H-Likelihood Approach to Deep Neural Networks with Temporal-Spatial Random Effects for High-Cardinality Categorical Features

Hangbin Lee [*1] **Youngjo Lee** [*1]

Abstract

Deep Neural Networks (DNNs) are one of the most powerful tools for prediction, but many of them implicitly assume that the data are statistically independent. However, in the real world, it is common for large-scale data to be clustered with temporal-spatial correlation structures. Variational approaches and integrated likelihood approaches have been proposed to obtain approximate maximum likelihood estimators (MLEs) for correlated data. However, due to the large size of data, they cannot provide exact MLEs. In this study, we propose a new hierarchical likelihood approach to DNNs with correlated random effects for clustered data. By jointly optimizing the the negative h-likelihood loss, we can provide exact MLEs for both mean and dispersion parameters, as well as the best linear unbiased predictors for the random effects. Moreover, the hierarchical likelihood allows a computable procedure for restricted maximum likelihood estimators of dispersion parameters. The proposed two-step algorithm enables online learning for the neural networks, whereas the integrated likelihood cannot decompose like a widely-used loss function in DNNs. The proposed h-likelihood approach offers several advantages, which we demonstrate through numerical studies and real data analyses.

1. Introduction

Deep neural network (DNN) models have served as the method of learning highly nonlinear relationship between the input and output variables with strong prediction performance (LeCun et al., 2015; Goodfellow et al., 2016). However, most DNN models implicitly assume independence of the data and ignore underlying correlation structures, despite large-scale data in the real world often being clustered by multiple categorical features. Recently, there have been emerging attempts to enhance the prediction for clustered data by introducing the random effects into the DNN models (Tran et al., 2020; Mandel et al., 2021; Simchoni & Rosset, 2021; 2022).

Simchoni & Rosset (2021; 2022) proposed linear mixed model neural network (LMMNN) models with single independent random effects and extended LMMNN models to multiple random effects allowing temporal-spatial correlation structure. However, their conventional integrated likelihood approach is computationally intractable because it does not allow decomposition like an ordinary loss function in DNNs. They proposed the use of block-diagonal approximation to the covariance matrix to obtain approximate maximum likelihood estimators (MLEs) for their LMMNN models. However, their approximate likelihood can give a severe bias in parameter estimation for models with correlated random effects. Also, this difficulty prevents them from obtaining restricted maximum likelihood estimators (REMLEs) for LMMNN models. Variational approach can be an alternative. However, this cannot provide exact MLEs either but only approximate MLEs.

Lee & Nelder (1996) proposed the use of h-likelihood as an extension of classical likelihood for statistical models with random effects. In LMMs, the h-likelihood is Henderson's joint likelihood (Henderson et al., 1959) of which the joint maximization gives the MLEs for fixed effects and the best linear unbiased predictors (BLUPs) for random effects. However, it does not give MLEs for variance components by a simple joint maximization. This causes the computational difficulty of Simchoni & Rosset (2021; 2022). In this paper, we introduce the new h-likelihood for LMMNN models with various temporal-spatial random effects from the multiple categorical features. The proposed negative h-likelihood serves as a loss function, which allows the exact MLEs for all fixed parameters and BLUPs for random effects. The proposed negative h-likelihood for LMMNN models allows the highly non-linear functions of input variables and multiple random effects with complex covariance structures, which is the key to overcoming the computational difficulties in

[1] Department of Statistics, Seoul National University, Seoul 08826, Republic of Korea. Correspondence to: Youngjo Lee <youngjo@snu.ac.kr>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

LMMNN models.

In Section 2, we briefly review the integrated likelihood approach to LMMs. In Section 3, the h-likelihood for LMMs with multiple random effects is proposed. It is worth emphasizing that its simple joint maximization can give the MLEs for the whole fixed parameters and BLUPs for random effects, and bypasses the heavy computation difficulties to obtain the exact MLEs. In Section 4, we propose the use of negative h-likelihood as a loss function of LMMNN models and introduce a useful adjustment for random effect predictions. This allows online learning algorithm. To compare with the existing methods, we provide simulation studies in Section 5 and real data analyses in Section 6, followed by concluding remarks in Section 7. All the proofs and technical details are in Appendix.

2. Integrated Likelihood Approach for LMMs

Let $\mathbf{y}$ be a vector of N responses, $\mathbf{X}$ and $\mathbf{Z}$ be $N \times p$ and $N \times q$ model matrices for fixed effects $\boldsymbol{\beta} \in \mathbb{R}^p$ and random effects $\mathbf{v} \in \mathbb{R}^q$, respectively. We start with a standard LMM,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{v} + \mathbf{e}$$

where $\mathbf{e} \sim N(\mathbf{0}, \sigma_e^2 \mathbf{I}_N)$ is a vector of N random noises, $\mathbf{v} \sim N(\mathbf{0}, \mathbf{D})$ is a vector of q random effects, $\mathbf{I}_N$ is $N \times N$ identity matrix and $\mathbf{D} = \mathbf{D}(\boldsymbol{\lambda})$ is $q \times q$ covariance matrix parameterized by $\boldsymbol{\lambda}$. Let $\boldsymbol{\psi} = \left(\sigma_e^2, \boldsymbol{\lambda}\right)$ be the vector of dispersion parameters and $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\psi})$ be the vector of whole fixed parameters. To obtain the estimates for $\boldsymbol{\beta}$ and $\mathbf{v}$, Henderson et al. (1959) proposed to maximize the Henderson's joint likelihood,

$$\begin{aligned}\mathcal{J}(\boldsymbol{\theta}, \mathbf{v}) &= \log f_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{v}) = \log f_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{v}) + \log f_{\boldsymbol{\theta}}(\mathbf{v}) \\ &= -\frac{1}{2\sigma_e^2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{v}||^2 - \frac{N}{2}\log(2\pi\sigma_e^2) \\ &\quad - \frac{1}{2}\mathbf{v}^T\mathbf{D}^{-1}\mathbf{v} - \frac{1}{2}\log|2\pi\mathbf{D}|, \qquad (1)\end{aligned}$$

where $||\cdot||^2$ denotes the L2-norm and $|\cdot|$ denotes the determinant. For given variance components $\boldsymbol{\psi} = \left(\sigma_e^2, \boldsymbol{\lambda}\right)$, optimization of the joint likelihood (1) gives MLEs for $\boldsymbol{\beta}$ and the BLUPs for $\mathbf{v}$,

$$\begin{aligned}\widehat{\boldsymbol{\beta}} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y} - \mathbf{Z}\widehat{\mathbf{v}}), \\ \widehat{\mathbf{v}} &= \widehat{\mathrm{E}}(\mathbf{v}|\mathbf{y}) = (\mathbf{Z}^T\mathbf{Z} + \sigma_e^2\mathbf{D}^{-1})^{-1}\mathbf{Z}^T(\mathbf{y} - \mathbf{Z}^T\mathbf{X}\widehat{\boldsymbol{\beta}}).\end{aligned}$$

However, it cannot give MLEs for the variance components $\boldsymbol{\psi}$. For the MLEs of $\boldsymbol{\psi}$, the integrated likelihood has been used from the multivariate normal distribution of $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{V})$,

$$\begin{aligned}\ell(\boldsymbol{\theta}) &= \log \int \exp\left(\mathcal{J}(\boldsymbol{\theta}, \mathbf{v})\right) d\mathbf{v} \\ &= -\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) - \frac{1}{2}\log|2\pi\mathbf{V}|,\end{aligned}$$

where the marginal covariance matrix $\mathbf{V}$ is

$$\mathbf{V} = \mathbf{V}(\boldsymbol{\psi}) = \mathbf{Z}\mathbf{D}\mathbf{Z}^T + \sigma_e^2\mathbf{I}_N.$$

For given variance components, it is known that the MLEs for $\boldsymbol{\beta}$ from the integrated likelihood $\ell(\boldsymbol{\theta})$ is the same as Henderson's MLE for $\boldsymbol{\beta}$,

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{V}^{-1}\mathbf{y} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y} - \mathbf{Z}\widehat{\mathbf{v}}).$$

In LMMs, MLEs for variance components could be biased in finite sample. To reduce the bias, REMLEs for $\boldsymbol{\psi}$ are often used (Patterson & Thompson, 1971). In LMMs, REMLEs maximize the restricted likelihood,

$$\ell_R(\boldsymbol{\psi}) = \ell(\boldsymbol{\psi}; \widehat{\boldsymbol{\beta}}) - \frac{1}{2}\log|\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X}|, \qquad (2)$$

which is an adjusted profile likelihood (Cox & Reid, 1987; Lee et al., 2017). However, both the integrated likelihood $\ell(\boldsymbol{\theta})$ and the restricted likelihood $\ell_R(\boldsymbol{\psi})$ involve the computation of the inverse of $N \times N$ matrix $\mathbf{V}$. In LMMNNs with single independent random effects of Simchoni & Rosset (2021), $\mathbf{V}$ has a block-diagonal form. This allows computation of exact MLEs. Simchoni & Rosset (2022) noted that $\mathbf{V}$ is not a block-diagonal form in general, even for LMMs with single categorical feature, when the random effects have a complex correlation structure. In order to avoid computing $\mathbf{V}^{-1}$, they proposed the use of block-diagonal approximation to $\mathbf{V}$. However, it requires a rigorous theoretical justification and the resulting approximate MLEs can have severe biases.

