Prediction with High-Dimensional Regression via Hierarchically Structured Gaussian Mixtures and Latent Variables †

Chun-Chen Tu
University of Michigan, Ann Arbor, USA
Florence Forbes
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP‡, LJK, 38000 Grenoble, France
Benjamin Lemasson
Univ. Grenoble Alpes, Inserm, U1216, GIN, F-38000, Grenoble, France.
and Naisyin Wang
University of Michigan, Ann Arbor, USA

Appendices

A. Implementation Details

In this section, we provide the exact expressions of multiple elements in the proposed method as well as an EM algorithm to estimate the parameters in HGLLiM. The procedures to select tuning parameters, aiming at enhancing implementation feasibility, model stability and flexibility are described afterwards.

In Section **??**, we built the key prediction procedures based on an inverse conditional density,

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

with the indices k and ℓ representing the global and local cluster memberships, respectively, and a forward regression model,

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

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

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

† *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
‡ Institute of Engineering Univ. Grenoble Alpes

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

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

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

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

As described in Section **??**, the low-dimensional data Y could contain a latent component W. That is, the HGLLiM model considers three sets of latent variables: $Z_{1:N} = \{Z_N\}_{n=1}^N$, $U_{1:N} = \{U_n\}_{n=1}^N$ and $W_{1:N} = \{W_n\}_{n=1}^N$, where Z and U indicate global and local cluster assignment. We use the EM algorithm described in the next sub-section to obtain estimates of θ, which can be directly converted to θ^* in prediction.

A.1. The EM algorithm for HGLLiM

The EM algorithm for HGLLiM 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

$$r_{nkl} \;=\; p(Z_n = k, U_n = l|t_n, x_n; \theta); \tag{2}$$

and we let,

$$r_{nk} \;=\; p(Z_n = k|t_n, x_n; \theta). \tag{3}$$

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

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

where $p(t_n, x_n|Z_n = k, U_n = l; \theta) = p(x_n|t_n, Z_n = k, U_n = l)p(t_n|Z_n = k, U_n = l)$. The first term is given by $p(x_n|t_n, Z_n = k, U_n = l) = \mathcal{N}(x_n; A^t_{kl}t_n + b_{kl} + A^w_k c^w_k, A^w_k\Gamma^w_k A^{w\top}_k + \Sigma_k)$ and recall that the second term $p(t_n|Z_n = k, U_n = l) = \mathcal{N}(t; c^t_{kl}, \Gamma^t_{kl})$.

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 $p(w_n|Z_n = k, t_n, x_n; \theta)$ can be shown to be Gaussian with mean μ^w_{nk} and covariance matrix S^w_k. The estimation of the mean and covariance matrix is given by:

$$\begin{aligned}\tilde{\mu}^w_{nk} &= \sum_{l=1}^{M} \frac{r_{nkl}}{r_{nk}} \tilde{S}^w_k \left(A^{w\top}_k \Sigma^{-1}_k (x_n - A^t_{kl} t_n - b_{kl}) + (\Gamma^w_k)^{-1} c^w_k \right), \\ \tilde{S}^w_k &= \left\{ (\Gamma^w_k)^{-1} + A^{w\top}_k \Sigma^{-1}_k A^w_k \right\}^{-1}. \end{aligned} \tag{4}$$

The maximization step consists of two sub-steps. The first step aims to estimate parameters for a 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 $\{\rho_{kl}, c^t_{kl}, \Gamma^t_{kl}, \}^{K,M}_{k=1,l=1}$. Hereinafter, we let $r_{kl} = \sum_{n=1}^{N} r_{nkl}$ and $r_k = \sum_{n=1}^{N} r_{nk}$. We obtain:

$$\begin{aligned} \tilde{\rho}_{kl} &= \frac{r_{kl}}{N}, \\ \tilde{c}^t_{kl} &= \frac{\sum_{n=1}^{N} r_{nkl} t_n}{r_{kl}}, \\ \text{and } \tilde{\Gamma}^t_{kl} &= \frac{\sum_{n=1}^{N} r_{nkl} (t_n - \tilde{c}_{kl})(t_n - \tilde{c}^t_{kl})^\top}{r_{kl}}. \end{aligned} \tag{5}$$

