Inferring Tweedie Compound Poisson Mixed Models with Adversarial Variational Methods

Yaodong Yang, Rui Luo, Reza Khorshidi, Yuanyuan Liu
American International Group Inc.*

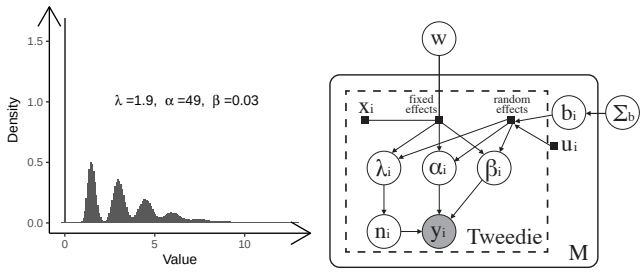
Abstract

The Tweedie Compound Poisson-Gamma model is routinely used for modelling non-negative continuous data with a discrete probability mass at zero. Mixed models with random effects account for the covariance structure related to the grouping hierarchy in the data. An important application of Tweedie mixed models is estimating the aggregated loss for insurance policies. However, the intractable likelihood function, the unknown variance function, and the hierarchical structure of mixed effects have presented considerable challenges for drawing inferences on Tweedie. In this study, we tackle the Bayesian Tweedie mixed-effects models via variational approaches. In particular, we empower the posterior approximation by implicit models trained in an adversarial setting. To reduce the variance of gradients, we reparameterize random effects, and integrate out one local latent variable of Tweedie. We also employ a flexible hyper prior to ensure the richness of the approximation. Our method is evaluated on both simulated and real-world data. Results show that the proposed method has smaller estimation bias on the random effects compared to traditional inference methods including MCMC; it also achieves a state-of-the-art predictive performance, meanwhile offering a richer estimation of the variance function.

1 Introduction

Tweedie models [1, 2] are special members in the exponential dispersion family; they specify a power-law relationship between the variance and the mean: $\text{Var}(Y) = \text{E}(Y)^{\mathcal{P}}$. For arbitrary positive $\mathcal{P}$, the index parameter of the variance function, outside the interval of $(0, 1)$, Tweedie corresponds to a particular stable distribution, *e.g.*, Gaussian ($\mathcal{P} = 0$), Poisson ($\mathcal{P} = 1$), Gamma ($\mathcal{P} = 2$), Inverse Gaussian ($\mathcal{P} = 3$). When $\mathcal{P}$ lies in the range of $(1, 2)$, the Tweedie model is equivalent to Compound Poisson–Gamma Distribution [3], hereafter *Tweedie* for simplicity. Tweedie serves as a special Gamma mixture model, with the number of mixtures determined by a Poisson-distributed random variable, parameterised by$\{\lambda, \alpha, \beta\}$ and denoted as: $Y = \sum_{i=1}^{T} G_i, T \sim \text{Poisson}(\lambda), G_i \overset{i.i.d}{\sim} \text{Gamma}(\alpha, \beta)$. Tweedie is heavily used for modelling non-negative heavy-tailed continuous data with a discrete probability mass at zero (see Fig. 1a). As a result, Tweedie gains its importance from multiple domains [4, 5], including actuarial science (aggregated loss/premium modelling), ecology (species biomass modelling), meteorology (precipitation modelling). On the other hand, in many field studies that require manual data collection, for example in insurance underwriting, the sampling heterogeneity from a hierarchy of groups/populations has to be considered. Mixed-effects models can represent the covariance structure related to the grouping hierarchy in the data by assigning common random effects to the observations that have the same level of a grouping variable; therefore, estimating the random effects is also an important component in Tweedie modelling.

*Presented at NIPS 2017 Workshop on *Advances in Approximate Bayesian Inference*. Correspondence to Yuanyuan Liu: <yuanyuan.liu@aig.com>.



(a) Tweedie Density Function (b) Tweedie Mixed-Effects model.

Figure 1: Density function and graphical model of Tweedie mixed-effect model.

Despite the importance of Tweedie mixed-effects models, the intractable likelihood function, the unknown variance index $\mathcal{P}$, and the hierarchical structure of mixed-effects (see Fig.1b) hinder the inferences on Tweedie models. Unsurprisingly, there has been little work (see Section 3 in *Appendix*) devoted to full-likelihood based inference on the Tweedie mixed model, let alone Bayesian treatments. In this work, we employ variational inference to solve Bayesian Tweedie mixed models. The goal is to introduce an accurate and efficient inference algorithm to estimate the posterior distribution of the fixed-effect parameters, the variance index, and the random effects.