Further difficulties arise when the model contains multiple categorical features $\mathbf{Z} = (\mathbf{Z}_1, ..., \mathbf{Z}_K)$ with corresponding random effects $\mathbf{v} = (\mathbf{v}_1, ..., \mathbf{v}_K)$,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{v} + \mathbf{e} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_1\mathbf{v}_1 + \cdots + \mathbf{Z}_K\mathbf{v}_K + \mathbf{e}, \quad (3)$$

where $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{D}_k)$ is q_k-dimensional vector for $k = 1, ..., K$. Simchoni & Rosset (2022) claimed that the use of block-diagonal approximation can avoid heavy computation in the inverse of $N \times N$ matrix. We found that the integrated likelihood can be computed by using the Woodbury formula,

$$\begin{aligned}\mathbf{V}^{-1} &= (\mathbf{Z}\mathbf{D}\mathbf{Z}^T + \sigma_e^2\mathbf{I}_N)^{-1} \\ &= \frac{1}{\sigma_e^2}\left[I_N - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z} + \sigma_e^2\mathbf{D}^{-1})^{-1}\mathbf{Z}^T\right],\end{aligned}$$

and the matrix determinant lemma,

$$\begin{aligned}\log|\mathbf{V}| &= \log|\mathbf{Z}\mathbf{D}\mathbf{Z}^T + \sigma_e^2\mathbf{I}_N| \\ &= \log|\mathbf{Z}^T\mathbf{Z}\mathbf{D} + \sigma_e^2 I_Q| + (N - Q)\log\sigma_e^2,\end{aligned}$$

where $\mathbf{D} = \text{block-diag}(\mathbf{D}_1, ..., \mathbf{D}_K)$. This formulation can reduce the computations of integrated likelihood without any approximations. However, $\mathbf{Z}^T\mathbf{Z}$ is not a block-diagonal matrix when $k \neq 1$. Thus, it still requires heavy computation for every mini-batch. We study how the h-likelihood overcomes the computational difficulties of an integrated likelihood approach.

3. New H-likelihood Approach for LMMs

In Henderson's joint likelihood, $\mathbf{v}$ is additive to the fixed effects $\boldsymbol{\beta}$ in the linear predictor of LMMs

$$\mathrm{E}(\mathbf{y}|\mathbf{v}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{v}.$$

Lee et al. (2017) called the $\mathbf{v}$-scale the weak-canonical scale and Lee & Nelder (1996) proposed the use of Henderson's joint likelihood $\mathcal{J}(\boldsymbol{\theta};\mathbf{v})$ as the h-likelihood for general non-normal models. However, its joint maximization cannot give the MLEs for the variance components, which leads to the use of integrated likelihood. Thus, the key to avoid computational difficulty due to integration is to define a new proper h-likelihood whose joint maximization gives the MLEs for the whole parameters including variance components. We define the h-likelihood for LMMs, which contain the multiple categorical features $\mathbf{Z} = (\mathbf{Z}_1, ..., \mathbf{Z}_K)$ with corresponding random effects $\mathbf{v} = (\mathbf{v}_1, ..., \mathbf{v}_K)$. Since

$$\begin{aligned}\log f_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{v}) + \log f_{\boldsymbol{\theta}}(\mathbf{v}) &= \log f_{\boldsymbol{\theta}}(\mathbf{y},\mathbf{v})\\ &= \log f_{\boldsymbol{\theta}}(\mathbf{v}|\mathbf{y}) + \log f_{\boldsymbol{\theta}}(\mathbf{y}),\end{aligned}$$

let us define the h-likelihood based on the canonical scale of random effects $\mathbf{v}^c$,

$$h = h(\boldsymbol{\theta}, \mathbf{v}^c) = \ell(\boldsymbol{\theta}) + \log f_{\boldsymbol{\theta}}(\mathbf{v}^c|\mathbf{y})$$

where $\ell(\boldsymbol{\theta}) = \log f_{\boldsymbol{\theta}}(\mathbf{y})$ is the integrated likelihood. Given θ, let $\widetilde{\mathbf{v}}^c$ be mode of h. A sufficient condition for $h(\boldsymbol{\theta}, \mathbf{v}^c)$ to be the h-likelihood is that $f_{\boldsymbol{\theta}}(\widetilde{\mathbf{v}}^c|\mathbf{y})$ is free of $\boldsymbol{\theta}$. In Appendix A1, we show that

$$\mathbf{v}^c = \left(\frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z} + \mathbf{D}^{-1}\right)^{\frac{1}{2}}\mathbf{v}$$

is the canonical scale and the resulting predictive likelihood at $\mathbf{v}^c$,

$$\log f_{\boldsymbol{\theta}}(\widetilde{\mathbf{v}}^c|\mathbf{y}) = \log f_{\boldsymbol{\theta}}(\widetilde{\mathbf{v}}|\mathbf{y}) + \log\left|\frac{d\mathbf{v}}{d\mathbf{v}^c}\right| = -\frac{1}{2}\log|2\pi\mathbf{I}_Q|$$

is free of $\boldsymbol{\theta}$. This leads to

$$h(\boldsymbol{\theta}, \widetilde{\mathbf{v}}^c) \propto \ell(\boldsymbol{\theta}),$$

so that the joint maximization of $h(\boldsymbol{\theta}, \mathbf{v})$ gives the MLEs for the whole fixed parameters. Let $h(\boldsymbol{\theta}, \mathbf{v})$ be a reparameterization of $h(\boldsymbol{\theta}, \mathbf{v}^c)$, then the h-likelihood can be expressed as

$$\begin{aligned}h = h(\boldsymbol{\theta}, \mathbf{v}) &= \log f_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{v}) + \log f_{\boldsymbol{\theta}}(\mathbf{v}) + \log\left|\frac{d\mathbf{v}}{d\mathbf{v}^c}\right|\\ &= \mathcal{J}(\boldsymbol{\theta}, \mathbf{v}) - \frac{1}{2}\log\left|\frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z} + \mathbf{D}^{-1}\right|.\end{aligned}$$

Thus, the h-likelihood $h(\boldsymbol{\theta}, \mathbf{v})$ is not proportional to the Henderson's joint likelihood $\mathcal{J}(\boldsymbol{\theta}, \mathbf{v})$ in (1), since $\log|d\mathbf{v}/d\mathbf{v}^c|$ depends upon the variance components. So the h-likelihood is different from the Henderson's joint likelihood. Given $\boldsymbol{\theta}$, the h-likelihood and joint likelihood of $\mathbf{v}$ are proportional. Thus, joint maximization of the h-likelihood provides BLUPs for random effects. With the model (3), the h-likelihood is

$$\begin{aligned}h = h(\boldsymbol{\theta}, \mathbf{v}) = &-\frac{1}{2\sigma_e^2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{v}||^2 - \frac{N}{2}\log\sigma_e^2\\ &-\frac{1}{2}\mathbf{v}^T\mathbf{D}^{-1}\mathbf{v} - \frac{1}{2}\log\left|\frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z}\mathbf{D} + \mathbf{I}_Q\right|.\end{aligned} \tag{4}$$

In Markov random field models or smoothing splines, the precision matrix of the random effects $\mathbf{P}_k = \mathbf{D}_k^{-1}$ are explicitly expressed and in independent random effect models $\mathbf{P}_k = \lambda_k^{-1}\mathbf{I}_{q_k}$. Let $\mathbf{P} = \text{block-diag}(\mathbf{P}_1, ..., \mathbf{P}_K)$. Then the canonical scale $\mathbf{v}^c$ becomes

$$\mathbf{v}^c = \left(\frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z} + \mathbf{P}\right)^{\frac{1}{2}}\mathbf{v}$$

and the h-likelihood becomes

$$\begin{aligned}h = &-\frac{1}{2\sigma_e^2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{v}||^2 - \frac{N}{2}\log\sigma_e^2\\ &-\frac{1}{2}\mathbf{v}^T\mathbf{P}\mathbf{v} + \frac{1}{2}\log|\mathbf{P}| - \frac{1}{2}\log\left|\frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z} + \mathbf{P}\right|,\end{aligned}$$

which does not requires the computation of $\mathbf{D}^{-1}$.

It is worth emphasizing that the h-likelihood approach does not require the inverse of $N \times N$ matrix but only $Q \times Q$ matrix where $Q = \sum_{k=1}^K q_k$. It is often true that $Q \ll N$. When $\sum_{k=2}^K q_K \ll q_1 < N$ and $\mathbf{D}_1 = \lambda_1\mathbf{I}_{q_1}$, it is not necessary to compute the inverse and the determinant of the whole $Q \times Q$ matrix but $(Q - q_1) \times (Q - q_1)$ matrix. In Appendix A2 and A3, we derive the first and the second derivatives of the h-likelihood, which can be obtained without computing the inverse of full $Q \times Q$ matrix directly.

The h-likelihood has advantage over the Henderson's joint likelihood, equivalent to the h-likelihood of Lee & Nelder (1996), in that it is computationally efficient and gives MLEs for all parameters. Given variance components, the joint likelihood and the h-likelihood provides common estimators. Thus, difference is ML estimation of variance components. In Appendix A1, we show that the restricted likelihood (2) is the adjusted profile h-likelihood,

$$\begin{aligned}&\ell_R(\boldsymbol{\psi}) = h_R(\boldsymbol{\psi})\\ &= h(\boldsymbol{\psi}; \widehat{\boldsymbol{\beta}}, \widehat{\mathbf{v}}^c) - \frac{1}{2}\log\left|\frac{1}{\sigma_e^2}\mathbf{X}^T\mathbf{X} - \frac{1}{\sigma_e^4}\mathbf{X}^T\mathbf{Z}\mathbf{A}^{-1}\mathbf{Z}^T\mathbf{X}\right|\end{aligned} \tag{5}$$

where $\mathbf{A} = \frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z} + \mathbf{D}^{-1}$. Since the additional log determinant term involves an inverse of $Q \times Q$ matrix, the REML procedure is computationally harder than the ML procedure.

4. H-likelihood Learning Algorithm for LMMNNs

Following Simchoni & Rosset (2022), we first extend the LMM (3) to the LMMNN with random effects for the multiple categorical features,

$$\mathbf{y} = f(\mathbf{X})\boldsymbol{\beta} + g_1(\mathbf{Z}_1)\mathbf{v}_1 + \cdots + g_K(\mathbf{Z}_K)\mathbf{v}_K + \mathbf{e} \quad (6)$$

where $f : \mathbb{R}^{p^*} \to \mathbb{R}^p$ and $g_k : \mathbb{R}^{q_k^*} \to \mathbb{R}^{q_k}$ are non-linear functions to be estimated by the neural networks, $\mathbf{X}$ and $\mathbf{Z}_k$ are $n \times p^*$ and $n \times q_k^*$ model matrix, respectively. LMMNN allows complex covariance structures of clustered data due to categorical variables, temporal-spatial structures, and combinations of these. Here, $f(\mathbf{X})$ denotes the last hidden layer including the bias node and $\boldsymbol{\beta}$ is the weight vector from the last hidden layer to the output layer.

