K and L_w. (b) The prediction MSE of different K and different methods against different *dropThreshold*.

training dataset as *dropThreshold* keeps increasing. These outlying data enlarge the model variance and downgrade the prediction performance. Table 3 shows the results for *dropThreshold* $= 0.5$. We observe that for $K = 5$, the cluster number is not sufficiently large for GLLiM to capture the non-linear trend in the data, which results in a relatively large prediction MSE. SMoGLLiM, on the other hand, adjusts the cluster number automatically and the prediction errors are smaller. In addition, SMoGLLiM

Table 3: The prediction MSE and the average cluster number of the orange juice dataset when $dropThreshold = 0.5$.

	K=5		K=10		K=15	
	MSE	#Cluster	MSE	#Cluster	MSE	#Cluster
GLLiM	0.1259	5.00	0.1210	10.00	0.0918	15.00
SMoGLLiM	0.0587	9.95	0.0681	11.85	0.0692	12.80
GLLiM Structure	0.0621	9.95	0.0742	11.85	0.0746	12.80
GLLiM Outlier	0.0976	9.95	0.1171	11.85	0.1044	12.80
SLLiM	0.1039	5.00	0.0788	10.00	0.0706	15.00
SLLiM Structure	0.0907	9.95	0.0747	11.85	0.0721	12.80

removes training outliers that would deteriorate the model performance. This explains why even though the cluster number is large enough ($K = 10$, $K = 15$), GLLiM still suffers for large prediction errors. We can also observe the benefit of removing outliers by comparing GLLiM Structure and GLLiM Outlier. The prediction errors for GLLiM Structure are smaller than those produced by GLLiM Outlier and the only difference between GLLiM Structure and GLLiM Outlier is whether training outliers, identified by SMoGLLiM, are removed. There are 11 outliers in the training dataset. SMoGLLiM could effectively identify and remove all of them. In addition to these outliers, some potential outlying samples that could result in unstable model are trimmed as well. Overall, about 6% to 10% of the training samples would be removed by SMoGLLiM. SLLiM and SLLiM Structure use t-distributions to accommodate the existence of outliers. They are expected to perform better than their Gaussian counterparts (GLLiM and GLLiM Outlier). When K is small, it is observed that accommodating outliers with t-distributions is not as effective as removing them by comparing SLLiM Structure and GLLiM Structure. However, as $K = 15$, the cluster size is small and removing outliers would further reduce the cluster size. Under $K = 15$, SMoGLLiM outperforms other methods since it controls the cluster size for robustness concern. In addition, SMoGLLiM estimates the covariance matrices under global cluster level and this estimation is more reliable compared to GLLiM Structure, which estimates covariance matrices locally. SLLiM does not remove any samples and thus the performance would be better than GLLiM Structure under this situation. Although removing outliers is more effective, accommodating outliers may still be an alternative to combat outliers when the cluster size is the concern.

4.3. A magnetic resonance vascular fingerprint dataset

It is of great interest to scientific community to be able to efficiently assess microvascular properties, such as, blood volume fraction, vessel diameter, and blood oxygenation, in brain so that the ability in diagnosis and management of brain diseases can be improved. Recently, a new approach called magnetic resonance vascular fingerprinting (MRvF) was proposed as an alternative to overcome the limitations of analytical methods in measuring microvascular properties. The approach was built on a system in which the signal acquired in each voxel, also called "fingerprint", was compared to a dictionary obtained from numerical simulation. Finding the closest match to a fingerprint record in the dictionary allows a direct link between the parameters of the simulations and the

microvascular variables (also referred to as parameters in these studies) at the image voxel (Lemasson et al., 2016; Ma et al., 2013).

A synthetic MRv fingerprint (hereafter referred to as fingerprint) dataset composed of $1,383,648$ observations was created to serve as a "search/match" library. Each observation in the library consists of a fingerprint measurement and associated parameters: mean vessel radius (Radius), Blood Volume Fraction (BVf) and a measurement of blood oxygenation (DeltaChi). The goal is to predict these parameters ($L_t = 3$) using the fingerprint measurement ($D = 32$). In addition to these three parameters, other parameters (variables) that have influence over the fingerprint measurements include Apparent Diffusion Coefficient (ADC), vessel direction (Dir) and vessel geometry (Geo).

Model building time is a critical issue for large and dataset. As the number of samples increases, the time of computing posterior probability in the E-step increases. In addition, it takes longer for the EM algorithm to converge and it would be more difficult to find a proper initial setting. To speed up the computation, we could take advantage of the hierarchical structure of SMoGLLiM. The model building step can be accelerated by subsetting the dataset into smaller groups and applying SMoGLLiM on the resulting groups in parallel. Finally, the prediction can be conducted using the estimated model aggregated from different groups.

We divide the synthetic library into 20 groups (see Appendix B for more details) according to different combinations of Dir (6 levels), Geo (3 categories) and Radius (3 levels). In addition, we observe the peak value in fingerprint could play an important role when forming cluster. Thus, groups are further divided into "Low-peak" and "High-peak" subgroups, according to the average of the three highest values in the fingerprint signal. The threshold to separate these two subgroups is set to be 5.

Training is performed within each group separately. In each group, the cluster membership is estimated and model parameters are updated based on the cluster membership iteratively until convergence. Once the models for each group are obtained, the prediction can be conducted by combining models from different groups. This can be easily done by creating a new latent global-cluster indicator, $Z^* = (G, Z)$, where G indicates the group assignment and Z is the cluster assignment within group G. By replacing Z with Z^*, the backward parameter vector, θ, aggregates all backward parameters from different groups. The forward regression model parameter vector, θ^*, can be updated accordingly using equations described in Section 3.1. Prediction is done by taking expectation over the forward conditional density with the newly updated θ^*. One key potential drawback of adopting group-division is using unsuitable group-assignment strategy. When this happens, the posterior probability of a data point belonging to a group A given the estimated model parameters could remain high even though this data was originally assigned to group B. Thus, to accommodate the group-division, parallel computing strategy, we create a latent global cluster indicator Z^*. Replacing Z with Z^* in (11) and summing all r_{nk} in (11) with Z^* belonging to a group g produce the posterior probability of data point n belonging to that group g. The group with the highest group posterior probability represents the most suitable group assignment and it bears the largest weights when conducting prediction. If a data point is indeed assigned to the group it originally belongs to, we consider the group assignment being accurate. In our analysis, the accuracies of group assignment for SMoGLLiM and GLLiM Structure

are 92.11%, 92.34%, respectively. A similar rate of 93.62% is obtained for the dictionary matching method, in which the accuracy of group assignment is obtained when a point and its closest match identified for prediction belong to the same group. Additionally, the highest group posterior probability is greater than 0.99 for over 97% of the data. These numbers imply that, for the analysis of this fingerprint dataset, our strategy on how the groups should be formed in conducting parallel computing is adequate.