2 Tweedie Mixed-Effects Models

Tweedie EDM $f_Y(y|\mu,\phi,\mathcal{P})$ equivalently describes the compound Poisson–Gamma distribution when $\mathcal{P}\in(1,2)$. Tweedie assumes Poisson arrivals of the events, and Gamma–distributed "cost" for each individual event. Judging on whether the aggregated number of arrivals N is zero, the joint density for each observation can be written as:

$$P(Y,N=n|\lambda,\alpha,\beta)=d_0(y)\cdot e^{-\lambda}\cdot\mathbb{1}_{n=0}+\frac{y^{n\alpha-1}e^{-y/\beta}}{\beta^{n\alpha}\Gamma(n\alpha)}\cdot\frac{\lambda^n e^{-\lambda}}{n!}\cdot\mathbb{1}_{n>0}, \tag{1}$$

where $d_0(\cdot)$ is the Dirac Delta function at zero. See Section 1 in *Appendix* for the connections between $\{\mu,\phi,\mathcal{P}\}$ and $\{\lambda,\alpha,\beta\}$. A mixed-effects model contains both fixed effects and random effects; graphical model is shown in Fig.1b. We denote the mixed model as: $\kappa(\mu_i)=f_{\boldsymbol{w}}(\boldsymbol{X}_i)+\boldsymbol{U}_i^\top\boldsymbol{b}_i, \boldsymbol{b}_i\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma_b})$ where $\boldsymbol{X}_i$ is the i-th row of the design matrix of fixed-effects covariates, $\boldsymbol{w}$ are parameters of the fixed-effects model which could represent linear function or deep neural network. $\boldsymbol{U}_i$ is the i-th row of the design matrix associated with random effects, $\boldsymbol{b}_i$ is the coefficients of the random effect which is usually assumed to follow a multivariate normal distribution with zero mean and covariance $\boldsymbol{\Sigma_b}$, $\kappa(\cdot)$ is the link function, and μ_i is the mean of the i-th response variable Y. In this work, we have considered the random effects on the intercept, but our method can easily be extended to the cases of random slopes.

In the context of conducting Bayesian inference on Tweedie mixed models, we define 1) the observed data $D=(\boldsymbol{x}_i,\boldsymbol{u}_i,y_i)_{i=1,\dots,M}$. 2) global latent variables $\{\boldsymbol{w},\boldsymbol{\sigma_b}\}$, we assume $\boldsymbol{\Sigma_b}$ is a diagonal matrix with its diagonal elements $\boldsymbol{\sigma_b}$; 3) local latent variables $\{n_i,\boldsymbol{b}_i\}$, indicating the number of arrivals and the random effect. The parameters of Tweedie is thus denoted by $(\lambda_i,\alpha_i,\beta_i)=f_{\lambda,\alpha,\beta}(\boldsymbol{x}_i,\boldsymbol{u}_i|\boldsymbol{w},\boldsymbol{\sigma_b})$. The latent variables thus contain both local and global ones $\boldsymbol{z}=(\boldsymbol{w},n_i,\boldsymbol{b}_i,\boldsymbol{\sigma_b})_{i=1,\dots,M}$, and they are assigned with prior distribution $P(\boldsymbol{z})$. The joint log-likelihood is computed by summing over the number of observations M by $\sum_{i=1}^{M}\log\left[P(y_i|n_i,\lambda_i,\alpha_i,\beta_i)\cdot P(n_i|\lambda_i)\cdot P(\boldsymbol{b}_i|\boldsymbol{\sigma_b})\right]$. The goal here is to find the posterior distribution of $P(\boldsymbol{z}|D)$ and make future predictions via $P(y_{\text{pred}}|D,\boldsymbol{x}_{\text{pred}},\boldsymbol{u}_{\text{pred}})=\int P(y_{\text{pred}}|\boldsymbol{z},\boldsymbol{x}_{\text{pred}},\boldsymbol{u}_{\text{pred}})P(\boldsymbol{z}|D)\,\mathrm{d}\,\boldsymbol{z}$.

3 Methods

Adversarial Variational Bayes. Variational Inference (VI) [6] approximates the posterior distribution, often complicated and intractable, by proposing a class of probability distributions $Q_\theta(\boldsymbol{z})$ (so-called inference models), and then finding the best set of parameters θ by minimizing the KL divergence between the proposal and the true distribution, *i.e.*, $\mathrm{KL}(Q_\theta(\boldsymbol{z})||P(\boldsymbol{z}|D))$. Minimizing the KL divergence is equivalent to maximizing the evidence of lower bound (ELBO) [7], expressed as Eqn.2. Optimizing the ELBO requires the gradient information of ∇_θELBO. In our experiments, we find that the model-free gradient estimation method – REINFORCE [8] fails to yield reasonable

results due to the unacceptably-high variance issue [9], even equipped with *baseline* trick [10] or *local expectation* [11]. We also attribute the unsatisfactory results to the over-simplified proposal distribution in traditional VI. Here we try to solve these two issues by employing the implicit inference models and additional variance reduction technique.

$$\theta^* = \arg\max_{\theta} \mathrm{E}_{Q_\theta(\boldsymbol{z})} \Big[-\log \frac{Q_\theta(\boldsymbol{z})}{P(\boldsymbol{z})} + \log P(D|\boldsymbol{z}) \Big] \tag{2}$$