The extension of the h-likelihood (4) to the proposed model (6) is straightforward. By replacing $\mathbf{X}$ and $\mathbf{Z}_k$ to $f(\mathbf{X})$ and $g_k(\mathbf{Z}_k)$ for $k = 1, ..., K$, respectively, the canonical scale $\mathbf{v}^c = (\mathbf{v}_1^c, ..., \mathbf{v}_K^c)$ is given by

$$\mathbf{v}^c = \left(\frac{1}{\sigma_e^2} g(\mathbf{Z})^T g(\mathbf{Z}) + \mathbf{D}^{-1} \right)^{\frac{1}{2}} \mathbf{v}$$

where $g(\mathbf{Z}) = (g_1(\mathbf{Z}_1), ..., g_K(\mathbf{Z}_K))$. Then, the objective function for training the network is defined by the negative h-likelihood,

$$\begin{aligned} \text{Loss} = -2h = &\frac{1}{\sigma_e^2} \sum_{i=1}^{N} \left[y_i - f(\mathbf{x}_i)^T \boldsymbol{\beta} - g(\mathbf{z}_i)^T \mathbf{v} \right]^2 \\ &+ \sum_{k=1}^{K} \mathbf{v}_k^T \mathbf{D}_k^{-1} \mathbf{v}_k + c(\boldsymbol{\psi}), \end{aligned} \quad (7)$$

where $c(\boldsymbol{\psi}) = \log \left| \sigma_e^{-2} g(\mathbf{Z})^T g(\mathbf{Z}) \mathbf{D} + \mathbf{I}_Q \right| + N \log \sigma_e^2$ is a function of $\boldsymbol{\psi}$ and $g(\mathbf{Z})$ only. Each component of the negative h-likelihood has straight-forward interpretation:

- $\frac{1}{\sigma_e^2} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{v}||^2$ represents the conditional log-density $-2 \log f_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{v})$, which can be decomposed for online learning.

- $\mathbf{v}^T \mathbf{D}^{-1} \mathbf{v}$ represents the log-density $-2 \log f_{\boldsymbol{\theta}}(\mathbf{v})$, which can be viewed as a kernel regularizer for the weights of categorical features.

- The remaining term $c(\boldsymbol{\psi})$ is a function of dispersion parameters, which does not affect learning of mean parameters, i.e., all the weights in neural network and random effects.

Therefore, the h-likelihood loss for LMM can be understood as the sum of the squared loss, kernel regularizer for random effects, and an additional function for yielding MLEs of dispersion parameters.

Let

$$\widehat{y}_i = \mathrm{E}(y_i|\mathbf{v}) = f(\mathbf{x}_i)^T \boldsymbol{\beta} + g(\mathbf{z}_i)^T \mathbf{v},$$

then the loss function becomes

$$\text{Loss} = \sum_{i=1}^{N} \left[\frac{(y_i - \widehat{y}_i)^2}{\sigma_e^2} + \frac{\sum_{k=1}^{K} \mathbf{v}_k^T \mathbf{D}_k^{-1} \mathbf{v}_k}{N} \right] + c(\boldsymbol{\psi}).$$

and its gradient with respect to the mean parameters in $f, g, \boldsymbol{\beta}, \mathbf{v}$ is given by

$$\begin{aligned} \nabla \text{ Loss} &= \nabla \sum_{i=1}^{N} \left[\frac{(y_i - \widehat{y}_i)^2}{\sigma_e^2} + \frac{\sum_{k=1}^{K} \mathbf{v}_k^T \mathbf{D}_k^{-1} \mathbf{v}_k}{N} \right] \\ &\propto \sum_{i=1}^{N} \left[\nabla (y_i - \widehat{y}_i)^2 + \frac{\sigma_e^2}{N} \sum_{k=1}^{K} \nabla \mathbf{v}_k^T \mathbf{D}_k^{-1} \mathbf{v}_k \right], \end{aligned}$$

which does not involve the log-determinant of $Q \times Q$ matrices in $c(\boldsymbol{\psi})$. Note further that the gradient with respect to the random effects is $\nabla_{\mathbf{v}_k} \mathbf{v}_k^T \mathbf{D}_k^{-1} \mathbf{v}_k = 2\mathbf{v}_k / \lambda_k$ when $\mathbf{v}_k$ is independent random effect. Even if every pair of $\mathbf{v}_k$ is correlated, it only involves the inverse of $q_k \times q_k$ matrix. Thus, for given variance components $\boldsymbol{\psi}$, optimization of the negative h-likelihood loss (7) with respect to the mean parameters can naturally decompose for online learning frameworks. Furthermore, it can be interpreted as the optimization of the sum of squared error $\sum_i (y_i - \widehat{y}_i)^2$ with the penalty function $\sum_k \sigma_e^2 \mathbf{v}_k^T \mathbf{D}_k^{-1} \mathbf{v}_k$. In LMMs, MLEs for mean parameters are robust against estimation of dispersion parameters, whereas MLEs for dispersion parameters are sensitive to estimation of mean parameters. Thus, we update the variance components every m epoch, not every mini-batch.

An advantage of the h-likelihood is that it avoids heavy computation in the integrated likelihood. Figure 1 shows our two-step algorithm with the negative h-likelihood loss. The proposed algorithm allows online learning of mean parameters including random effects while saving the computational cost required for estimation of dispersion parameters.

- **M-step:** Update the mean parameters $(f, g, \boldsymbol{\beta}, \mathbf{v})$ in the neural network for every mini-batch.

- **V-step:** Update the variance components in $\boldsymbol{\psi}$ using the whole training data for every m epoch.

Figure 2 shows the MSE vs. time curves of the h-likelihood approach and the improved integrated likelihood approach with the Woodbury formula and the matrix determinant lemma. This assess the relative efficiency of the two methods in terms of computational complexity and accuracy

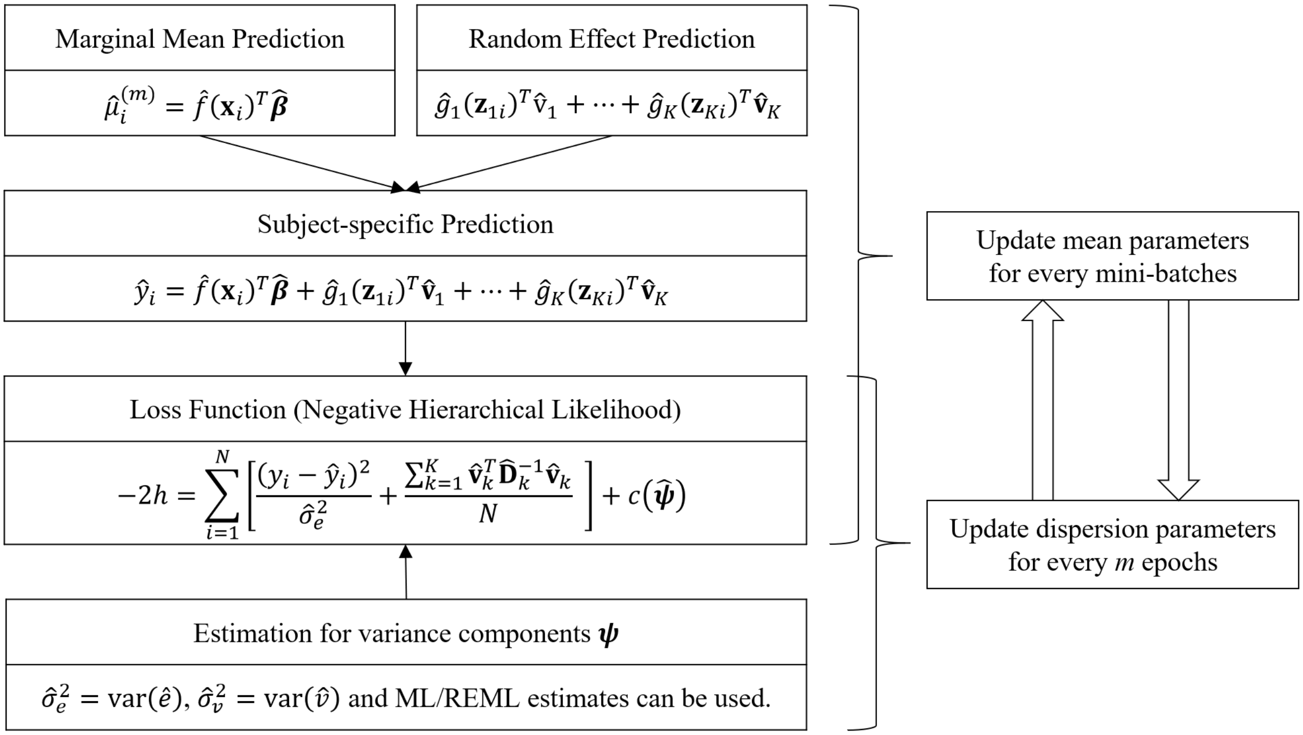
Marginal Mean Prediction
$\hat{\mu}_i^{(m)} = \hat{f}(\mathbf{x}_i)^T\hat{\boldsymbol{\beta}}$
Random Effect Prediction
$\hat{g}_1(\mathbf{z}_{1i})^T\hat{\mathbf{v}}_1 + \cdots + \hat{g}_K(\mathbf{z}_{Ki})^T\hat{\mathbf{v}}_K$
Subject-specific Prediction
$\hat{y}_i = \hat{f}(\mathbf{x}_i)^T\hat{\boldsymbol{\beta}} + \hat{g}_1(\mathbf{z}_{1i})^T\hat{\mathbf{v}}_1 + \cdots + \hat{g}_K(\mathbf{z}_{Ki})^T\hat{\mathbf{v}}_K$
Loss Function (Negative Hierarchical Likelihood)
$-2h = \sum_{i=1}^{N}\left[\frac{(y_i - \hat{y}_i)^2}{\hat{\sigma}_e^2} + \frac{\sum_{k=1}^{K}\hat{\mathbf{v}}_k^T\hat{\mathbf{D}}_k^{-1}\hat{\mathbf{v}}_k}{N}\right] + c(\hat{\boldsymbol{\psi}})$
Estimation for variance components $\boldsymbol{\psi}$
$\hat{\sigma}_e^2 = \text{var}(\hat{e})$, $\hat{\sigma}_v^2 = \text{var}(\hat{v})$ and ML/REML estimates can be used.
Update mean parameters for every mini-batches
Update dispersion parameters for every m epochs

Figure 1. A sketch of the proposed model fitting algorithm via h-likelihood.