In terms of saving in computation time via group-division and parallel computing, major factors influencing the model building time are the number of training samples and the total number of clusters. When dividing into groups, the time allotted to realize these 2 tasks could be effectively reduced and model could be built in parallel.

When the training dataset is divided into groups and parallelization is adopted, for SMoGLLiM and GLLiM Structure, it takes about 549.86 and 362.94 seconds, respectively, for each method to complete the EM computation. In comparison, it takes 19341.51 and 14107.63 seconds without using the parallel computing strategy. We evaluate and compare the performance of different methods through cross-validation. The numbers of testing and training data are picked so that every data in the smallest group could be covered in 20-fold cross-validation. In addition, we add a small amount of the real image data in the training dataset for calibration; fingerprint samples from the real world were noisier than their synthetic counterparts. This practice enables the training model to readily accommodate the real fingerprint samples in prediction. The ratio of the synthetic samples to the real image samples is 4 to 1. For each fold, there would be 68920 training samples and 2040 testing samples. The cluster number and latent factor number are selected using the method described in Appendix A.3 and were set to $K = 1240$ and $L_w = 9$, respectively.

In Lemasson et al. (2016), numerical performances of dictionary matching method were presented. For comparison purpose, we implement the dictionary matching method adopted in Lemasson et al. (2016). The coefficient of determination (r^2) is used to measure the similarity between a testing sample and the training samples (dictionary). The coefficient of determination, r^2, between testing sample y^{test} and the training sample y^{train} is calculated as:

$$r^2 = 1 - \frac{\sum_{d=1}^{D}(y_d^{test} - y_d^{train})^2}{\sum_{d=1}^{D}(y_d^{test} - \bar{y}^{test})^2}, \tag{9}$$

where $\bar{y}^{test} = \frac{1}{D}\sum_{d=1}^{D} y_d^{test}$. The matched fingerprint is the training fingerprint with the largest r^2 and we could predict the parameters of the testing data using the matched fingerprint.

Table 4 shows the 50%, 90% and 99% quantiles of the squared errors for different parameters using different methods through cross-validation. The outcomes reported under the 50th and 90th percentiles give the indication of "average" and "almost-all" prediction performances for each method. The 99th percentile values allow the comparisons of worse-case scenarios. We observe that the prediction is close to the true value for 90% of the predicted values. Using GLLiM, we would obtain slightly large values of E^2 for Radius. However, all four methods reach similar values of E^2 for Radius at the 99% quantile. For BVf, GLLiM performs worse than the other methods but its 99% squared error level is still acceptable. The prediction performances of BVf for all methods are

Table 4: The 50%, 90% and 99% quantiles of squared error using different methods.

	Dictionary matching			GLLiM			SMoGLLiM			GLLiM Structure		
	50%	90%	99%	50%	90%	99%	50%	90%	99%	50%	90%	99%
Radius	$< 10^{-4}$	0.2843	21.44	$< 10^{-4}$	0.3114	21.44	$< 10^{-4}$	0.2144	21.44	$< 10^{-4}$	0.2144	21.44
BVf	$< 10^{-4}$	$< 10^{-4}$	0.0023	$< 10^{-4}$	$< 10^{-4}$	0.0406	$< 10^{-4}$	$< 10^{-4}$	0.0091	$< 10^{-4}$	$< 10^{-4}$	0.0242
DeltaChi	$< 10^{-4}$	0.0143	0.3571	$< 10^{-4}$	0.0132	0.5972	$< 10^{-4}$	0.0009	0.2236	$< 10^{-4}$	0.0007	0.3361

Table 5: The mean predicted values within ROIs of different vascular parameters from different categories.

	9L			C6			F98			Stroke			Healthy		
	Radius	BVf	DeltaChi	Radius	BVf	DeltaChi	Radius	BV	DeltaChi	Radius	BVf	DeltaChi	Radius	BVf	DeltaChi
Dictionary matching	21.85	14.49	0.98	13.59	4.17	0.77	11.56	3.86	0.65	14.69	4.22	0.60	8.16	3.58	0.76
GLLiM	20.14	14.33	0.93	16.01	4.01	0.76	13.14	3.96	0.66	13.51	4.49	0.63	7.96	3.51	0.72
SMoGLLiM	22.12	14.71	1.03	13.67	4.25	0.79	11.13	4.01	0.62	14.31	4.13	0.62	8.54	3.63	0.74
GLLiM Structure	21.52	14.25	0.94	13.81	4.52	0.74	11.23	3.97	0.61	14.41	4.25	0.63	8.34	3.56	0.80

better than those of other parameters, with the relationship between BVf and Y being the strongest among all parameters. For DeltaChi, the E^2s for dictionary matching are larger than those of other methods at the 90% quantile level. At the 99% quantile level, its performances become similar to those of SMoGLLiM and GLLiM Structure. Note that the model is built using Radius, BVf, ADC and DeltaChi. The parameter ADC is included for evaluating the prediction performance on the real image data. However, for a model-based approach, it is known that ADC is not a very effective parameter and its inclusion in fact could downgrade the overall prediction performances. When the goals do not include predicting ADC, we would obtain a slightly better predictive model to predict other parameters, including BVf, as shown in Appendix B.

We now apply these methods to a fingerprint data set collected at an animal study. This dataset contains samples from 115 rats categorized into 5 different groups: healthy, 3 kinds of tumors (9L, C6 and F98) and stroke. For each rat, there are 5 brain slices of 128×128 voxels and each voxel contains 32 dimension fingerprint information. For each slice, the lesion (unhealthy) and the striatum (healthy) areas are labeled and they form the region of interest (ROI). Here, BVf is of particular interest since its values are different in the lesion and the striatum region, which can be used to distinguish healthy and unhealthy regions. Figure 5 shows the predicted BVf image using different methods. As indicated in Lemasson et al. (2016), the values of true BVf is not available at the voxel level, and instead, they are measured over the whole ROI's. Nevertheless, the comparison between the true values and those obtained by dictionary matching method, at the ROI level, indicates that the method has successfully provided close-to-truth match; see Lemasson et al. (2016). Table 5 shows the mean prediction results within the ROI's obtained by different methods. Three additional methods considered here, besides the method used in Lemasson et al. (2016), are GLLiM, SMoGLLiM and GLLiM Structure. All four methods provide similar prediction results.