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

$$\tilde{A}^w_k = \tilde{X}_k \tilde{V}_k^\top (S^w_k + \tilde{V}_k \tilde{V}_k^\top)^{-1} \tag{6}$$

where

$$\begin{aligned} \tilde{V}_k &= \frac{1}{\sqrt{r_k}} [\sqrt{r_{1k}} (\tilde{\mu}^w_{1k} - \tilde{\mu}^w_k) ... \sqrt{r_{Nk}} (\tilde{\mu}^w_{Nk} - \tilde{\mu}^w_k)], \\ \tilde{X}_k &= \frac{1}{\sqrt{r_k}} [\sqrt{r_{1k}} (x_1 - \sum_{l=1}^{M} \frac{r_{1kl}}{r_{1k}} \tilde{x}_{kl}) ... \sqrt{r_{Nk}} (x_N - \sum_{l=1}^{M} \frac{r_{Nkl}}{r_{Nk}} \tilde{x}_{kl})], \\ \tilde{\mu}^w_k &= \sum_{n=1}^{N} \frac{r_{nk}}{r_k} \tilde{\mu}^w_{nk}, \\ \tilde{x}_{kl} &= \sum_{n=1}^{N} \frac{r_{nkl}}{r_{kl}} x_n. \end{aligned}$$

Note the difference between how X and V are being centered. For X, 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 X and update A_{kl}^t and b_{kl}, accordingly. Letting $x_{nk}^* = x_n - \tilde{A}_k^w \tilde{\mu}_{nk}^w$, we get:

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

where

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

Finally, we can update Σ_k by:

$$\tilde{\Sigma}_k = \tilde{A}_k^w \tilde{S}_k^w \tilde{A}_k^w + \sum_{n=1}^{N} \frac{r_{nk}}{r_k} [x_n - \sum_{l=1}^{M} \frac{r_{nkl}}{r_{nk}} (\tilde{A}_{kl}^t t_n + \tilde{b}_{kl}) - \tilde{A}_k^w \tilde{\mu}_{nk}^w][x_n - \sum_{l=1}^{M} \frac{r_{nkl}}{r_{nk}} (\tilde{A}_{kl}^t t_n + \tilde{b}_{kl}) - \tilde{A}_k^w \tilde{\mu}_{nk}^w]^\top. \quad (7)$$

A.2. Tuning parameter selection

For HGLLiM, 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 clusters, 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 X and Y. 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

A.0 shows the cluster number selected using BIC when L_w is fixed; while Table A.0 shows the number of latent factors selected by BIC when K is fixed. These two tables show the roles played by K and L_w as how they compensate each other. The model complexity 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 HGLLiM, 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 use 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 cross-validation algorithm 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 HGLLiM 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 A.1 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 A.1 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

Table A.0: 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 A.0: 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

prediction MSE is not sensitive to the choice of *dropThreshold* within a reasonable range. We show this using the outcomes in Section **??**.

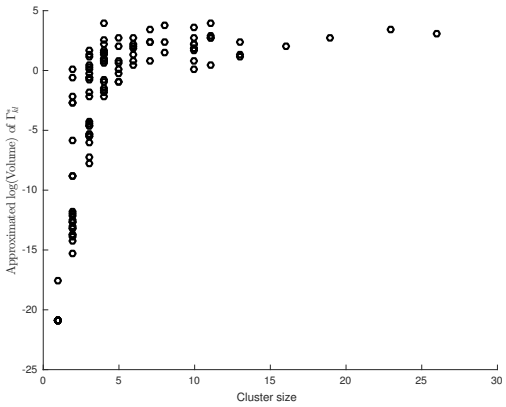


Fig. A.1: The logarithm of approximated volume of Γ_{kl}^* against the cluster size.

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

In Table B.1, 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 B.2. 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 complexity when $t_6 = 3$. Thus, it is necessary to separate more groups on $t_6 = 3$ to reflect the complexity. 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 B.1.

Table B.1: Numbers of unique values and range of the 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 B.1: 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

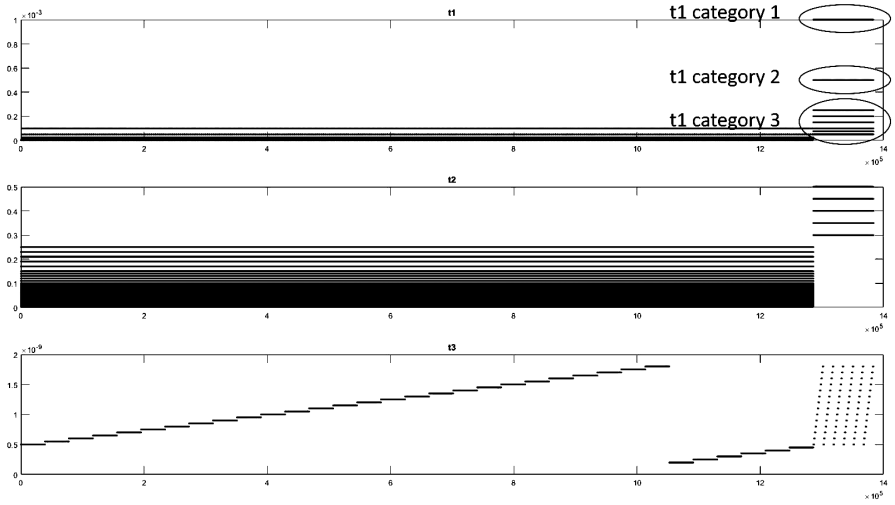
To construct a 20-fold cross-validation, the testing sample size is picked so that all data within the smallest group would be used. The smallest group size is 2030 (Group 3 to Group 14). Within these groups, we select 102 testing samples from each group. Some data could have replicates but the number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