(MSE). The MSE of the h-likelihood approach (blue) decreased more rapidly than that of the integrated likelihood approach (red). These results provide evidence that the proposed h-likelihood approach is computationally more efficient than the integrated likelihood approach, even when the latter is improved by using the Woodbury formula and the matrix determinant lemma.

In early stage of learning, the method-of-moments estimators (MMEs) could be used for training the variance components, because MLEs are often sensitive to the bias in the mean parameters and MMEs take less computational cost. It is worth noting that the MMEs require the random effect predictors $\widehat{\mathbf{v}}$, which are not provided by the integrated likelihood while training the network. When the number of dispersion parameters is small, second order optimization algorithms can be used for the covariance kernel, such as the RBF kernel. Newton-Raphson method is implemented for estimation of dispersion parameters.

4.1. REML procedure

The restricted h-likelihood of the proposed model (6) can be obtained by replacing $\mathbf{X}$ and $\mathbf{Z}$ in (5) with $\widehat{f}(\mathbf{X})$ and $\widehat{g}(\mathbf{Z})$. For given $\widehat{f}$ and $\widehat{g}$, the restricted h-likelihood is given by

$$h_R(\boldsymbol{\psi}) = h(\boldsymbol{\psi}; \widehat{f}, \widehat{g}, \widehat{\boldsymbol{\beta}}, \widehat{\mathbf{v}}) - \frac{1}{2}\log\left|\frac{1}{\sigma_e^2}\widehat{f}(\mathbf{X})^T\widehat{f}(\mathbf{X}) - \frac{1}{\sigma_e^4}\widehat{f}(\mathbf{X})^T\widehat{g}(\mathbf{Z})\mathbf{A}^{-1}\widehat{g}(\mathbf{Z})^T\widehat{f}(\mathbf{X})\right|,$$

where $\mathbf{A} = \frac{1}{\sigma_e^2}\widehat{g}(\mathbf{Z})^T\widehat{g}(\mathbf{Z}) + \mathbf{D}^{-1}$, which allows REML procedure for LMMNN models.

4.2. Adjustments for Random Effects

In LMMs, constraints are imposed on the random effects $\mathrm{E}(\mathbf{v}) = \mathbf{0}$. Without the constraints, the proposed model (6) has additional parameter $\mu_k = \mathrm{E}(\mathbf{v}_k)$ and the transformation

$$\begin{aligned} \beta_0^* &= \beta + \epsilon_k \\ \mathbf{v}_k^* &= \mathbf{v}_k - \epsilon_k \sim N\left(\mu_k^* = \mu_k - \epsilon_k, \mathbf{D}_k\right) \end{aligned}$$

gives the same h-likelihood, so that the parameters may not be identifiable. Thus, when the DNN models contains the random effects, the bias in local minima can cause poor predictions. Simchoni & Rosset (2022) considered two cases of $g_k(\cdot)$, the identity function $g_k(\mathbf{Z}_k) = \mathbf{Z}_k$ and $g_k(\mathbf{Z}_k) = \mathbf{Z}_k\mathbf{W}_k$ where $\mathbf{W}_k$ is $q_k^* \times q_k$ matrix with $q_k \leq q_k^*$. When $g_k(\cdot)$ is identity function, we propose the following

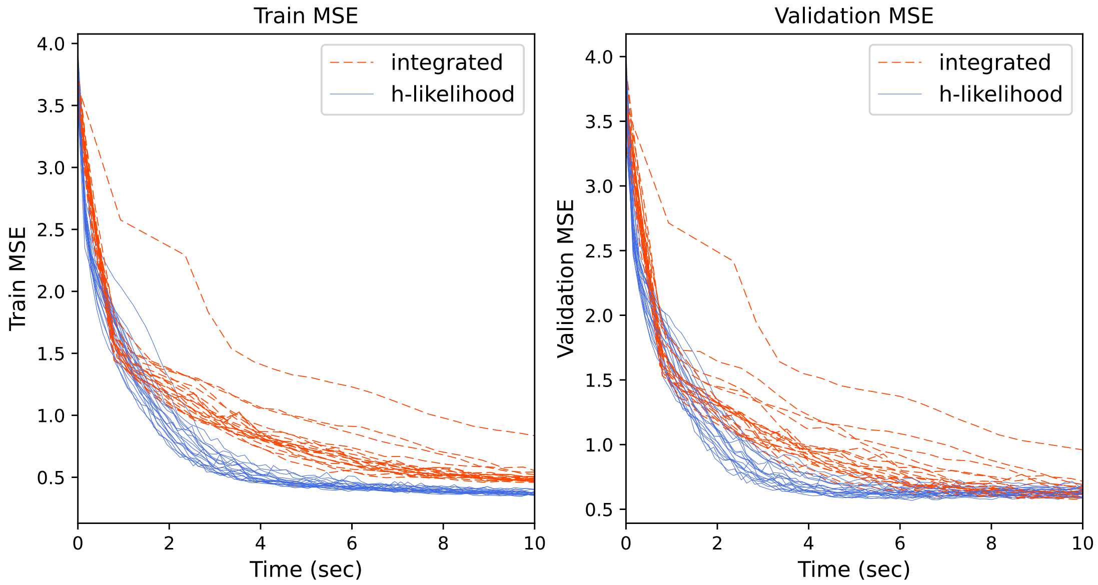

Figure 2. MSE curves of the integrated likelihood approach and the proposed h-likelihood approach from 20 repetitions. $N = 10,000$ data are generated from the normal distribution with a nonlinear function $f(\mathbf{x}) = (x_1 + x_2)\cos(x_1 + x_2) + 2x_1x_2$, $q = 100$ dimensional Gaussian random effects $\mathbf{v}_1$ and $\mathbf{v}_2$ from $N(0, I_{100})$, and $\sigma_e^2 = 1$.

adjustment for local minima by putting constraints on the random effect predictors,

$$\widehat{\mathbf{v}}_k^* = \widehat{\mathbf{v}}_k - \frac{\widehat{\mathbf{v}}_k^T \widehat{\mathbf{D}}^{-1}\mathbf{1}_{q_k}}{\mathbf{1}_{q_k}^T \widehat{\mathbf{D}}_k^{-1}\mathbf{1}_{q_k}},$$
$$\widehat{\beta}_0^* = \widehat{\beta}_0 + \frac{\widehat{\mathbf{v}}_k^T \widehat{\mathbf{D}}^{-1}\mathbf{1}_{q_k}}{\mathbf{1}_{q_k}^T \widehat{\mathbf{D}}_k^{-1}\mathbf{1}_{q_k}}, \tag{8}$$

where $\mathbf{1}_{q_k} = (1, ..., 1)^T$. When the random effect $\mathbf{v}_k$ is independent, i.e., $\mathbf{D}_k = \lambda_k \mathbf{I}_{q_k}$, the adjustment becomes

$$\widehat{\mathbf{v}}_k^* = \widehat{\mathbf{v}}_k - \frac{1}{q_k}\sum_{j=1}^{q_k} \widehat{v}_{kj} \quad \text{and} \quad \widehat{\beta}_0 = \widehat{\beta} + \frac{1}{q_k}\sum_{j=1}^{q_k} \widehat{v}_{kj}.$$

Following theorem shows that the proposed adjustment (8) can always reduce the proposed loss function.

Theorem 4.1. *In the LMMNN* (6)*, suppose that* $(\widehat{\boldsymbol{\theta}}^*, \widehat{\mathbf{v}}^*)$ *is the replacement of* $\widehat{\beta}_0$ *and* $\widehat{\mathbf{v}}_k$ *in* $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{v}})$ *with the adjusted values* $\widehat{\boldsymbol{\theta}}^*$ *and* $\widehat{\mathbf{v}}_k^*$ *in* (8)*, then*

$$h(\widehat{\boldsymbol{\theta}}^*, \widehat{\mathbf{v}}^*) \geq h(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{v}})$$

and the equality holds if and only if $\widehat{\mathbf{v}}_k^T \widehat{\mathbf{D}}^{-1}\mathbf{1}_{q_k} = 0$*, i.e.,* $\widehat{\mathbf{v}}_k^* = \widehat{\mathbf{v}}_k$.

Proof is in Appendix A5. The proposed algorithm is described in Algorithm 1.

Algorithm 1 Two-step Algorithm for H-likelihood

```
Input: $\mathbf{x}_i$, $\mathbf{z}_i$
Initialize all the fixed and random parameters.
repeat
  < M-step >
  for epoch $= 1$ to m do
    Update the mean parameters in $f$, $g$, $\boldsymbol{\beta}$ and $\mathbf{v}$ for every mini-batch.
  end for
  < V-step >
  Update dispersion parameters in $\boldsymbol{\psi}$ by using the whole training data (full batch).
until the loss function is not improved for pre-determined number of times
Adjust the random effect predictors $\widehat{\mathbf{v}}$ as in (8).
```

5. Comparison with existing methods

To show the performance of their integrated likelihood approaches, namely the LMMNN with and without assuming spatial correlation (LMMNN-R and LMMNN-E), Simchoni & Rosset (2022) reported the results from various existing methods, one-hot encoding (OHE), entity embedding (EMB; Guo & Berkhahn (2016)), convolutional neural network (CNN; LeCun et al. (1998)) and stochastic variational deep kernel learning (SV-DKL; Wilson et al. (2016b)). To

Table 1. Mean and standard error of test MSPEs. Results of existing models are cited from Simchoni & Rosset (2022).

l^2	OHE	EMB	CNN	SV-DKL	LMMNN-E	LMMNN-R	**HL (PROPOSED)**
0.1	1.35 (.01)	1.34 (.01)	1.28 (.02)	1.26 (.03)	1.26 (.02)	1.29 (.07)	**1.11 (.03)**
1.0	1.33 (.01)	1.34 (.02)	1.27 (.02)	1.12 (.01)	1.18 (.02)	1.13 (.02)	**1.03 (.03)**
10.0	1.34 (.01)	1.30 (.02)	1.22 (.02)	1.09 (.03)	1.10 (.01)	1.10 (.01)	**1.10 (.05)**

Table 2. Estimated variance components on average when $\sigma_e^2 = \sigma_v^2 = 1$. Results of LMMNN-R are cited from Simchoni & Rosset (2022).