AVB [12, 13, 14, 15, 16, 17] empowers the VI methods by using neural networks as the inference models; this allows more efficient and accurate approximations to the posterior distribution. Since a neural network is black-box, the inference model is implicit thus have no closed form expression, even though this does not bother to draw samples from it. To circumvent the issue of computing the gradient from implicit models, the mechanism of adversarial learning is introduced; an additional discriminative network $T_\phi(\boldsymbol{z})$ is used to model the first term in Eqn.2. By building a model to distinguish the latent variables that are sampled from the prior distribution $p(\boldsymbol{z})$ from those that are sampled from the inference network $Q_\theta(\boldsymbol{z})$, namely, optimising the blow equation (where $\sigma(x)$ is the sigmoid function):

$$\phi^* = \arg\min_{\phi} \Big[-\mathrm{E}_{Q_\theta(\boldsymbol{z})} \log \sigma(T_\phi(\boldsymbol{z})) - \mathrm{E}_{P(\boldsymbol{z})} \log(1 - \sigma(T_\phi(\boldsymbol{z})) \Big], \tag{3}$$

we can estimate the ratio as $\mathrm{E}_{Q_\theta(\boldsymbol{z})}[\log \frac{Q_\theta(\boldsymbol{z})}{P(\boldsymbol{z})}] = \mathrm{E}_{Q_\theta(\boldsymbol{z})}[T_{\phi^*}(\boldsymbol{z})]$. AVB considers optimizing Eqn.2 and Eqn.3 as a two-layer minmax game. We apply stochastic gradient descent alternately to find a Nash-equilibrium. Such Nash-equilibrium, if reached, is a global optimum of the ELBO [14].

Reparameterizable Random Effect. In mixed-effects models, the random effect is conventionally assumed to be $\boldsymbol{b}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma_b})$. In fact, they are reparameterisable (see Theorem.1 in *Appendix*). As such, we model the random effects by the reparameterization trick [18]; $\boldsymbol{b}_i$ is now written as $\boldsymbol{b}(\boldsymbol{\epsilon}_i) = \boldsymbol{0} + \boldsymbol{\sigma_b} \odot \boldsymbol{\epsilon}_i$ where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The $\boldsymbol{\sigma_b}$ is a set of latent variables generated by the inference network and the random effects become *deterministic* given the auxiliary noise $\boldsymbol{\epsilon}_i$. Note that when computing the gradients, we no longer need to sample the random effect directly, instead, we can back-propagate the path-wise gradients which could dramatically reduce the variance [19].

"Hyper Priors." The priors of the latent variables are fixed in traditional VI. Here we however parameterize the prior $P_\psi(\boldsymbol{w})$ by ψ and make ψ also trainable when optimising Eqn.(2). We refer ψ as a kind of *hyper prior* to $\boldsymbol{w}$. The intuition is to not constraint the posterior approximation by one over-simplified prior. We would like the prior to be adjusted so as to make the posterior $Q_\theta(\boldsymbol{z})$ close to a set of prior distributions, or a self-adjustable prior; this could further ensure the expressiveness of $Q_\theta(\boldsymbol{z})$. We can also apply the reparameterization trick if the class of prior is reparameterizable.

Variance Reduction. We find that integrating out the latent variable n_i by Monte Carlo in Eqn.1 gives significantly lower variance in computing the gradients. As n_i is a Poisson generated sample, the variance will explode in the cases where Y_i is zero but the sampled n_i is positive, and Y_i is positive but the sampled n_i is zero. This also accounts for why the direct application of REINFORCE algorithm fails to work. In practice, we find limiting the number of Monte Carlo samples between $2 - 10$, dependent on the dataset, has the similar performance as summing over to larger number.

We summarize the algorithm pseudo-code in Section 2 in *Appendix*.

4 Experiments and Results

We compare our method with six traditional inference methods on Tweedie. See Section 3 in *Appendix* for their reviews and Section 4 for the implementation details. We evaluate our method on two public datasets of modelling the aggregate loss for auto insurance polices [25], and modelling the length density of fine roots in ecology [26]. We separate the dataset in $50\%/25\%/25\%$ for train/valid/test respectively. Considering the sparsity and right-skewness of the data, we use the ordered Lorenz curve and its corresponding Gini index [27, 28] as the metric. Assuming for the i_{th}/N observations, y_i is the ground truth, p_i to be the results from the baseline predictive model, $\hat{y}_i$ to be the predictive results from the model. We sort all the samples by the relativity $r_i = \hat{y}_i/p_i$ in an increasing order, and then compute the empirical cumulative distribution as $(\hat{F}_p(r) = \frac{\sum_{i=1}^n p_i \cdot \mathbb{1}(r_i \leqslant r)}{\sum_{i=1}^n p_i}, \hat{F}_y(r) = \frac{\sum_{i=1}^n y_i \cdot \mathbb{1}(r_i \leqslant r)}{\sum_{i=1}^n y_i})$. The plot of $(\hat{F}_p(r), \hat{F}_y(r))$ is an ordered Lorenz curve, and Gini index is the twice the area between the