There are 1,385,509 samples in the real image dataset. For the dictionary matching method using a parallel for-loop (*parfor*) and a pre-processing technique as in Lemasson et al. (2016), it took about 2.4 hours (precisely 8639.53 seconds) to match the whole real image samples to the training dataset ($N^{train} = 1,383,648$). A direct computation without *parfor* and pre-processing took 429507.79 seconds and reach the

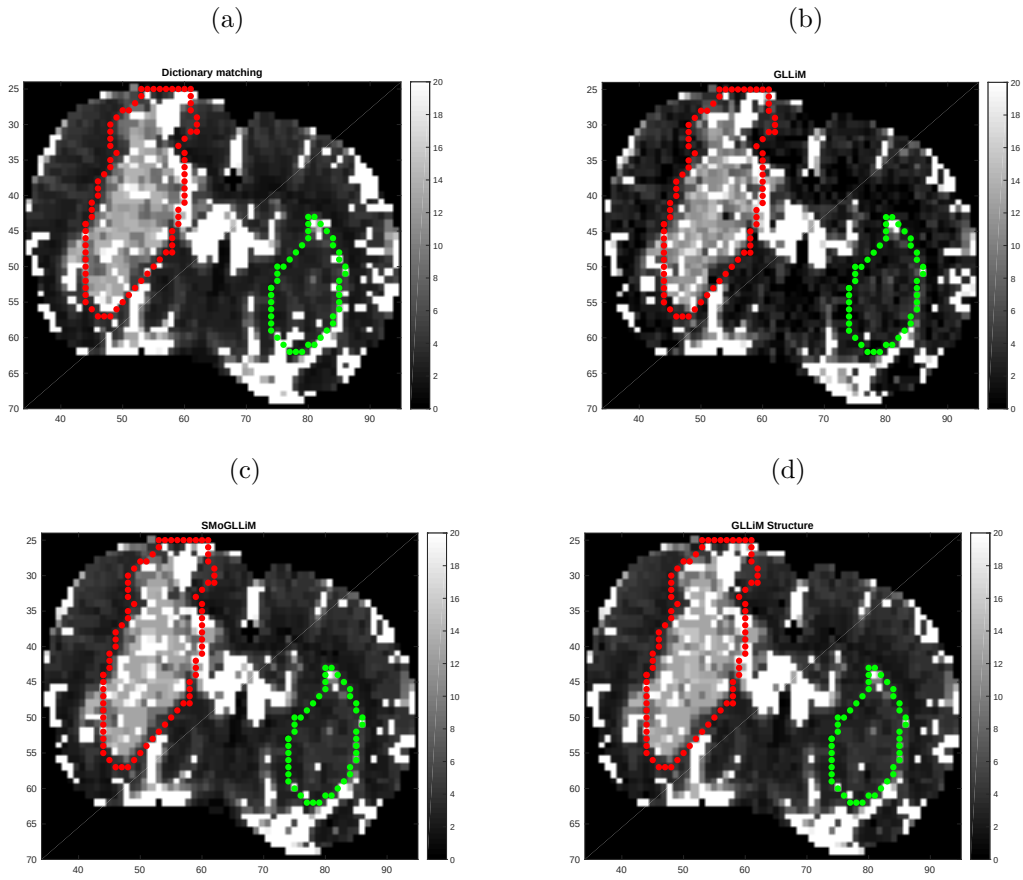


Fig. 5: The predicted BVf images of one animal from the 9L group using either (a) dictionary matching, (b) GLLiM, (c) SMoGLLiM or (d) GLLiM Structure. The lesion ROI and striatum ROI are labeled in red and green, respectively.

same outcomes. For the model-based method, utilizing the grouped structure and the parallel computing technique, it takes 1058.32/2133.51/1922.37 seconds for GLLiM/SMoGLLiM/GLLiM Structure to process the real image dataset. Thus, the prediction procedure of GLLiM/SMoGLLiM/GLLiM Structure is much more efficient than the dictionary matching method.

The parameter ADC was not thoroughly investigated in Lemasson et al. (2016). The main reason is that ADC values, obtained using the dictionary matching approach, were not comparable as the ones measured *in vivo* by MRI. Since, *in vivo* ADC values are available at the voxel-level, understanding how the synthetic data and real measurements differ for a given parameter is scientifically important to developing new instruments and future knowledge advancements. Here, we study ADC and use it to evaluate the prediction performances of different methods. Figure 6 shows the true ADC image and the images of the differences between the true and predicted ADC values. The differences are shown in the ratio against the signal levels for each ROIs. Most of the predictions

Table 6: The 50%, 90% 99% quantiles of ADC squared errors for different methods on different image categories.

	Dictionary matching			GLLiM			SMoGLLiM			GLLiM Structure		
	50%	90%	99%	50%	90%	99%	50%	90%	99%	50%	90%	99%
9L	1.1180	3.9803	10.6829	0.2392	0.5684	14.5668	0.1132	0.7613	11.8721	0.1018	0.7154	10.9574
C6	1.1208	4.4719	14.4888	0.3043	2.6091	26.7575	0.3252	1.9840	22.5427	0.3138	1.7764	20.0213
F98	1.0994	4.2373	14.4888	0.3802	3.4129	55.4479	0.2951	2.3672	35.3199	0.2801	2.4129	50.8133
Stroke	1.1663	5.8045	14.8888	0.4779	4.5668	66.1164	0.3218	3.0975	55.7821	0.3192	3.1424	53.9315
Health	1.0931	3.8086	7.7912	0.2131	1.2510	14.5668	0.1527	1.1087	11.9597	0.1054	1.1145	13.2165

made by dictionary matching are deviated from the true values. On the other hand, SMoGLLiM and GLLiM Structure provide better ADC images. There are some voxels with extreme differences (dark red or dark blue) that all methods cannot predict well. The prediction on these voxels are limited by the information provided by the synthetic fingerprint data. If no suitable training information could be used for conducting prediction, the nearest training samples would be used and results in unreliable outcomes. This would be discussed later in details. Table 6 shows the 50%, 90% and 99% quantiles of the ADC squared errors. We still obtain predictions with large errors using GLLiM/SMoGLLiM/GLLiM Structure. However, for majority of the data, the squared errors are smaller than those obtained by the dictionary matching method. We figure out that the data with large errors come from the High-peak subgroup and there is no suitable cluster to conducting prediction for these data. For GLLiM/SMoGLLiM/GLLiM Structure, if a suitable cluster for conducting prediction does not exist, the cluster with the closest Mahalanobis distance would be applied for prediction. However, the largest membership posterior probability r_{nkl} among all k, l in Equation (10) would be smaller than the majority of the data. This information could be utilized to identify unreliable prediction results. The worst case of dictionary matching seems to produce smaller prediction error when being compared to other methods. Nevertheless, this is due to the nature of the difference among approaches. The dictionary matching method always predict using values obtained from a member in the dictionary, so that its prediction error cannot go beyond what would be provided by the possible values in the dictionary. This phenomenon does not apply to model-based methods. When prediction is conducted on the data outside of the range of the training dataset, the prediction error could become considerably large, as shown by the outcome of 99 percentiles of prediction squared errors. As a result, even though dictionary matching seems to outperform other model-based methods at these extreme cases, it does not necessary indicates that dictionary method is practically useful for these cases, with the outcomes being so much worse than predicting the rest of the dataset. Our model-based approaches, on the other hand, do have the advantage of identifying these troublesome cases for further considerations.