TRUE	LMMNN-R			HL (MLE)			HL (REMLE)		
l^2	$\widehat{\sigma}_e^2$	$\widehat{\sigma}_v^2$	$\widehat{l}^2$	$\widehat{\sigma}_e^2$	$\widehat{\sigma}_v^2$	$\widehat{l}^2$	$\widehat{\sigma}_e^2$	$\widehat{\sigma}_v^2$	$\widehat{l}^2$
0.1	1.12	0.99	0.48	0.9337	0.9832	0.0967	0.9337	0.9830	0.0971
1.0	1.12	1.10	1.49	1.0013	1.0594	1.0085	1.0013	1.0588	1.0083
10.0	1.11	0.74	4.93	0.9623	0.7124	8.9290	0.9623	0.7131	8.9338

study the performance of the proposed model, we first review the existing methods for comparison.

- OHE is a basic approach to handle the categorical features, but it becomes challenging when the number of categories is large.
- EMB is known to improve OHE by mapping the high-cardinality categorical features into the low-dimensional Euclidean spaces.
- CNN is the most widely used method to analyze visual images. For spatial data, CNN can be applied by handling the locations as images.
- SV-DKL is a stochastic variational procedure which generalize the deep kernel learning (Wilson et al., 2016a). It is considered as a SOTA method for handling spatial data. Deep kernel learning combines the non-parametric flexibility of kernel methods with the inductive biases of deep learning architectures. Wilson et al. (2016b) showed that SV-DKL can take advantages over alternative scalable Gaussian process models and stand-alone DNNs.
- LMMNN-E transforms the locations into a 1000 dimensional vector which is treated as a single independent random effects.
- LMMNN-R uses the RBF covariance kernel for the spatial random effects. It has the similar model formulation with our proposed HL methods but different loss function and learning algorithm using the block-diagonal approximation.

The h-likelihood approach gives exact MLEs, whereas SOTA methods such as SV-DKL, LMMNN-E and LMMNN-R provide only approximate MLEs.

6. Numerical Studies

We present numerical studies using spatial data to demonstrate the performance of the proposed method. Following Simchoni & Rosset (2022), we generate the data as follows. For $i = 1, ..., N$, input variable $\mathbf{x}_i = (x_{i1}, ..., x_{i10})^T$ are sampled from $U(-1, 1)$ distribution and

$$y_i = x_{i+} \cdot \cos x_{i+} + 2x_{i1}x_{i2} + \mathbf{z}_i^T \mathbf{v} + \epsilon_i$$

where $x_{i+} = x_{i1} + \cdots + x_{i10}$, the noise ϵ_i is sampled from $N(0, \sigma_e^2)$, and a vector of random effects $\mathbf{v}$ is sampled from the multivariate normal distribution with zero mean and covariance represented by RBF kernel, for $i, j \in \{1, ..., q\}$,

$$\text{Cov}(v_i, v_j) = \sigma_v^2 \exp\left\{-\frac{(s_i - s_j)^2}{2l^2}\right\},$$

where s_i and s_j are 2-dimensional locations sampled from $U(-10, 10) \times U(-10, 10)$ grid.

We generate $N = 100,000$ data points with $q = 1,000$ random effects. We randomly separate the data into training set (60%), validation set (20%) and test set (20%). All experiments are repeated 100 times. To fit the proposed method, Adam optimizer is used for the mean parameters, and Newton-Raphson methods is used for the variance components. Since the MLEs for variance components could be sensitive to the bias in the mean parameters, method-of-moments estimators in Appendix A4 are used in early stages. Standard multi-layer perceptrons (MLPs) with 4 hidden layers of 100-50-25-12 neurons and 25% dropout were applied for all the experiments. Sigmoid activation function is used for the last hidden layer to obtain the REMLEs, and ReLU activation function is used for the others. Early stopping criteria with validation loss is employed to prevent overfitting. The proposed method is implemented using Python based on Keras (Chollet et al., 2015) and Tensorflow (Abadi

Table 3. Test MSPEs for Asthma data set. Results of existing models are cited from Simchoni & Rosset (2022).

DATA	IGNORE	EMB	CNN	SV-DKL	LMMNN-E	LMMNN-R	**HL (PROPOSED)**
INCOME	.034 (.00)	.032 (.00)	.032 (.00)	.030 (.00)	.027 (.00)	.028 (.00)	0.028 (.00)
AIR QUALITY	.285 (.02)	.260 (.04)	.163 (.06)	.044 (.01)	.088 (.02)	.035 (.00)	0.023 (.00)
CARS	.152 (.00)	.092 (.00)	.137 (.00)	.149 (.00)	.136 (.00)	.084 (.00)	0.084 (.00)

et al., 2015), and all the experiments are made on Nvidia RTX 2080Ti GPU.

We report the mean and standard error of mean squared prediction errors (MSPEs) of test data,

$$\text{MSPE} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (y_i - \widehat{y}_i)^2.$$

Since the prediction is insensitive to the estimation of dispersion parameters, the MLE and REMLE have the same prediction MSPE. Difference between LMMNN methods and HL method is dispersion parameter estimation of the HL method. Thus, the exact ML estimation of dispersion parameters enhance the predictability. Table 1 shows that the proposed method is better than all the existing methods. Table 2 shows the estimation of variance components. For the length-scale parameter l^2 in RBF kernel, block-diagonal approximation of LMMNN method produces severely biased estimates, whereas the proposed method estimates accurately. For both σ_e^2 and σ_v^2, the proposed exact MLEs are slightly better than the approximate MLEs using block-diagonal approximation. Compared to MLE, REMLE is slightly less biased, but the difference is small despite the additional computing cost. In LMMs of finite samples, REMLEs often reduces the bias of the MLEs, but in LMMNN with large N, the improvement seems negligible.

To demonstrate the usefulness of the adjustment (8) of random effects predictor, we report the root mean squared errors (RMSEs) of random effects predictors. Without adjustment, mean and standard error of RMSEs are 0.14 (0.06). With adjustment, mean and standard error of RMSEs are 0.13 (0.05). We have focused that the adjustment improves not only the random effect prediction but also estimation of MLEs for variance components, which gives the good prediction performance of the HL procedure.

In summary, the proposed HL method outperforms the existing methods including the SOTA methods of variational approach and integrated likelihood approach for the spatial data, including SV-DKL (Wilson et al., 2016b), LMMNN-E and LMMNN-R (Simchoni & Rosset, 2022).

7. Real Data Analysis

Simchoni & Rosset (2022) analyzed several data sets. They used the 5-fold cross validation (CV) procedures where 80% of the data is used to predict and the remaining 20% is test data. Standard MLPs with two hidden layers of 10-3 neurons and ReLU activation function were used for all the data sets. RBF kernel was used for spatial correlation. Instead of OHE, analysis ignoring correlation structure (Ignore) was shown, since OHE perform similarly to EMB in simulation studies.

7.1. Income data

Income data (MuonNeutrino, 2019) have mean yearly income in dollars for $71,371$ US census tracts from $3,108$ counties. The response variable is log-income and in addition to the location features (longitude and latitude), the data contain $p = 30$ input variables. Here, $N = 71,371$, $K = 1$, and $q = 3,108$.

7.2. Air quality data

Centers for Disease Control and Prevention reported air quality data (CDC, 2020) of PM2.5 particles level in $71,347$ US census tracts. Simchoni & Rosset (2022) analyzed the air quality data by using additional features from the income data. The response variable is PM2.5 particles level with $p = 32$ input variables. Here, $N = 71,347$, $K = 1$, and $q = 3,107$.

7.3. Cars data

Cars data (Reese, 2020) have the price of $N = 97,729$ used cars. The response variable is log-price of the cars. It contains $q_1 = 15,226$ models, $q_2 = 12,235$ locations to give $Q = q_1 + q_2 = 27,461$, and $p = 73$ input variables. Since $\mathbf{D}_1 = \lambda_1 \mathbf{I}_{q_1}$, we only need to compute the $q_2 \times q_2$ inverse matrix, instead of either $N \times N$ or $Q \times Q$ inverse matrices.

7.4. Prediction results

Table 3 shows the mean of the MSPEs for test data from 5-fold CV procedure. In air quality data, the proposed method has the smallest MSPEs. In income data, it has comparable MSPEs to the smallest MSPE of LMMNN-E without spatial random effects. In cars data, the proposed method and LMMNN-R outperform the other methods. Figure 3 shows the predicted values of output variables against the true values for the income data and air quality data.

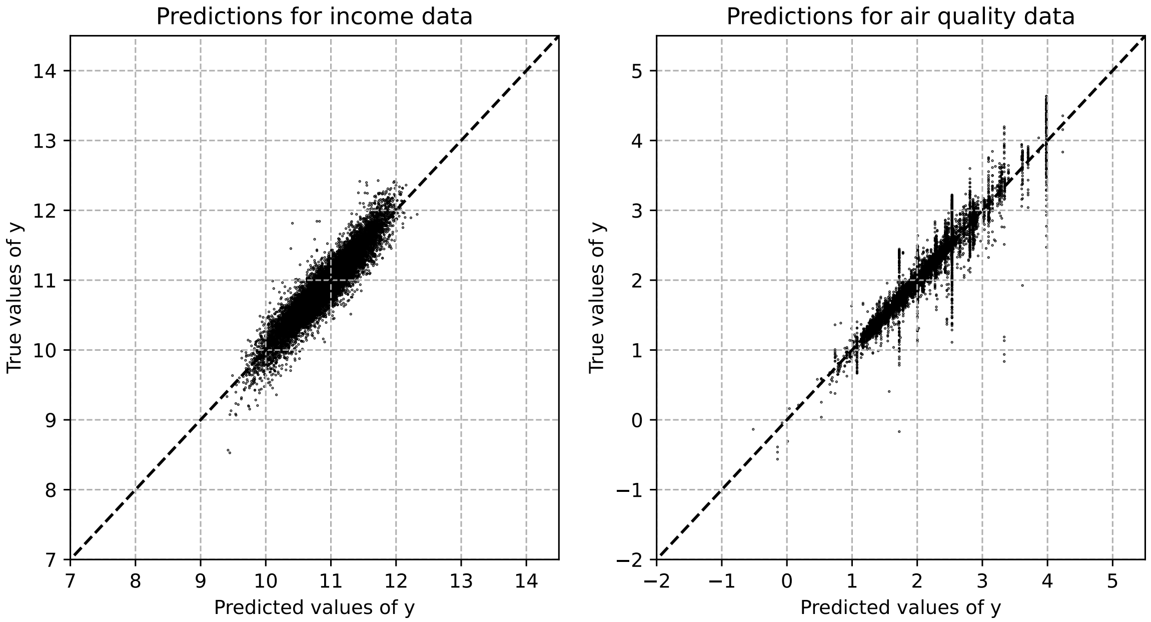

Figure 3. The HL predictors from income data (left) and air quality data (right).