Table 1: The pairwise Gini index comparison with standard error based on 20 random splits

Baseline \ Model	GLM[20]	PQL [20]	Laplace [21]	AGQ[22]	MCMC[23]	TDBoost[24]	AVB
GLM (AutoClaim)	/	$-2.97_{6.28}$	$1.75_{5.68}$	$1.75_{5.68}$	$-15.02_{7.06}$	$1.61_{6.32}$	$\mathbf{9.84_{5.80}}$
PQL	$7.37_{5.67}$	/	$7.50_{6.26}$	$7.50_{5.72}$	$6.73_{5.95}$	$0.81_{6.22}$	$\mathbf{9.02_{6.07}}$
Laplace	$2.10_{4.52}$	$-1.00_{5.94}$	/	$8.84_{5.36}$	$4.00_{4.61}$	$21.45_{4.84}$	$\mathbf{20.61_{4.54}}$
AGQ	$2.10_{4.52}$	$-1.00_{5.94}$	$8.84_{5.36}$	/	$4.00_{4.61}$	$21.45_{4.84}$	$\mathbf{20.61_{4.54}}$
MCMC	$14.75_{6.80}$	$-1.06_{6.41}$	$3.12_{5.99}$	$3.12_{5.99}$	/	$7.82_{5.83}$	$\mathbf{11.88_{5.50}}$
TDBoost	$17.52_{4.80}$	$17.08_{5.36}$	$19.30_{5.19}$	$19.30_{5.19}$	$11.61_{4.58}$	/	$\mathbf{20.30_{4.97}}$
AVB	$-0.17_{4.70}$	$0.049_{5.62}$	$3.41_{4.94}$	$3.41_{4.94}$	$0.86_{4.62}$	$11.49_{4.93}$	/
GLM (FineRoot)	/	$23.18_{9.30}$	$35.87_{6.52}$	$35.87_{6.52}$	$-15.73_{10.63}$	$35.71_{6.52}$	$\mathbf{35.64_{6.53}}$
PQL	$-21.61_{8.34}$	/	$30.90_{8.86}$	$30.90_{8.86}$	$-21.45_{8.38}$	$24.19_{8.36}$	$\mathbf{28.97_{8.92}}$
Laplace	$-12.55_{7.37}$	$-14.72_{8.81}$	/	$15.07_{6.41}$	$-12.55_{7.37}$	$15.33_{7.36}$	$\mathbf{10.61_{7.20}}$
AGQ	$-12.55_{7.37}$	$-14.72_{8.81}$	$15.07_{6.41}$	/	$-12.55_{7.37}$	$15.33_{7.36}$	$\mathbf{10.61_{7.20}}$
MCMC	$17.27_{10.25}$	$22.53_{9.31}$	$35.10_{6.54}$	$35.10_{6.54}$	/	$35.10_{6.54}$	$\mathbf{34.87_{6.55}}$
TDBoost	$22.47_{6.80}$	$8.50_{9.09}$	$22.63_{6.80}$	$22.63_{6.80}$	$11.61_{6.80}$	/	$\mathbf{22.39_{6.80}}$
AVB	$-8.261_{7.66}$	$-10.886_{8.98}$	$2.13_{7.28}$	$2.13_{7.28}$	$-8.26_{7.66}$	$11.00_{7.74}$	/

Lorenz curve and the 45° line. The Gini index is not symmetrical if different baselines are used. In short, a model with a larger Gini index indicates that it has greater capability of separating the observations. In the insurance context, Gini index profiles the model's capability of distinguishing policy holders with different risk levels.

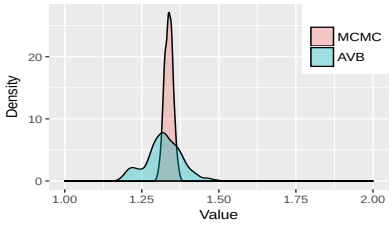


Figure 2: Posterior estimation of the index parameter $\mathcal{P}$ estimation on the AutoClaim dataset.

Algo. \ Param.	$\mathcal{P}_{\text{auto}}$	$\sigma^2_{\text{b,auto}}$	$\mathcal{P}_{\text{root}}$	$\sigma^2_{\text{b,root}}$
PQL	/	$8.94 \cdot 10^{-5}$	/	$1.0346 \cdot 10^{-3}$
LAPLACE	1.3394	$4.06 \cdot 10^{-4}$	1.4202	$6.401 \cdot 10^{-3}$
AGQ	1.3394	$4.027 \cdot 10^{-4}$	1.4202	$6.401 \cdot 10^{-3}$
MCMC	1.3458	$4.575 \cdot 10^{-3}$	1.4272	$3.490 \cdot 10^{-2}$
AVB	1.3403	$3.344 \cdot 10^{-3}$	1.4237	$2.120 \cdot 10^{-2}$

Table 2: Estimation on $\mathcal{P}$ and σ_b^2 for both AutoClaim and FineRoot data with random effects.