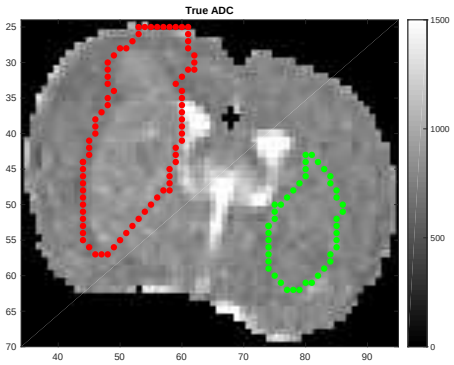
5. Discussion and conclusion

We propose SMoGLLiM as a parsimonious version of GLLiM. SMoGLLiM adopts a two level hierarchical structure of clusters. The associations of observed data are estimated under local cluster level; while the parameters for latent variables and covariance matrices are estimated under global-cluster level to avoid over-parameterization. In addition, we

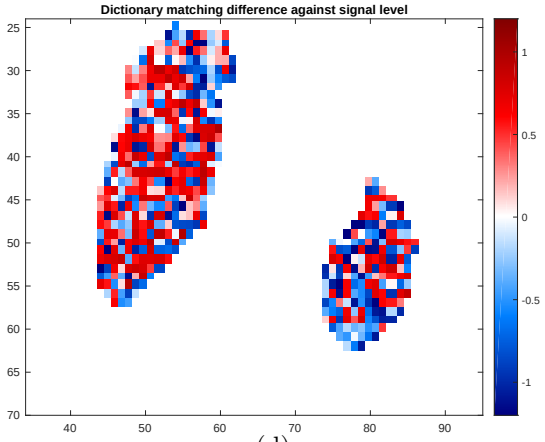
implement a refined, namely constrained with outliers-trimming, version of SMoGLLiM to prevent existence of small clusters and identify outliers. This added refined steps tend to enhance model stability and reduce prediction variation. Jointly selecting suitable cluster-size, K, and dimension of latent W, L_w, under GLLiM could be computationally challenging when a CV search is directly implemented. Our proposal of using BIC to locate a much smaller range for SMoGLLiM to select the global cluster size K and L_w by CV reduces the computational challenges for this task. SMoGLLiM further leads to a post-learning version of GLLiM, called GLLiM structure. By using local means and local variances, while with unfitted points removed, GLLiM structure tends to reach improved empirical performances.

The motivation behind SMoGLLiM and GLLiM Structure is to achieve precise prediction by constructing stable training models. The resulted training model may or may not reflect the exact true model that generates the data, but it captures the key structure and establishes a model that can be stably estimated using the data available. The size and ity of this model would be determined by the data. Finally, the learning procedure could be accelerated by dividing data into groups and adopting parallelization computation technique. We illustrate that this practice is readily accommodated by SMoGLLiMs hierarchical model structure.

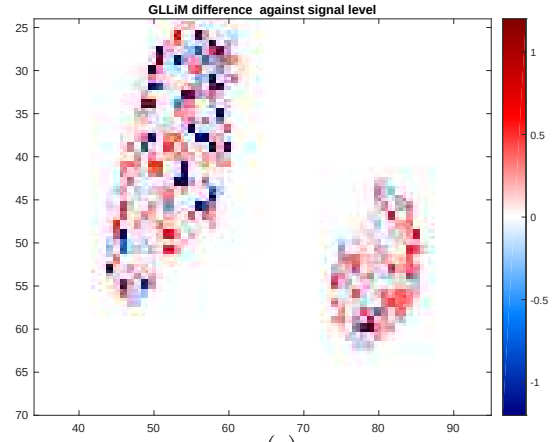
(a)



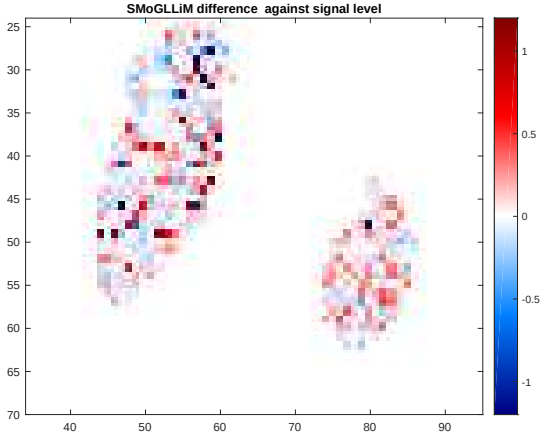
(b)



(c)



(d)



(e)

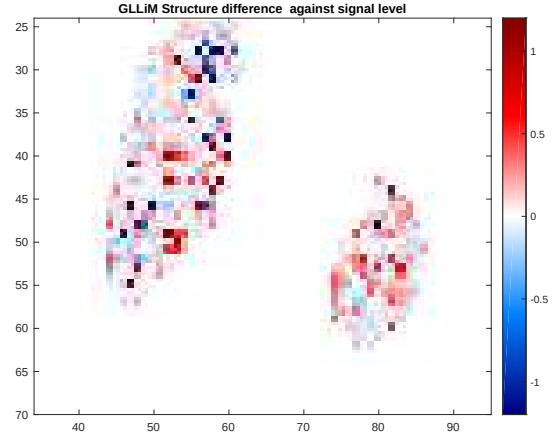


Fig. 6: The true ADC images and the differences between the true values and the predicted values against the signal levels of one animal from the 9L group. Differences are normalized by the average true ADC values in each ROI. (a) The true ADC image. Difference maps between true values and predicted values against the signal levels using either (b) dictionary matching method, (c) GLLiM, (d) SMoGLLiM or (e) GLLiM Structure.

A. Implementation

In this section, an EM algorithm for SMoGLLiM is provided for parameter estimation. We then present an extended version, tailored for ensuring stability and outlier trimming. The selection of tuning parameters is discussed, aiming at obtaining better prediction performances.

A.1. The EM algorithm for SMoGLLiM

Three sets of latent variables are involved in the SMoGLLiM model: $Z_{1:N} = \{Z_N\}_{n=1}^N$, $U_{1:N} = \{U_n\}_{n=1}^N$ and $W_{1:N} = \{W_n\}_{n=1}^N$. The first two variables indicate global and local cluster assignment and the last one is the latent covariate.

The EM algorithm for SMoGLLiM can be divided into several steps: E-Z, U step for estimating the posterior probability of being assigned to a global or a local cluster, E-W step for finding estimation of latent variable W and a maximization step for estimating parameters at the local and global cluster levels.