When Simchoni & Rosset (2022)'s block-diagonal approximation works well (income data and cars data), the proposed method and LMMNN-R behave similarly, whereas the approximation does not work well (air quality data), the proposed method outperforms LMMNN-R. However, not only the correlation matrix, but also the data, parameters, and the batch size can affect the accuracy of the approximation. Thus, it is hard to know whether the approximation will work well or not.

8. Concluding Remarks

In LMMs, the conventional integrated likelihood has been successfully implemented to obtain the MLEs. However, with the surge of DNN models, the integrated likelihood encounters a computational difficulty due to the large size of data. Variational methods and approximate integrated likelihood approach have been proposed to obtain approximate MLEs. However, they could have non-negligible biases, so the algorithm to obtain the exact MLEs is of interest. Lee & Nelder (1996) proposed the h-likelihood to avoid numerically difficult integration. However, it does not give the exact MLEs for variance components. In this paper, we introduce a new h-likelihood for LMMs, which gives the MLEs for whole parameters and BLUPs for random effects.

For LMMNN models, the two-step algorithm enables online learning by minimizing the negative h-likelihood loss function. Its joint optimization produces exact MLEs for mean and dispersion parameters and BLUP for the random effects. The algorithm also avoids a difficulty to implement the REMLE procedure for variance components. In LMMNN models, we found that an adjustment for random effect predictors is useful for enhancing the performance of variance component estimation. In this paper, we only considered simple MLP for the neural network $f(\mathbf{x})$, but more complex architectures can be easily implemented.

Via simulations and real data analyses, we show that predictive performance of HL method outperforms the existing methods, OHE, EMBED, CNN, and SOTA methods, SV-DKL, LMMMNN-E and LMMNN-R.

In the future we hope to make the proposed method more computationally efficient, applicable to non-normal hierarchical models such as hierarchical generalized linear models (Lee & Nelder, 1996) with neural networks.

Software and Data

Source codes for numerical studies and real data analyses in this paper are available on Github: `https://github.com/hangbin221/deepHGLM`.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2019R1A2C1002408).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/.

CDC. Daily census tract-level pm2.5 concentrations, 2020. URL https://data.cdc.gov/Environmental-Health-Toxicology/Daily-Census-Tract-Level-PM2-5-Concentrations-2016/7vu4-ngxx.

Chollet, F. et al. Keras, 2015. URL https://keras.io.

Cox, D. R. and Reid, N. Parameter orthogonality and approximate conditional inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 49(1):1–18, 1987.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Guo, C. and Berkhahn, F. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.

Henderson, C. R., Kempthorne, O., Searle, S. R., and Von Krosigk, C. The estimation of environmental and genetic trends from records subject to culling. *Biometrics*, 15(2):192–218, 1959.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Lee, Y. and Nelder, J. A. Hierarchical generalized linear models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(4):619–656, 1996.

Lee, Y., Nelder, J. A., and Pawitan, Y. *Generalized linear models with random effects: unified analysis via H-likelihood*. Chapman and Hall/CRC, 2017.

Mandel, F., Ghosh, R. P., and Barnett, I. Neural networks for clustered and longitudinal data using mixed effects models. *Biometrics*, 2021.

MuonNeutrino. Us census demographic data, 2019. URL https://www.kaggle.com/datasets/muonneutrino/us-census-demographic-data.

Patterson, H. D. and Thompson, R. Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58(3):545–554, 1971.

Reese, A. Used cars dataset - vehicles listings from craigslist.org, 2020. URL https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data.

Simchoni, G. and Rosset, S. Using random effects to account for high-cardinality categorical features and repeated measures in deep neural networks. *Advances in Neural Information Processing Systems*, 34:25111–25122, 2021.

Simchoni, G. and Rosset, S. Integrating random effects in deep neural networks. *arXiv preprint arXiv:2206.03314*, 2022.

Tran, M.-N., Nguyen, N., Nott, D., and Kohn, R. Bayesian deep net GLM and GLMM. *Journal of Computational and Graphical Statistics*, 29(1):97–113, 2020.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In *Artificial intelligence and statistics*, pp. 370–378. PMLR, 2016a.

Wilson, A. G., Hu, Z., Salakhutdinov, R. R., and Xing, E. P. Stochastic variational deep kernel learning. *Advances in Neural Information Processing Systems*, 29, 2016b.

A. Appendix

A.1. Derivation of the new h-likelihood

Since $\mathbf{v}|\mathbf{y}$ has the multivariate normal distribution,

$$\mathbf{v}|\mathbf{y} \sim N\left(\frac{1}{\sigma_e^2}\mathbf{A}^{-1}\mathbf{Z}^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}),\ \mathbf{A}^{-1}\right),$$

where $\mathbf{A} = \sigma_e^{-2}\mathbf{Z}^T\mathbf{Z} + \mathbf{D}^{-1}$, the distribution of $\mathbf{v}^c|\mathbf{y}$ is given by

$$\mathbf{v}^c|\mathbf{y} \sim N\left(\frac{1}{\sigma_e^2}\mathbf{A}^{-\frac{1}{2}}\mathbf{Z}^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}),\ \mathbf{I}_Q\right),$$

which leads to $\widetilde{\mathbf{v}}^c = \sigma_e^{-2}\mathbf{A}^{-\frac{1}{2}}\mathbf{Z}^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})$ and the predictive likelihood

$$\log f_{\boldsymbol{\theta}}(\widetilde{\mathbf{v}}^c|\mathbf{y}) = -\frac{1}{2}\log|2\pi\mathbf{I}_Q| = \text{constant}.$$

Thus, $\mathbf{v}^c = \mathbf{A}^{\frac{1}{2}}\mathbf{v}$ is the canonical scale to give the h-likelihood,

$$\begin{aligned} h(\boldsymbol{\theta},\mathbf{v}^c) = &-\frac{1}{2\sigma_e^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\mathbf{A}^{-\frac{1}{2}}\mathbf{v}^c\right)^T\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\mathbf{A}^{-\frac{1}{2}}\mathbf{v}^c\right) - \frac{N}{2}\log(2\pi\sigma_e^2) \\ &-\frac{1}{2}\mathbf{v}^{cT}\mathbf{A}^{-\frac{1}{2}}\mathbf{D}^{-1}\mathbf{A}^{-\frac{1}{2}}\mathbf{v}^c - \frac{1}{2}\log\left|2\pi\mathbf{A}^{\frac{1}{2}}\mathbf{D}\mathbf{A}^{\frac{1}{2}}\right|. \end{aligned}$$

of which the joint maximization gives the MLEs for the whole parameters.

The first derivatives of the h-likelihood with respect to $\boldsymbol{\beta}$ and $\mathbf{v}^c$ are

$$\begin{aligned} \frac{\partial h(\boldsymbol{\theta},\mathbf{v}^c)}{\partial\boldsymbol{\beta}} &= \frac{1}{\sigma_e^2}\mathbf{X}^T\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\mathbf{A}^{-\frac{1}{2}}\mathbf{v}^c\right), \\ \frac{\partial h(\boldsymbol{\theta},\mathbf{v}^c)}{\partial\mathbf{v}^c} &= \frac{1}{\sigma_e^2}\mathbf{A}^{-\frac{1}{2}}\mathbf{Z}^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) - \mathbf{v}^c, \end{aligned}$$

and the second derivatives are

$$\frac{\partial^2 h(\boldsymbol{\theta},\mathbf{v}^c)}{\partial\boldsymbol{\beta}^2} = -\frac{1}{\sigma_e^2}\mathbf{X}^T\mathbf{X}, \qquad \frac{\partial^2 h(\boldsymbol{\theta},\mathbf{v}^c)}{\partial\boldsymbol{\beta}\partial\mathbf{v}^c} = -\frac{1}{\sigma_e^2}\mathbf{A}^{-\frac{1}{2}}\mathbf{Z}^T\mathbf{X}, \qquad \frac{\partial^2 h(\boldsymbol{\theta},\mathbf{v}^c)}{\partial\mathbf{v}^{c2}} = -\mathbf{I}_Q,$$

which leads to

$$\left|-\frac{\partial^2 h(\boldsymbol{\theta},\mathbf{v}^c)}{\partial(\boldsymbol{\beta},\mathbf{v}^c)^2}\right| = \left|\begin{pmatrix} \mathbf{I}_Q & \frac{1}{\sigma_e^2}\mathbf{A}^{-\frac{1}{2}}\mathbf{Z}^T\mathbf{X} \\ \frac{1}{\sigma_e^2}\mathbf{X}^T\mathbf{Z}\mathbf{A}^{-\frac{1}{2}} & \frac{1}{\sigma_e^2}\mathbf{X}^T\mathbf{X} \end{pmatrix}\right| = \left|\frac{1}{\sigma_e^2}\mathbf{X}^T\mathbf{X} - \frac{1}{\sigma_e^4}\mathbf{X}^T\mathbf{Z}\mathbf{A}^{-1}\mathbf{Z}^T\mathbf{X}\right|.$$

Thus, the adjusted profile h-likelihood is given by

$$h_R(\boldsymbol{\psi}) = h(\boldsymbol{\psi};\widehat{\boldsymbol{\beta}},\widehat{\mathbf{v}}^c) - \frac{1}{2}\log\left|\frac{1}{\sigma_e^2}\mathbf{X}^T\mathbf{X} - \frac{1}{\sigma_e^4}\mathbf{X}^T\mathbf{Z}\mathbf{A}^{-1}\mathbf{Z}^T\mathbf{X}\right|,$$

which is the integrated likelihood,

$$\begin{aligned} h_R(\boldsymbol{\psi}) &= \log\iint\exp(h(\boldsymbol{\theta},\mathbf{v}^c))d\mathbf{v}^c d\boldsymbol{\beta} = \log\iint f_{\boldsymbol{\theta}}(\mathbf{y},\mathbf{v}^c)d\mathbf{v}^c d\boldsymbol{\beta} = \log\int f_{\boldsymbol{\theta}}(\mathbf{y})d\boldsymbol{\beta} \\ &= \log\int\exp(\ell(\boldsymbol{\theta}))d\boldsymbol{\beta} = \ell(\boldsymbol{\psi};\widehat{\boldsymbol{\beta}}) - \frac{1}{2}\log|\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X}| = \ell_R(\boldsymbol{\psi}). \end{aligned}$$

Thus, the restricted likelihood is an adjusted profile h-likelihood.