C. Parallel processing of the fingerprint dataset

Model building time is a critical issue for large and complex datasets. As the number of samples increases, the time of computing posterior probabilities in the E-step increases. In addition, it takes longer for the EM algorithm to converge and it is more difficult to find a proper initial setting. To speed up the computation, we can take advantage of the hierarchical structure of HGLLiM. The model building step can be accelerated by subsetting the dataset into smaller groups and applying HGLLiM 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

(a)



(b)

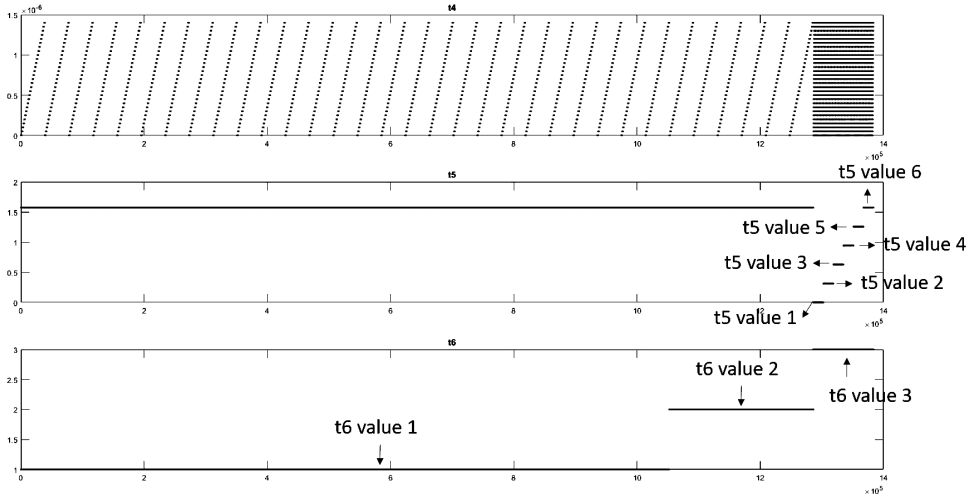


Fig. B.2: The distribution of parameters (T). The x-axis shows the index of observations and y-axis marks the values of each observation in different dimensions. (a) Dimension 1 to 3; (b) Dimension 4 to 6.

levels). In addition, we observe the peak value in the fingerprint signal 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 5.

Training is performed within each group separately. In each group, the cluster mem-

berships are estimated and model parameters are iteratively updated based on the cluster memberships, 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 global latent cluster indicator, $Z^* = (G, Z)$, where G indicates the group assignment and Z is the original global-cluster assignment within group G. By replacing Z with Z^*, the inverse parameter vector, θ, aggregates all inverse parameters from different groups. The forward regression model parameter vector, θ^*, can be updated accordingly using the equations described in Section **??**. 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 to use 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 point was originally assigned to group B. To investigate influence of incorrect group assignment, we calculate the posterior probability of each data point being assigned to each group. The group with the highest 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 HGLLiM 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.

For HGLLiM 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 and show that the model-based methods can achieve comparable results

D. The cross-validation results for the synthetic fingerprint dataset

In this section we evaluate and compare the performance of different methods on the synthetic fingerprint dataset 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.

Table D.2 shows the 50%, 90% and 99% quantiles of the prediction squared errors for different parameters using different methods through cross-validation. We observe that the prediction is close to the true value for 90% of the predicted values. Using GLLiM, we obtain slightly larger 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 better than those of other parameters, with the relationship between BVf and Y being the strongest among all parameters. For

Table D.2: The 50%, 90% and 99% quantiles of prediction squared errors using different methods. The models are built upon 3 microvascular parameters: Radius, BVf and DeltaChi.

	Dictionary Matching			GLLiM			HGLLiM			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

DeltaChi, the E^2's 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 HGLLiM 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, 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 D.2.

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 B.2. 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 D.2 shows the cross-validation results on the restricted synthetic fingerprint dataset.

Comparing Table D.2 to Table D.2, 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 complexity, 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.

Table D.2: The 50%, 90% and 99% quantiles of prediction squared errors using different methods.

	Dictionary matching			GLLiM			HGLLiM			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