E-Z, U Step:
We denote the posterior probability of observation n being assigned to global cluster k, local cluster l, based on the observed data, to be

$$r_{nkl} \;=\; p(Z_n = k, U_n = l | t_n, y_n; \theta); \tag{10}$$

and equivalently,

$$r_{nk} \;=\; p(Z_n = k | t_n, y_n; \theta). \tag{11}$$

The posterior probability of sample n being assigned to local cluster (k, l) is given by,

$$\begin{aligned} r_{nkl} &= p(Z_n = k, U_n = l | t_n, y_n; \theta) \\ &= \frac{\rho_{kl} p(t_n, y_n | Z_n = k, U_n = l; \theta)}{\sum_{i=1}^K \sum_{j=1}^M \rho_{ij} p(t_n, y_n | Z_n = i, U_n = j; \theta)}, \end{aligned}$$

where $p(t_n, y_n | Z_n = k, U_n = l; \theta) = p(y_n | t_n, Z_n = k, U_n = l) p(t_n | Z_n = k, U_n = l)$. The first term is given by $p(y_n | t_n, Z_n = k, U_n = l) = \mathcal{N}(y_n; A_{kl}^t t_n + b_{kl} + A_k^w c_k^w, A_k^w \Gamma_k^w A_k^{w\top} + \Sigma_k)$ and recall that the second term $p(t_n | Z_n = k, U_n = l) = \mathcal{N}(t; c_{kl}^t, \Gamma_{kl}^t)$.

A direct derivation shows that

$$\begin{aligned} r_{nk} &= p(Z_n = k | t_n, y_n; \theta) \\ &= \sum_{l=1}^M r_{nkl}. \end{aligned}$$

E-W Step:
The distribution $p(w_n | Z_n = k, t_n, y_n; \theta)$ can be shown to be Gaussian with mean μ_{nk}^w and covariance matrix S_k^w. The estimation of the mean and covariance matrix is given

by:

$$\tilde{\mu}_{nk}^{w} = \sum_{l=1}^{M} \frac{r_{nkl}}{r_{nk}} \tilde{S}_k^w \left(A_k^{w\top} \Sigma_k^{-1} (y_n - A_{kl}^t t_n - b_{kl}) + (\Gamma_k^w)^{-1} c_k^w \right),$$
$$\tilde{S}_k^w = \left\{ (\Gamma_k^w)^{-1} + A_k^{w\top} \Sigma_k^{-1} A_k^w \right\}^{-1}. \tag{12}$$

The maximization step consists of two sub-steps. The first step aims to estimate parameters for Gaussian Mixture Model and the second one focuses on estimating parameters for mapping.

M-GMM Step:

In this step we only consider the parameters related to the Gaussian Mixture Model. In particular, we want to estimate $\{\rho_{kl}, c_{kl}^t, \Gamma_{kl}^t, \}_{k=1,l=1}^{K,M}$. Hereinafter, we let $r_{kl} = \sum_{n=1}^{N} r_{nkl}$ and $r_k = \sum_{n=1}^{N} r_{nk}$. We obtain:

$$\tilde{\rho}_{kl} = \frac{r_{kl}}{N}, \tag{13}$$
$$\tilde{c}_{kl}^t = \frac{\sum_{n=1}^{N} r_{nkl} t_n}{r_{kl}},$$
$$\text{and } \tilde{\Gamma}_{kl}^t = \frac{\sum_{n=1}^{N} r_{nkl} (t_n - \tilde{c}_{kl})(t_n - \tilde{c}_{kl}^t)^\top}{r_{kl}}.$$

M-mapping Step:
The M-Mapping step aims to estimate $\{A_{kl}^t, b_{kl}, A_k^w, \Sigma_k\}_{k=1,l=1}^{K,M}$. It is assumed that T and W are independent given the cluster assignment. Based on this, we could update A_k^w first:

$$\tilde{A}_k^w = \tilde{Y}_k \tilde{V}_k^\top (\tilde{V}_k \tilde{V}_k^\top)^{-1} \tag{14}$$

where

$$\tilde{V}_k = \frac{1}{\sqrt{r_k}} [\sqrt{r_{1k}} (\tilde{\mu}_{1k}^w - \tilde{\mu}_k^w) \ldots \sqrt{r_{Nk}} (\tilde{\mu}_{Nk}^w - \tilde{\mu}_k^w)],$$
$$\tilde{Y}_k = \frac{1}{\sqrt{r_k}} [\sqrt{r_{1k}} (y_1 - \sum_{l=1}^{M} \frac{r_{1kl}}{r_{1k}} \tilde{y}_{kl}) \ldots \sqrt{r_{Nk}} (y_N - \sum_{l=1}^{M} \frac{r_{Nkl}}{r_{Nk}} \tilde{y}_{kl})],$$
$$\tilde{\mu}_k^w = \sum_{n=1}^{N} \frac{r_{nk}}{r_k} \tilde{\mu}_{nk}^w,$$
$$\tilde{y}_{kl} = \sum_{n=1}^{N} \frac{r_{nkl}}{r_{kl}} y_n.$$

Note the difference between how Y and V being centered. For Y, we center it against the local cluster mean, while we let V be centered at the global cluster level.

Once we obtain A_k^w we subtract the latent variables component from Y and update A_{kl}^t and b_{kl}, accordingly. Letting $y_{nk}^* = y_n - \tilde{A}_k^w \tilde{\mu}_{nk}^w$, we get:

$$\tilde{A}_{kl}^t = \tilde{Y}_{kl}^* \tilde{T}_{kl}^\top (\tilde{T}_{kl} \tilde{T}_{kl}^\top)^{-1},$$
$$\tilde{b}_{kl} = \sum_{n=1}^{N} \frac{r_{nkl}}{r_{kl}} (y_{nk}^* - \tilde{A}_{kl}^t t_n),$$

where

$$\tilde{T}_{kl} = \frac{1}{\sqrt{r_{kl}}} [\sqrt{r_{1kl}}(t_1 - \tilde{t}_{kl}) ... \sqrt{r_{Nkl}}(t_N - \tilde{t}_{kl})],$$
$$\tilde{Y}_{kl}^* = \frac{1}{\sqrt{r_{kl}}} [\sqrt{r_{1kl}}(y_{1k}^* - \tilde{y}_{kl}) ... \sqrt{r_{Nkl}}(y_{Nk}^* - \tilde{y}_{kl})],$$
$$\tilde{t}_{kl} = \sum_{n=1}^{N} \frac{r_{nkl}}{r_{kl}} t_n,$$
$$\tilde{y}_{kl} = \sum_{n=1}^{N} \frac{r_{nkl}}{r_{kl}} y_{nk}^*.$$

Finally, we can update Σ_k by:

$$\tilde{\Sigma}_k = \tilde{A}_k^w \tilde{S}_k^w \tilde{A}_k^w + \sum_{n=1}^{N} \frac{r_{nk}}{r_k} [y_n - \sum_{l=1}^{M} \frac{r_{nkl}}{r_{nk}} (\tilde{A}_{kl}^t t_n + \tilde{b}_{kl}) - \tilde{A}_k^w \tilde{\mu}_{nk}^w][y_n - \sum_{l=1}^{M} \frac{r_{nkl}}{r_{nk}} (\tilde{A}_{kl}^t t_n + \tilde{b}_{kl}) - \tilde{A}_k^w \tilde{\mu}_{nk}^w]^\top. \tag{15}$$

A.2. Refined alternative to improve prediction

The EM algorithm as stated in Appendix A.1 gives the key structure of the method. However, implementing it as it is may induce enlarged prediction variation caused by small clusters and could be subject to the influence of outliers. Refining the iterative algorithm, for variation reduction is necessary for stability concerns. The stability can be achieved by constraining the sizes of the clusters (control both covariance volume and prediction variance) and trimming outliers. Each cluster is associated with a cluster weight, which is calculated as $\sum_{n=1}^{N} r_{nkl}$ for cluster (k, l). Each data point in a cluster whose cluster weight is smaller than a pre-determined *minSize* is reassigned to a larger cluster that gives the data point the lowest prediction squared error. The point is kept in that cluster when the prediction squared error is less than a pre-determined *dropThreshold*; otherwise, it would be excluded from the current EM iteration when updating the estimated parameters. The refined algorithm is described as follows:

(a) The algorithm is initialized by adopting the parameters θ, $\tilde{\mu}_{nk}^w$, $\tilde{S}_k^w$ and cluster assignment r_{nkl} obtained from the EM algorithm described in Appendix A.1.