A.2. The computation of h-likelihood when q_1 is large

Suppose that the model contains a large q_1 dimensional independent random effects with $\mathbf{D}_1 = \lambda_1 \mathbf{I}_{q_1}$ Since $\mathbf{Z}_1^T\mathbf{Z}_1$ is diagonal, the determinant $\left|\frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z}\mathbf{D}+\mathbf{I}_Q\right|$ in the h-likelihood (4) can be expressed as

$$\begin{aligned}\log\left|\frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z}\mathbf{D}+\mathbf{I}_Q\right| &= \log\begin{vmatrix}\frac{1}{\sigma_e^2}\mathbf{Z}_1^T\mathbf{Z}_1\mathbf{D}_1+\mathbf{I}_{q_1} & \frac{1}{\sigma_e^2}\mathbf{Z}_1^T\mathbf{Z}_2\mathbf{D}_2 & \cdots & \frac{1}{\sigma_e^2}\mathbf{Z}_1^T\mathbf{Z}_K\mathbf{D}_K \\ \frac{1}{\sigma_e^2}\mathbf{Z}_2^T\mathbf{Z}_1\mathbf{D}_1 & \frac{1}{\sigma_e^2}\mathbf{Z}_2^T\mathbf{Z}_2\mathbf{D}_2+\mathbf{I}_{q_2} & \cdots & \frac{1}{\sigma_e^2}\mathbf{Z}_2^T\mathbf{Z}_K\mathbf{D}_K \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\sigma_e^2}\mathbf{Z}_K^T\mathbf{Z}_1\mathbf{D}_1 & \frac{1}{\sigma_e^2}\mathbf{Z}_K^T\mathbf{Z}_2\mathbf{D}_2 & \cdots & \frac{1}{\sigma_e^2}\mathbf{Z}_K^T\mathbf{Z}_K\mathbf{D}_K+\mathbf{I}_{q_K}\end{vmatrix} \\ &= \log|\mathbf{B}_{11}| + \log\left|\mathbf{I}_{Q-q_1}+\frac{1}{\sigma_e^2}\mathbf{Z}_{-1}^T\mathbf{Z}_{-1}\mathbf{D}_{-1}-\frac{1}{\sigma_e^4}\mathbf{Z}_{-1}^T\mathbf{Z}_1\mathbf{D}_1\mathbf{B}_{11}^{-1}\mathbf{Z}_1^T\mathbf{Z}_{-1}\mathbf{D}_{-1}\right|\end{aligned}$$

where $\mathbf{Z}_{-1} = (\mathbf{Z}_2, ..., \mathbf{Z}_K)$, $\mathbf{D}_{-1} = \text{block-diag}(\mathbf{D}_2, ..., \mathbf{D}_K)$,

$$\mathbf{B}_{11} = \frac{1}{\sigma_e^2}\mathbf{Z}_1^T\mathbf{Z}_1\mathbf{D}_1+\mathbf{I}_{q_1} = \text{diag}\left(\frac{\lambda_1}{\sigma_e^2}n_{1j}+1\right)_{j=1,\dots,q_1},$$

and $n_{1j} = \sum_{t=1}^N z_{1jt}$ is the number of observations in the j-th category of the first categorical variable $\mathbf{Z}_1$. Since the first term is the determinant of diagonal matrix $\mathbf{B}_{11}$ and the second term is the determinant of the size $\sum_{k=2}^K q_K \ll q_1$ matrix, the h-likelihood can be easily computed without handling the inverse computation of $Q \times Q$ matrices. In Appendix A3 below, the first and the second derivatives of the h-likelihood with respect to the variance components are derived and they can be obtained without computing the inverse of full $Q \times Q$ matrix.

A.3. First and second derivatives with respect to variance components

Let $s_e = \log\sigma_e^2$ be the log-variance of random noise and $\boldsymbol{\lambda}_k = (\lambda_{k1}, ..., \lambda_{kj_k})$ be the vector of j_k dispersion parameters involved in $\mathbf{D}_k$ for $k = 1, ..., K$, then the objective function can be expressed as

$$\begin{aligned}\text{Loss} &= e^{-s_e}\left(\mathbf{y}-\widehat{\mathbf{y}}\right)^T\left(\mathbf{y}-\widehat{\mathbf{y}}\right)+Ns_e+\sum_{k=1}^K \mathbf{v}_k^T\mathbf{D}_k^{-1}\mathbf{v}_k+\log|\mathbf{B}| \\ &= a_0(s_e)+\sum_{k=1}^K a_k(\boldsymbol{\lambda}_k)+\log|\mathbf{B}(s_e,\boldsymbol{\lambda}_1,...,\boldsymbol{\lambda}_K)|\end{aligned}$$

where $\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}}+\mathbf{Z}\widehat{\mathbf{v}}$, $\mathbf{B} = \mathbf{A}\mathbf{D} = e^{-s_e}\mathbf{Z}^T\mathbf{Z}\mathbf{D}+\mathbf{I}_Q$, $a_0(s_e) = e^{-s_e}\left(\mathbf{y}-\widehat{\mathbf{y}}\right)^T\left(\mathbf{y}-\widehat{\mathbf{y}}\right)+Ns_e$ and $a_k(\boldsymbol{\lambda}_k) = \mathbf{v}_k^T\mathbf{D}_k^{-1}\mathbf{v}_k$. Here the derivatives of $\log|\mathbf{B}|$ is difficult to evaluate. The first deirvatives of $\mathbf{B}$ are given by

$$\begin{aligned}\frac{\partial\mathbf{B}}{\partial s_e} &= -e^{-s_e}\mathbf{Z}^T\mathbf{Z}\mathbf{D} = \mathbf{I}_Q-\mathbf{B}, \\ \frac{\partial\mathbf{B}}{\partial\lambda_{kj}} &= e^{-s_e}\mathbf{Z}^T\mathbf{Z}\frac{\partial\mathbf{D}}{\partial\lambda_{kj}} = e^{-s_e}\left(\mathbf{0}_{k-},\ \mathbf{Z}^T\mathbf{Z}_k\frac{\partial\mathbf{D}_k}{\partial\lambda_{kj}},\ \mathbf{0}_{k+}\right),\end{aligned}$$

where $\mathbf{0}_{k-}$ and $\mathbf{0}_{k+}$ are zero matrices of size $Q \times (q_1+\cdots+q_{k-1})$ and $Q \times (q_{k+1}+\cdots+q_K)$, respectively, so that

$$\mathbf{Z}^T\mathbf{Z}\frac{\partial\mathbf{D}}{\partial\lambda_{kj}} = \left(\mathbf{0}_{k-},\ \mathbf{Z}^T\mathbf{Z}_k\frac{\partial\mathbf{D}_k}{\partial\lambda_{kj}},\ \mathbf{0}_{k+}\right) = \begin{pmatrix}\mathbf{0} & \cdots & \mathbf{0} & \mathbf{Z}_1^T\mathbf{Z}_k\frac{\partial\mathbf{D}_k}{\partial\lambda_{kj}} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{Z}_K^T\mathbf{Z}_k\frac{\partial\mathbf{D}_k}{\partial\lambda_{kj}} & \mathbf{0} & \cdots & \mathbf{0}\end{pmatrix},$$

and the non-zero second derivatives are given by

$$\begin{aligned}
\frac{\partial^2\mathbf{B}}{\partial s_e^2} &= e^{-s_e}\mathbf{Z}^T\mathbf{Z}\mathbf{D} = \mathbf{B} - \mathbf{I}_Q,\\
\frac{\partial^2\mathbf{B}}{\partial s_e\partial\lambda_{kj}} &= -e^{-s_e}\mathbf{Z}^T\mathbf{Z}\frac{\partial\mathbf{D}}{\partial\lambda_{kj}} = -\frac{\partial\mathbf{B}}{\partial\lambda_{kj}},\\
\frac{\partial^2\mathbf{B}}{\partial\lambda_{kj}^2} &= e^{-s_e}\mathbf{Z}^T\mathbf{Z}\frac{\partial^2\mathbf{D}}{\partial\lambda_{kj}^2} = e^{-s_e}\left(\mathbf{0}_{k-},\ \mathbf{Z}^T\mathbf{Z}_k\frac{\partial^2\mathbf{D}_k}{\partial\lambda_{kj}^2},\ \mathbf{0}_{k+}\right),\\
\frac{\partial^2\mathbf{B}}{\partial\lambda_{ki}\partial\lambda_{kj}} &= e^{-s_e}\mathbf{Z}^T\mathbf{Z}\frac{\partial\mathbf{D}}{\partial\lambda_{kj}} = e^{-s_e}\left(\mathbf{0}_{k-},\ \mathbf{Z}^T\mathbf{Z}_k\frac{\partial^2\mathbf{D}_k}{\partial\lambda_{ki}\partial\lambda_{kj}},\ \mathbf{0}_{k+}\right).
\end{aligned}$$