Gini comparison is presented in Table.1 (see corresponding Lorenz Curve plots in Fig.3,4 in *Appendix*). Here only fixed effects are considered. AVB method shows the state-of-the-art performance on both AutoClaim and FineRoot datasets, even comparatively better than the boosting tree method that is considered to have a strong performance. By inspecting the posterior estimation of $\mathcal{P}$ in Fig.2, we find AVB shows a richer representation of the index parameter, with two modals at 1.20 and 1.31. As $\mathcal{P}$ uniquely characterizes a Tweedie distribution, compared with one single value that traditional profiled likelihood method offers, flexible choices of $\mathcal{P}$ enable customized descriptions for insurance policies that may have different underlying risk profiles. Also note that AVB uses a neural sampler that does not involve any rejection procedures; unlike MCMC, it holds the potentials for large–scale predictions on high dimensional dataset.

To compare the estimations on random effects, we add "CARTYPE" in the AutoClaim data as the random effect with 6 groups and "PLANT" in the FineRoot data with 8 groups. From table 2, we can see that all the algorithms estimate $\mathcal{P}$ in a reasonable similar range while the results from MCMC and AVB are closer. The estimation of σ_b^2 has a large variance. The estimation from AGQ and Laplace methods are around ten times smaller than MCMC and AVB results while PQL produces even smaller estimations. This is consistent with our finding in the simulation study in *Appendix* that AGQ and Laplace methods tends to underestimate the random effect variance. The AVB and MCMC estimations are considered more accurate; as the estimated variance is comparatively smaller than MCMC, the AVB estimation can be believed to be more accurate.

Due to the space limit, we refer to Section 5 in *Appendix* for the study on the simulated data where the true covariance $\boldsymbol{\sigma_b}$ of the random effects and index parameter $\mathcal{P}$ are known. Our method shows accurate estimation on $\mathcal{P}$ as well as smallest estimating bias on $\boldsymbol{\sigma_b}$ under different scenarios.

5 Conclusions

We present Adversarial Variational Bayes in solving Bayesian Tweedie mixed models. We reparameterize the random effects, integrate out Poisson local latent variable, and introduce flexible hyper priors to ensure the stability of training and the richness of the approximation. Our method outperforms all traditional methods on both simulated data and real-world data; meanwhile, the estimations on the variance parameter and the random effects are competitively accurate and stable.

Disclaimer

This is not AIG work product and the paper expresses the views of the authors only in their individual capacity and not those of American International Group, Inc. or its affiliate entities.

References

[1] Bent Jorgensen. *The theory of dispersion models*. CRC Press, 1997.

[2] Bent Jorgensen. Exponential dispersion models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 127–162, 1987.

[3] Umut Simsekli, Ali Taylan Cemgil, and Yusuf Kenan Yilmaz. Learning the beta-divergence in tweedie compound poisson matrix factorization models. In *ICML (3)*, pages 1409–1417, 2013.

[4] Bent Jørgensen. *Exponential dispersion models*. Number 48. Instituto de matematica pura e aplicada, 1991.

[5] Gordon K Smyth and Bent Jørgensen. Fitting tweedie's compound poisson model to insurance claims data: dispersion modelling. *Astin Bulletin*, 32(01):143–157, 2002.

[6] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[7] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[8] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[9] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158, 1995.

[10] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

[11] Michalis Titsias RC AUEB and Miguel Lázaro-Gredilla. Local expectation gradients for black box variational inference. In *Advances in Neural Information Processing Systems*, pages 2638–2646, 2015.

[12] Ferenc Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

[13] Theofanis Karaletsos. Adversarial message passing for graphical models. *arXiv preprint arXiv:1612.05048*, 2016.

[14] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*, 2017.

[15] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[16] Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. In *Advances in Neural Information Processing Systems*, pages 496–504, 2016.

[17] Dilin Wang and Qiang Liu. Learning to draw samples: With application to amortized mle for generative adversarial learning. *arXiv preprint arXiv:1611.01722*, 2016.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *arXiv preprint arXiv:1603.00788*, 2016.

[20] Peter McCullagh. Generalized linear models. *European Journal of Operational Research*, 16(3):285–292, 1984.

[21] Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, 81(393):82–86, 1986.

[22] Qing Liu and Donald A Pierce. A note on gausshermite quadrature. *Biometrika*, 81(3):624–629, 1994.

[23] Yanwei Zhang. Likelihood-based and bayesian methods for tweedie compound poisson linear mixed models. *Statistics and Computing*, 23(6):743–757, 2013.

[24] Yi Yang, Wei Qian, and Hui Zou. Insurance premium prediction via gradient tree-boosted tweedie compound poisson models. *Journal of Business & Economic Statistics*, pages 1–15, 2017.

[25] Karen CH Yip and Kelvin KW Yau. On modeling claim frequency data in general insurance with extra zeros. *Insurance: Mathematics and Economics*, 36(2):153–163, 2005.