(b) The estimating procedure iterates through the following substeps until the algorithm converges:

(i) Trimming step: In order to remove outliers, we scan through all local clusters and remove all samples whose in-sample prediction squared error are greater than a pre-determined *dropThreshold*. The in-sample prediction squared error for n-th sample is calculated as:

$$E_n^2 = ||t_n^{pred} - t_n||_2^2, \tag{16}$$

where t_n is the true value and t_n^{pred} is the prediction from Equation (4). Note that the low dimension data $\{t_n\}_{n=1}^N$ are standardized before training so that each dimension would be equally weighted. The samples with in-sample prediction squared error larger than *dropThreshold* are considered as outliers and are temporarily removed. Specifically, r_{n^*kl} is set to 0 for all possible combination of (k, l) and n^* is the training samples such that $E_{n^*}^2 > dropThreshold$.

(ii) Maximization step with a cluster size constraint: The estimation of θ is similar to the Maximization step described in Appendix A.1 but with an addition cluster size constraint. Before estimating parameters for each local cluster (k, l), we first check the associated cluster weight. If the cluster weight is smaller than the given $minSize$, we force the training data originally assigned to this cluster to either be assigned to other clusters during the E-Z, U step, or, if no appropriate cluster could be found, be trimmed during the next Trimming Step.

(iii) Update step for the latent variables: Estimation of $\tilde{\mu}_{nk}^w$, $\tilde{S}_k^w$ and r_{nkl} are done using E-W and E-Z, U step described in Appendix A.1.

All outcomes in Section 4 are obtained using the algorithm presented in this section.

A.3. Tuning parameter selection

For SMoGLLiM, there are several user-defined parameters: the dimension of the latent variables L_w, the number of global clusters K, the number of local clusters M, the minimum allowed cluster size $minSize$ and the maximum allowed in-sample prediction error $dropThreshold$. Through the changes of these tuning parameters, the algorithm can be used to analyze all kind of data. We identify default recommendations for certain parameters, that work for almost all cases, and suggest simple procedures that can be used to select others.

- K and L_w: The number of cluster, K, reflects the number of local linear associations between covariates and responses. On the other hand, the number of latent factors, L_w, models the variation that cannot be captured by these linear associations. The combination of (K, L_w) influences the ability of capturing the mean and covariance structure of the relationship between Y and X. Selecting K and L_w through cross-validation is time-consuming, particularly because there could be a large set of potential K to be considered. We propose a method to restrict the searching space via the use of BIC. Using the face dataset as an example, Table 7 shows the cluster number selected using BIC when L_w is fixed; while Table 8 shows the number of latent factors selected by BIC when K is fixed. These two tables show the roles

Table 7: The value of BIC and K selected by BIC for a given L_w. For a fixed L_w, row 1: the minimum value of BIC; and row 2: the number of clusters, K, that achieves this BIC.

	L_w=0	L_w=1	L_w=2	L_w=8	L_w=9	L_w=10
BIC	-8.75e+05	-9.35e+05	-9.48e+05	-1.08e+06	-1.09e+06	-1.11e+06
K	14	13	10	6	6	6

Table 8: The value of BIC and the L_w selected by BIC for a given K. For a fixed K, row 1: the minimum value of BIC; and row 2: the dimension of W, L_w, that achieves this BIC.

	K=5	K=10	K=15	K=20	K=25	K=30	K=35	K=40
BIC	-1.11e+06	-1.03e+06	-9.72e+05	-9.33e+05	-8.90e+05	-8.53e+05	-8.14e+05	-8.09e+05
L_w	10	8	7	3	1	1	0	0

played by K and L_w as how they compensate each other. The model ity increases as we increase K or L_w. Therefore, BIC prefers the combination of either a small K with a large L_w or a large K with a small L_w. It is also known that BIC is conservative, thus the parameters are most likely underestimated. Though it matters less here, with additional sub-clustering steps in SMoGLLiM, we slightly adjust the K and L_w selected by BIC to improve prediction performance. We construct a search grid of K and L_w described as follows. First, we select K using BIC under a small L_w. This cluster number is called K^{BIC}. Next, we fix the cluster number to K^{BIC} and select the corresponding number of latent factors, $L_w^{K^{BIC}}$. To identify the possible range of K and L_w, we increase the cluster number and select the corresponding number of latent factors. As an example, we could set the cluster number as $K^{BIC}+15$ and find the corresponding number of latent factors, $L_w^{K^{BIC}+15}$, again by BIC. Note that $L_w^{K^{BIC}+15}$ is smaller than $L_w^{K^{BIC}}$. If not, we can use $K = K^{BIC}+20$ or even $K^{BIC}+25$, until the resulting L_w is smaller than $L_w^{K^{BIC}}$ and this K would be the upper bound we used for values of K. Applying an equivalent consideration of preventing being too conservative, we could extend the search range of $L_w^{K^{BIC}}$ to $L_w^{K^{BIC}}+2$. Finally, a cross-validation is conducted within the range of $(K^{BIC}, K^{BIC}+15)$ and $(L_w^{K^{BIC}+15}, L_w^{K^{BIC}}+2)$ for searching the combination of K and L_w that achieves the best performance.

- M: It is assumed that there would be one or more local clusters within each global cluster. The choice of M depends on how the data structure is. We found that the final result would not be sensitive to M; the EM algorithm combined with the refining algorithm would adjust itself and unneeded local clusters would be dissolved.

- *minSize*: A two dimension grid search can be used to search for the best combination of *minSize* and *dropThreshold* and we have explored this option. However, this practice could be time-consuming. To obtain an appropriate suggested value for *minSize* we calculate the matrix volume of Γ_{kl}^*, the covariance matrix used in prediction, and look for the drop off. Using the face dataset as an example, we

implement SMoGLLiM with $K = 15$, $M = 5$ and set $L_w = 2$. The volume of Γ^*_{kl} is approximated by the product of top three eigenvalues. Figure 7 shows the relationship between volumes of Γ^*_{kl} versus cluster sizes. A small covariance matrix is likely to cause a surge in likelihood and difficulties for nearby testing sample to be classified as a member of the cluster, both lead to inflation of the prediction MSE. Figure 7 suggests that small covariance matrices could be expected when the cluster size is smaller than 4. In view of this, we set $minSize = 5$ for this case. Our empirical experiences imply that this simple approach leads to comparable outcomes to the more complicated two-dimensional grid search algorithm.