Thus, the first derivatives of $\log|\mathbf{B}|$ are

$$\begin{aligned}
\frac{\partial\log|\mathbf{B}|}{\partial s_e} &= tr\left[\mathbf{B}^{-1}\frac{\partial\mathbf{B}}{\partial s_e}\right] = tr[\mathbf{B}^{-1} - \mathbf{I}_Q] = tr[\mathbf{B}^{-1}] - Q\\
\frac{\partial\log|\mathbf{B}|}{\partial\lambda_{kj}} &= tr\left[\mathbf{B}^{-1}\frac{\partial\mathbf{B}}{\partial\lambda_{kj}}\right] = e^{-s_e}tr\left[[\mathbf{B}^{-1}]_k\mathbf{Z}^T\mathbf{Z}_k\frac{\partial\mathbf{D}_k}{\partial\lambda_{kj}}\right]
\end{aligned}$$

and the second derivatives of $\log|\mathbf{B}|$ are

$$\begin{aligned}
\frac{\partial^2\log|\mathbf{B}|}{\partial s_e^2} &= tr\left[\mathbf{B}^{-1}\frac{\partial^2\mathbf{B}}{\partial s_e^2}\right] - tr\left[\left(\mathbf{B}^{-1}\frac{\partial\mathbf{B}}{\partial s_e}\right)^2\right] = tr[\mathbf{B}^{-1} - \mathbf{B}^{-2}]\\
\frac{\partial^2\log|\mathbf{B}|}{\partial\lambda_{kj}^2} &= tr\left[\mathbf{B}^{-1}\frac{\partial^2\mathbf{B}}{\partial\lambda_{kj}^2}\right] - tr\left[\left(\mathbf{B}^{-1}\frac{\partial\mathbf{B}}{\partial\lambda_{kj}}\right)^2\right]\\
&= e^{-s_e}tr\left[[\mathbf{B}^{-1}]_k\mathbf{Z}^T\mathbf{Z}_k\frac{\partial^2\mathbf{D}_k}{\partial\lambda_{kj}^2}\right] - e^{-s_e}tr\left[\left([\mathbf{B}^{-1}]_k\mathbf{Z}^T\mathbf{Z}_k\frac{\partial\mathbf{D}_k}{\partial\lambda_{kj}}\right)^2\right]\\
\frac{\partial^2\log|\mathbf{B}|}{\partial s_e\partial\lambda_{kj}} &= tr\left[\mathbf{B}^{-1}\frac{\partial^2\mathbf{B}}{\partial s_e\partial\lambda_{kj}}\right] - tr\left[\mathbf{B}^{-1}\frac{\partial\mathbf{B}}{\partial s_e}\mathbf{B}^{-1}\frac{\partial\mathbf{B}}{\partial\lambda_{kj}}\right]\\
&= -e^{-s_e}tr\left[[\mathbf{B}^{-2}]_k\mathbf{Z}^T\mathbf{Z}_k\frac{\partial\mathbf{D}_k}{\partial\lambda_{kj}}\right]\\
\frac{\partial^2\log|\mathbf{B}|}{\partial\lambda_{ki}\partial\lambda_{kj}} &= tr\left[\mathbf{B}^{-1}\frac{\partial^2\mathbf{B}}{\partial\lambda_{ki}\partial\lambda_{kj}}\right] - tr\left[\mathbf{B}^{-1}\frac{\partial\mathbf{B}}{\partial\lambda_{ki}}\mathbf{B}^{-1}\frac{\partial\mathbf{B}}{\partial\lambda_{kj}}\right]\\
&= e^{-s_e}tr\left[[\mathbf{B}^{-1}]_k\mathbf{Z}^T\mathbf{Z}_k\frac{\partial^2\mathbf{D}_k}{\partial\lambda_{ki}\partial\lambda_{kj}}\right] - e^{-s_e}tr\left[[\mathbf{B}^{-1}]_k\mathbf{Z}^T\mathbf{Z}_k\frac{\partial\mathbf{D}_k}{\partial\lambda_{ki}}[\mathbf{B}^{-1}]_k\mathbf{Z}^T\mathbf{Z}_k\frac{\partial\mathbf{D}_k}{\partial\lambda_{kj}}\right]
\end{aligned}$$

where $[\mathbf{B}^{-1}]_k$ is the submatrix of $\mathbf{B}^{-1}$ from $(q_1 + \cdots + q_{k-1} + 1)$-th row to $(q_1 + \cdots + q_k)$-th row. In real data analyses, one of the categorical features has sometimes extremely high cardinality $q_1 \gg \sum_{k=2}^K q_k$. In such cases, the corresponding random effect $\mathbf{v}_1$ is assumed to be independent but $\mathbf{B} = \frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z}\mathbf{D} + \mathbf{I}_Q$ is not a diagonal, so the computation of the derivatives involves the inverse of extremely high dimensional matrix. However, here the matrix $\mathbf{B}$ is a sparse matrix such that

$$\mathbf{B}^{-1} = \begin{pmatrix}\mathbf{B}_{11} & \mathbf{B}_{12}\\ \mathbf{B}_{21} & \mathbf{B}_{22}\end{pmatrix}^{-1} = \begin{pmatrix}\mathrm{diag}\left(\frac{\lambda_1 n_{1j}}{\sigma_e^2} + 1\right) & \frac{1}{\sigma_e^2}\mathbf{Z}_1^T\mathbf{Z}\mathbf{D}\\ \frac{1}{\sigma_e^2}\mathbf{Z}^T\mathbf{Z}_1\mathbf{D}_1 & \frac{1}{\sigma_e^2}\mathbf{Z}_{-1}^T\mathbf{Z}_{-1}\mathbf{D}_{-1} + \mathbf{I}_{Q-q_1}\end{pmatrix}^{-1},$$

so the inverse $\mathbf{B}^{-1}$ can be computed by using decomposition of the submatrices.

Note here that the second derivative of $\mathbf{D}_k$ becomes zero when $\mathbf{v}_k$ is independent. Suppose that $\mathbf{v}_1, ..., \mathbf{v}_{K-1}$ are independent random effects and the last random effect $\mathbf{v}_K$ is correlated, i.e., $\mathbf{D}_k = \lambda_k\mathbf{I}_{q_k}$ for $k = 1, ..., K-1$ and $\mathbf{D}_K = \mathbf{D}_K(\boldsymbol{\lambda}_K)$ for $\boldsymbol{\lambda}_K = (\lambda_{K1}, ..., \lambda_{KJ})$. The computation can be further reduced, because for $k = 1, ..., K-1$, the first and the second derivatives of $\mathbf{D}_k$ are the identity matrix and zero matrix, respectively.

A.4. Methods-of-moments estimators in early stage

In early stage of learning, including the initial values, MMEs of variance components are used because it is computationally fast and less sensitive to the bias in the mean parameters. For $j = 1, ..., q_k$, each v_{kj} has normal distribution with mean zero and variance λ_k. Thus, we can use

$$\widehat{\lambda}_k = \frac{1}{q_k - 1} \sum_{j=1}^{q_k} (v_{kj} - \bar{v}_k)^2,$$

for the variance of random effects and

$$\widehat{\sigma}_e^2 = \frac{1}{N-1} \sum_{j=1}^{q_k} \left[y_i - \widehat{f}(\mathbf{x}_i)^T \widehat{\boldsymbol{\beta}} - \sum_{k=1}^{K} \widehat{g}_k(\mathbf{z}_{ki})^T \widehat{\mathbf{v}}_k \right]^2$$

for the variance of noise.

A.5. Proof of Theorem 1

Let $\widehat{\beta}_0^* = \widehat{\beta}_0 + \delta$ and $\widehat{\mathbf{v}}_k^* = \widehat{\mathbf{v}}_k - \delta$. Note here that δ does not affect the predicted values of the output variable $\widehat{\mathbf{y}}$, because $\widehat{\beta}_0^* + \mathbf{Z}_k \widehat{\mathbf{v}}_k^* = \widehat{\beta}_0 + \mathbf{Z}_k \widehat{\mathbf{v}}_k$. The first derivative of h-likelihood with respect to δ is given by

$$\frac{\partial h(\widehat{\boldsymbol{\theta}}^*, \widehat{\mathbf{v}}^*)}{\partial \delta} = \frac{\partial}{\partial \delta} \left(-\frac{1}{2} (\widehat{\mathbf{v}}_k - \delta)^T \widehat{\mathbf{D}}_k^{-1} (\widehat{\mathbf{v}}_k - \delta) \right) = \widehat{\mathbf{v}}_k^T \widehat{\mathbf{D}}_k^{-1} \mathbf{1}_{q_k} - \delta \cdot \mathbf{1}_{q_k}^T \widehat{\mathbf{D}}_k^{-1} \mathbf{1}_{q_k},$$

which leads to the solution $\delta = \widehat{\mathbf{v}}_k^T \widehat{\mathbf{D}}_k^{-1} \mathbf{1}_{q_k} / \mathbf{1}_{q_k}^T \widehat{\mathbf{D}}_k^{-1} \mathbf{1}_{q_k}$, where $\mathbf{1}_{q_k} = (1, ..., 1)^T$. The second derivative is given by

$$\frac{\partial^2 h(\widehat{\boldsymbol{\theta}}^*, \widehat{\mathbf{v}}^*)}{\partial \delta^2} = -\mathbf{1}_{q_k} \widehat{\mathbf{D}}_k^{-1} \mathbf{1}_{q_k} < 0,$$

since $\widehat{\mathbf{D}}_k$ should be positive definite. Thus, for given $\widehat{\boldsymbol{\theta}}$ and $\widehat{\mathbf{v}}$, the h-likelihood has the unique maximum at $\delta = \widehat{\mathbf{v}}_k^T \widehat{\mathbf{D}}^{-1} \mathbf{1}_{q_k} / \mathbf{1}_{q_k}^T \widehat{\mathbf{D}}^{-1} \mathbf{1}_{q_k}$. This implies that the adjustment (8) can always increase the h-likelihood,

$$h(\widehat{\boldsymbol{\theta}}^*, \widehat{\mathbf{v}}^*) \geq h(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{v}}),$$

and the equality holds if and only if $\widehat{\mathbf{v}}_k^T \widehat{\mathbf{D}}_k^{-1} \mathbf{1}_{q_k} = 0$.