[26] HN De Silva, AJ Hall, DS Tustin, and PW Gandar. Analysis of distribution of root length density of apple trees on different dwarfing rootstocks. *Annals of Botany*, 83(4):335–345, 1999.

[27] Edward W Frees, Glenn Meyers, and A David Cummings. Summarizing insurance scores using a gini index. *Journal of the American Statistical Association*, 106(495):1085–1098, 2011.

[28] Edward W Jed Frees, Glenn Meyers, and A David Cummings. Insurance ratemaking and a gini index. *Journal of Risk and Insurance*, 81(2):335–366, 2014.

[29] Robert WM Wedderburn. Quasi-likelihood functions, generalized linear models, and the gaussnewton method. *Biometrika*, 61(3):439–447, 1974.

[30] Marie Davidian and Raymond J Carroll. Variance function estimation. *Journal of the American Statistical Association*, 82(400):1079–1091, 1987.

[31] John A Nelder and Daryl Pregibon. An extended quasi-likelihood function. *Biometrika*, 74(2):221–232, 1987.

[32] DR Cox and N Reid. A note on the calculation of adjusted profile likelihood. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 467–471, 1993.

[33] José C Pinheiro and Edward C Chao. Efficient laplacian and adaptive gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics*, 2012.

[34] Peter K Dunn and Gordon K Smyth. Series evaluation of tweedie exponential dispersion model densities. *Statistics and Computing*, 15(4):267–280, 2005.

[35] Peter K Dunn and Gordon K Smyth. Evaluation of tweedie exponential dispersion model densities by fourier inversion. *Statistics and Computing*, 18(1):73–86, 2008.

[36] Yi Yang, Wei Qian, and Hui Zou. A boosted tweedie compound poisson model for insurance premium. *arXiv preprint arXiv:1508.06378*, 2015.

[37] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *arXiv preprint arXiv:1603.02754*, 2016.

[38] Wei Qian, Yi Yang, and Hui Zou. Tweedies compound poisson model with grouped elastic net. *Journal of Computational and Graphical Statistics*, 25(2):606–625, 2016.

[39] Shijing Si, David A van Dyk, Ted von Hippel, Elliot Robinson, Aaron Webster, and David Stenning. A hierarchical model for the ages of galactic halo white dwarfs. *Monthly Notices of the Royal Astronomical Society*, 468(4):4374–4388, 2017.

[40] Dani Gamerman and Hedibert F Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press, 2006.

[41] Göknur Giner and Gordon K Smyth. statmod: Probability calculations for the inverse gaussian distribution. *arXiv preprint arXiv:1603.06687*, 2016.

[42] Norman E Breslow and David G Clayton. Approximate inference in generalized linear mixed models. *Journal of the American statistical Association*, 88(421):9–25, 1993.

[43] David Firth and Heather L Turner. Bradley-terry models in r: the bradleyterry2 package. *Journal of Statistical Software*, 48(9), 2012.

[44] Yi Yang, Wei Qian, and Hui Zou. A boosted tweedie compound poisson model for insurance premium. *Preprint*, 2013.

[45] Yi Yang, Wei Qian, Hui Zou, and Maintainer Yi Yang. Package tdboost. 2016.

Appendix

1 Tweedie Exponential Dispersion Model

The Tweedie model is a special member in exponential dispersion model (EDM). The density function of EDM is defined by a two-parameter function as:

$$f_Y(y|\theta,\phi) = h(y,\phi)\exp\left\{\frac{1}{\phi}(\theta y - \eta(\theta))\right\} \tag{4}$$

where θ is the natural (canonical) parameter, ϕ is the dispersion parameter, $\eta(\theta)$ is the cumulant function ensuring normalization, and $h(y,\phi)$ is the base measure independent of the parameter θ. As a member of the exponential family, EDM has the property that the mean μ and the variance $\mathrm{Var}(y)$ can be computed from the first and second order derivatives of $\eta(\theta)$ *w.r.t* θ respectively. Since there exists a one-to-one mapping between θ and μ, $\eta''(\theta)$ can be denoted as a function of the mean μ, $\eta''(\theta) = v(\mu)$, which is known as the *variance function*. One important property is that for each variance function, it characterizes a unique EDM.

The Tweedie model specifies a power-law relationship between the variance and the mean: $\mathrm{Var}(y) = \mathrm{E}(y)^{\mathcal{P}}$. There exist stable distributions for all positive $\mathcal{P} \notin (0,1)$. Corresponding to the general form of EDM in Eqn.(4), Tweedie's natural parameter θ and cumulant function $\eta(\theta)$ are:

$$\theta = \begin{cases} \log\mu & \text{if } \mathcal{P}=1 \\ \frac{\mu^{1-\mathcal{P}}}{1-\mathcal{P}} & \text{if } \mathcal{P}\neq 1 \end{cases}, \quad \eta(\theta) = \begin{cases} \log\mu & \text{if } \mathcal{P}=2 \\ \frac{\mu^{2-\mathcal{P}}}{2-\mathcal{P}} & \text{if } \mathcal{P}\neq 2 \end{cases} \tag{5}$$