- $dropThreshold$: As $minSize$ being fixed, $dropThreshold$ could be simply estimated by a K-fold cross-validation. From the experimental results, we establish that the prediction MSE is not sensitive to the choice of $dropThreshold$ within a reasonable range. We show this using the outcomes in Section 4.

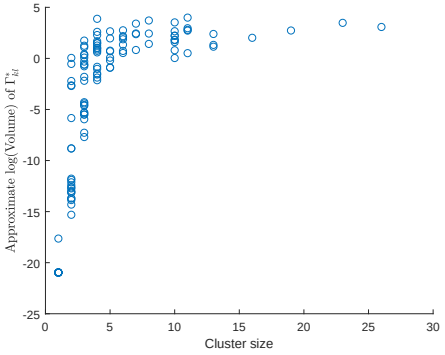


Fig. 7: The logarithm of approximated volume of Γ^*_{kl} against the cluster size.

B. The distribution of microvascular parameters in the synthetic fingerprint dataset and group separation

In Table 9, we summarize the values and the range of the microvascular parameters ($t_1 \sim t_6$) of the fingerprint dictionary. The values for each parameter are shown in Figure 8. There are 1,383,648 observations in the dictionary. The dataset is divided to cover as many kinds of data as possible for cross-validation purpose. First, we use t_6 to form Group 1 ($t_6 = 1$) and Group 2 ($t_6 = 2$). Our exploratory analysis shows the high ity when $t_6 = 3$. Thus, it is necessary to separate more groups on $t_6 = 3$ to reflect the ity. For data with $t_6 = 3$, we divide t_1 into 3 categories and consider 6 different values in t_5. All together, for $t_6 = 3$, we construct 18 groups (Group 3 to Group 20). The available size of each group is shown in Table 10.

To construct a 20-fold cross-validation, the testing sample size is picked so that all data within the smallest group would be used. The smallest group size is 2030 (Group 3 to Group 14). Within these groups, we select 102 testing samples from each group. Some data could have replicates but the number of replicates would be no more than 2. This

Table 9: The unique values and the range of microvascular parameters

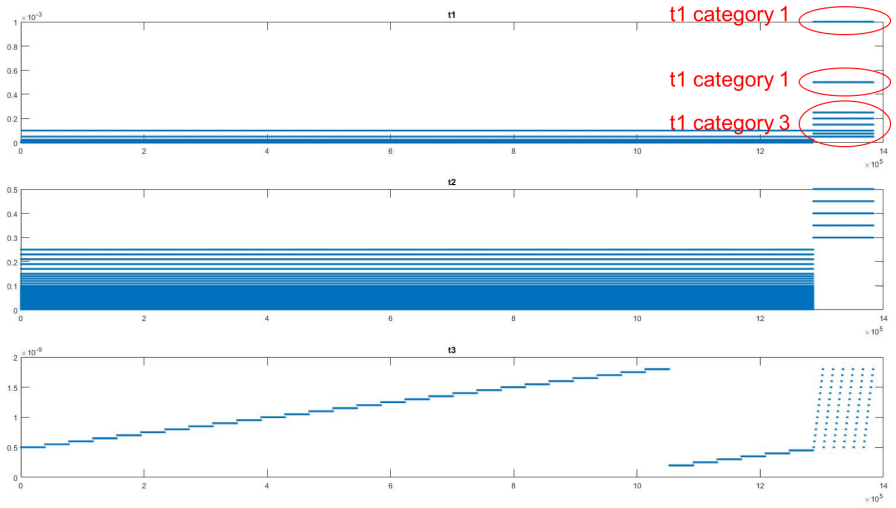
Parameter	Parameter meaning	No. of unique values	Range
t_1	R (μm)	38	$0.5 \sim 1000$
t_2	BV (%)	47	$0.25 \sim 50$
t_3	ADC ($\mu m \cdot s^{-1}$)	33	$2 \times 10^{-10} \sim 18 \times 10^{-10}$
t_4	DeltaChi (ppm)	29	$0 \sim 1.4$
t_5	Direction (radians)	6	0, 0.314, 0.628, 0.943, 1.257, 1.571
t_6	Geometry	3	1, 2, 3

Table 10: The size of each group.

Group ID	Value	Available Size	Group ID	Value	Available Size
Group1	t6=1	1052352	Group11	t1category2, t5value3	2030
Group2	t6=2	233856	Group12	t1category2, t5value4	2030
Group3	t1category1, t5value1	2030	Group13	t1category2, t5value5	2030
Group4	t1category1, t5value2	2030	Group14	t1category2, t5value6	2030
Group5	t1category1, t5value3	2030	Group15	t1category3, t5value1	12180
Group6	t1category1, t5value4	2030	Group16	t1category3, t5value2	12180
Group7	t1category1, t5value5	2030	Group17	t1category3, t5value3	12180
Group8	t1category1, t5value6	2030	Group18	t1category3, t5value4	12180
Group9	t1category2, t5value1	2030	Group19	t1category3, t5value5	12180
Group10	t1category2, t5value2	2030	Group20	t1category3, t5value6	12180

aims to make the number consistent through all groups and folds. After excluding testing data, we would randomly pick 10,000 for Group 1 and Group 2 as training samples. For Group 3 to Group 14, the remaining 1928 samples would become the training data. For Group 15 to Group 20, we would pick 2000 training samples. As a result, within each fold, there would be 55136 training samples (10,000 from Group 1 and Group 2, 1928 from Group 3 to Group 14, 2000 from Group 15 to Group 20) and 2040 testing data (102 from each group).

(a)



(b)

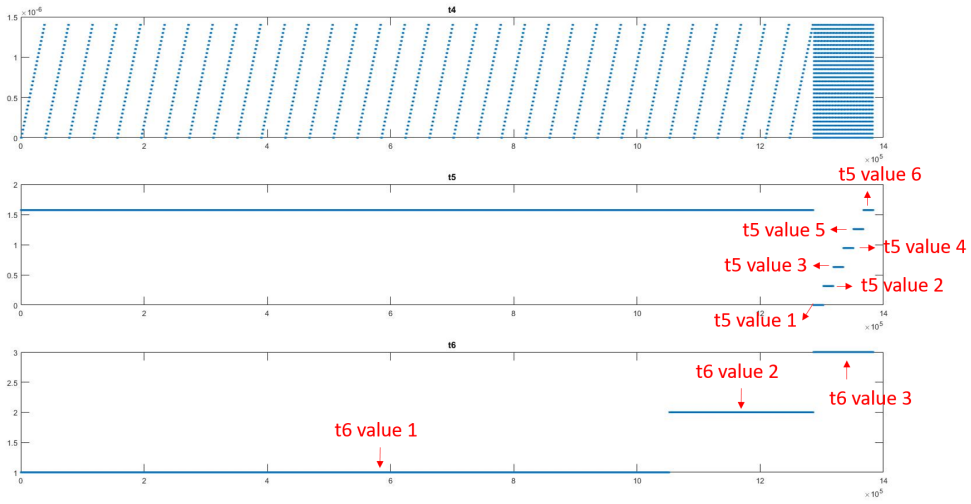


Fig. 8: The distribution of parameters (T). The x-axis shows the index of observations and y-axis marks the values of each observation in different dimensions. (a) Dimension 1 to 3; (b) Dimension 4 to 6.

Table 11: The 50%, 90% and 99% quantiles of squared error using different methods. The models are built upon 3 microvascular parameters: Radius, BVf and DeltaChi.

	Dictionary Matching			GLLiM			SMoGLLiM			GLLiM Structure		
	50%	90%	99%	50%	90%	99%	50%	90%	99%	50%	90%	99%
Radius	0.2144	69.3636	82.5297	$< 10^{-4}$	0.2916	21.44	$< 10^{-4}$	0.2144	21.44	$< 10^{-4}$	0.2144	21.44
BVf	$< 10^{-4}$	0.2277	0.2277	$< 10^{-4}$	$< 10^{-4}$	0.0261	$< 10^{-4}$	$< 10^{-4}$	0.0068	$< 10^{-4}$	$< 10^{-4}$	0.0269
DeltaChi	$< 10^{-4}$	0.0571	0.7000	$< 10^{-4}$	0.0108	0.6385	$< 10^{-4}$	0.0013	0.2012	$< 10^{-4}$	0.0005	0.3158

C. The cross-validation results under 3 microvascular parameters for synthetic fingerprint dataset

It is noticed that adding weakly informative parameter, such as ADC, in the model would downgrade the prediction performance. If predicting ADC is not the major task, we could obtain lower prediction error when training the model with Radius, BVf and DeltaChi. The results of using 3 parameters are shown in Table 11.

On the other hand, when adopting dictionary matching, testing data are compared to fingerprint observations, which are associated to 6 parameters as shown in Figure 8. With all parameters embedded inside fingerprint observations, the dictionary matching method is actually using information from 6 parameters. If we restrict the parameter space, i.e. only consider Radius, BVf and DeltaChi, there would be multiple fingerprints associated to the same set of the restricted parameters. To evaluate the performance under restricted parameter setting, we randomly select a fingerprint as the representative for the same set of parameters. Table 11 shows the cross-validation results on the restricted synthetic fingerprint dataset.

Comparing Table 11 to Table 4, we observe improvement on 90% quantiles for model-based methods, which indicates that we could obtain better prediction outcomes by removing ADC from the training data. On the contrary, the results of dictionary matching method become worse. This is a natural consequence of lacking sufficient details to categorizing and distinguishing samples in the dictionary. If the remaining parameters are insufficient to reflect the data ity, it is likely to match a testing data to an inadequate member within the dictionary and, as a result, we would obtain a large prediction error. This comparison shows the difference between the dictionary matching method and the model-based method. For dictionary matching method we hope to enumerate all possible distinction in the dictionary. Thus, the prediction performance deteriorates when this goal cannot be achieved. However, this may not apply to model-based methods, where the most appropriate model among the ones being considered is used to conduct prediction. The performance could improve when weakly informative parameter covariates are removed.

References

Banfield, J. D. and A. E. Raftery (1993). Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics 49*(3), 803–821.

Bouveyron, C., S. Girard, and C. Schmid (2007, September). High-dimensional data clustering. *Computational Statistics & Data Analysis 52*(1), 502–519.

De Veaux, R. D. (1989, November). Mixtures of linear regressions. *Computational Statistics & Data Analysis 8*(3), 227–245.

Deleforge, A., F. Forbes, and R. Horaud (2015). High-dimensional regression with gaussian mixtures and partially-latent response variables. *Statistics and Computing 25*(5), 893–911.

Elisseeff, A. and J. Weston (2001). A kernel method for multi-labelled classification. In *NIPS*, Volume 14, pp. 681–687.

Fraley, C. and A. E. Raftery (2002, June). Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association 97*(458), 611–631.

Frhwirth-Schnatter, S. (2006, November). *Finite Mixture and Markov Switching Models*. Springer Science & Business Media.

Gershenfeld, N. (1997, January). Nonlinear Inference and Cluster-Weighted Modeling. *Annals of the New York Academy of Sciences 808*(1), 18–24.

Goldfeld, S. M. and R. E. Quandt (1973, March). A Markov model for switching regressions. *Journal of Econometrics 1*(1), 3–15.

Hennig, C. (2000, July). Identifiablity of Models for Clusterwise Linear Regression. *Journal of Classification 17*(2), 273–296.

Kotz, S. and S. Nadarajah (2004). *Multivariate t-distributions and their applications*. Cambridge University Press.

Lemasson, B., N. Pannetier, N. Coquery, L. S. B. Boisserand, N. Collomb, N. Schuff, M. Moseley, G. Zaharchuk, E. L. Barbier, and T. Christen (2016, November). MR Vascular Fingerprinting in Stroke and Brain Tumors Models. *Scientific Reports 6*, 37071.

Ma, D., V. Gulani, N. Seiberlich, K. Liu, J. L. Sunshine, J. L. Duerk, and M. A. Griswold (2013, March). Magnetic Resonance Fingerprinting. *Nature 495*(7440), 187–192.

McLachlan, G. and D. Peel (2000). Mixtures of Factor Analyzers. In *Finite Mixture Models*, pp. 238–256. John Wiley & Sons, Inc. DOI: 10.1002/0471721182.ch8.

Perthame, E., F. Forbes, and A. Deleforge (2018, January). Inverse regression approach to robust nonlinear high-to-low dimensional mapping. *Journal of Multivariate Analysis 163*(Supplement C), 1–14.

Städler, N., P. Bühlmann, and S. van de Geer (2010, August). L1-penalization for mixture regression models. *TEST 19*(2), 209–256.

Subedi, S., A. Punzo, S. Ingrassia, and P. D. McNicholas (2013, March). Clustering and classification via cluster-weighted factor analyzers. *Advances in Data Analysis and Classification 7*(1), 5–40.

Tenenbaum, J. B., V. De Silva, and J. C. Langford (2000). A global geometric framework for nonlinear dimensionality reduction. *science 290*(5500), 2319–2323.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Wu, H.-M. (2012). Kernel sliced inverse regression with applications to classification. *Journal of Computational and Graphical Statistics*.

Xu, L., M. I. Jordan, and G. E. Hinton (1994). An Alternative Model for Mixtures of Experts. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS'94, Cambridge, MA, USA, pp. 633–640. MIT Press.

Yi, X. and C. Caramanis (2015). Regularized EM Algorithms: A Unified Framework and Statistical Guarantees. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28*, pp. 1567–1575. Curran Associates, Inc.