And the normalization quantity $h(y,\phi)$ can be obtained as:

$$\begin{aligned} h(y,\phi) &= \begin{cases} \frac{\mu^{1-\mathcal{P}}}{1-\mathcal{P}} & \text{if } y=0 \\ \frac{1}{y}\sum_{n=1}^{\infty} h_n(y,\phi,\mathcal{P}) & \text{if } y>0 \end{cases}, \\ h_n(y,\phi,\mathcal{P}) &= \frac{y^{n\alpha}}{y(\mathcal{P}-1)^{n\alpha}\phi^{n(1+\alpha)}(2-\mathcal{P})^n n!\,\Gamma(n\alpha)}, \end{aligned} \tag{6}$$

where $\sum_{n=1}^{\infty} h_n(y,\phi,\mathcal{P})$ is also known as Wright's generalized *Bessel* function. Note that $h(y,\phi)$ in Tweedie model is also a function of $\mathcal{P}$.

Tweedie EDM $\{\mu,\phi,\mathcal{P}\}$ with $\mathcal{P}\in(1,2)$ equivalently describes the Compound Poisson–Gamma distribution (CPGD) that is parameterized by $\{\lambda,\alpha,\beta\}$. The connection between the parameters $\{\lambda,\alpha,\beta\}$ of CPGD and the parameters $\{\mu,\phi,\mathcal{P}\}$ of Tweedie model follows as:

$$\begin{cases} \lambda = & \frac{\mu^{2-\mathcal{P}}}{\phi(2-\mathcal{P})} \\ \alpha = & \frac{2-\mathcal{P}}{\mathcal{P}-1} \\ \beta = & \phi(\mathcal{P}-1)\mu^{\mathcal{P}-1} \end{cases}, \quad \begin{cases} \mu = & \lambda\alpha\beta \\ \mathcal{P} = & \frac{\alpha+2}{\alpha+1} \\ \phi = & \frac{\lambda^{1-\mathcal{P}}(\alpha\beta)^{2-\mathcal{P}}}{2-\mathcal{P}} \end{cases} \tag{7}$$

Equivalent to Eqn.(1), the joint distribution $P(Y,N|\mu,\phi,\mathcal{P})$ has a closed-form expression. By replacing Eqn.(7) into Eqn.(1), we get the joint density function represented by $\{\mu,\phi,\mathcal{P}\}$ as:

$$\begin{aligned} P(Y,N|\mu,\phi,\mathcal{P}) = &\left[\exp\left(-\frac{\mu^{2-\mathcal{P}}}{\phi(2-\mathcal{P})}\right)\right]^{\mathbb{1}_{n=0}} \\ &* \Big[\exp\Big(n\Big(-\frac{\log(\phi)}{\mathcal{P}-1} + \frac{2-\mathcal{P}}{\mathcal{P}-1}\log\Big(\frac{y}{\mathcal{P}-1}\Big) \\ &\quad - \log(2-\mathcal{P})\Big) - \log\Gamma(n+1) \\ &\quad - \frac{1}{\phi}\Big(\frac{\mu^{1-\mathcal{P}}y}{\mathcal{P}-1} + \frac{\mu^{2-\mathcal{P}}}{2-\mathcal{P}}\Big) \\ &\quad - \log\Gamma\Big(\frac{2-\mathcal{P}}{\mathcal{P}-1}n\Big) - \log(y)\Big)\Big]^{\mathbb{1}_{n>0}}. \end{aligned} \tag{8}$$

2 Algorithm Design

Algorithm 1 AVB for Bayesian Tweedie Mixed-Effects Model

Input: training data $D = (\boldsymbol{x}_i, \boldsymbol{u}_i, y_i)$, $i = 1, ..., M$
while $\{\boldsymbol{\theta}, \boldsymbol{\psi}\}$ not converged **do**
  for each epoch **do**
    Sample noises $\boldsymbol{\epsilon}_{i,j=1,...,M} \sim N(\mathbf{0}, \boldsymbol{I})$,
    Map noise to prior $\boldsymbol{w}_j^P = P_{\boldsymbol{\psi}}(\boldsymbol{\epsilon}_j) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_j$, (trainable $\boldsymbol{\mu}, \boldsymbol{\sigma}$ are the "hyper priors")
    Map noise to posterior $\boldsymbol{z}_i^Q = (\boldsymbol{w}_i, \boldsymbol{\sigma_b})^Q = Q_\theta(\boldsymbol{\epsilon}_i)$,
    Reparameterize random effects $\boldsymbol{b}_j^P = \mathbf{0} + \boldsymbol{\sigma_b} \odot \boldsymbol{\epsilon}_j$
    Minimize Eqn.(3) over $\boldsymbol{\phi}$ via the gradients:
    $\nabla_{\boldsymbol{\phi}} \frac{1}{M} \sum_{i=1}^M \Big[-\log \sigma(T_{\boldsymbol{\phi}}(\boldsymbol{z}_i^Q)) - \log(1 - \sigma(T_{\boldsymbol{\phi}}(\boldsymbol{z}_j^P)) \Big]$, $\boldsymbol{z}_j^P = (\boldsymbol{w}_j, \boldsymbol{b}_j)^P$
  end for
  Sample noises $\boldsymbol{\epsilon}_{i=1,...,M} \sim N(\mathbf{0}, \boldsymbol{I})$,
  Map noise to posterior $\boldsymbol{z}_i^Q = (\boldsymbol{w}_i, \boldsymbol{\sigma_b})^Q = Q_\theta(\boldsymbol{\epsilon}_i)$,
  Sample a minibatch of $(\boldsymbol{x}_i, \boldsymbol{u}_i, y_i)$ from D,
  Minimize the ELBO over $\boldsymbol{\theta}, \boldsymbol{\psi}$ via the gradients:
  $\nabla_{\boldsymbol{\theta},\boldsymbol{\psi}} \frac{1}{M} \sum_{i=1}^M \Big[-T_{\boldsymbol{\phi}^*}(\boldsymbol{z}_i^Q) + \sum_{j=1}^T P(y_i | n_j, \boldsymbol{w}, \boldsymbol{b}_i, \boldsymbol{\sigma_b}, \boldsymbol{x}_i, \boldsymbol{u}_i) \cdot P(n_j | \boldsymbol{w}, \boldsymbol{x}_i, \boldsymbol{u}_i) \cdot P(\boldsymbol{b}_i \mid \boldsymbol{\sigma_b}) \Big]$
end while

Theorem 1 (Reparameterizable Random Effects)**.** *Random effects are generally reparameterizable. The reparameterization trick is applicable to random effects from most of commonly used distributions.*

Proof. Here we show how to reparametrize multivariate normal random effects. Suppose the i-th random effect variable $\boldsymbol{b}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma_b})$. Because $\boldsymbol{\Sigma_b}$ is positive definite, it is congruent to the identity matrix $\boldsymbol{I}$. So there exists an invertible matrix $\boldsymbol{\sigma_b}$ such that $\boldsymbol{\Sigma_b} = \boldsymbol{\sigma_b} \odot \boldsymbol{\sigma_b}^\top$. Set $\boldsymbol{\epsilon}_i = \boldsymbol{\sigma}^{-1} \odot (\boldsymbol{b}_i - \boldsymbol{\mu}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, so we have $\boldsymbol{\epsilon}_i = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_i$.

For other "location-scale" type distributions, like Laplace, Elliptical, Student's t, Logistic, etc, we can use the standard distribution (with location $= 0$, scale $= 0$) as the random noise, and let random effects $\boldsymbol{b}_i =$ location $+$ scale $\odot \epsilon$. See the Section 2.4 in [18] for other distribution families. □

3 Related Work

To date, most practises of Tweedie modelling are conducted within the quasi-likelihood (QL) framework [29] where the density function is approximated by the first and second order moments. Despite its wide applicability, QL has the deformity of requiring the index parameter p to be specified beforehand. QL does not come with the schema of estimating the variance function, even though the estimation of the index parameter plays an important role in hypothesis testing and uncertainty evaluation [30]. Extended quasi-likelihood (EQL) [31], together with the profile likelihood method [32] is proposed to fill the gap of index estimation for QL. Nonetheless, EQL is not capable of handling exact zero values; as a result, EQL requires observed data to be adjusted by an ad-hoc small positive number which again could "jeopardize" the inferences. Most important, a consistent concern on QL methodology is that the moment estimator may not be asymptotically equivalent to the maximum likelihood estimator [33], and it has been pointed out that for Tweedie model the profile likelihood approach should be based on the true likelihood rather than the EQL [23].

It was not until recently that numerical approximations to the Tweedie density function appeared. Well-known approaches include series expansion method [34] and Fourier inversion method [35]. Series expansion method sums an infinite series via a Taylor expansion of the characteristic function. Fourier inversion method applies an inversion of the characteristic function through numerical integration techniques for oscillating function. These approximation methods enable straightforward and fast evaluation of the normalization term, and thus can approximate the density function effectively. Based on the these, profiled likelihood method can access the "true" likelihood function directly, and more accurate estimation of $\mathcal{P}$ are feasible. [36] and [37] applied the gradient boosted trees that

try to incorporate non-linear interactions in estimating the mean value of the Tweedie model. [38] investigated the grouped elastic net and Lasso methods for the Tweedie model. Nonetheless, the dependence on density approximations also makes the above algorithms vulnerable to numerical issues. Series expansion methods are noted to suffer from unbounded increment in the number of required terms. Fourier inversion methods might fail to work in the small range of observed variables, and are also computationally heavy to implement.

When the Tweedie model is expended to incorporate the random effects, the profiled quasi-likelihood (PQL) method still suffers from the numerical restrictions. Likelihood-based inference methods require approximation of the integral over the distribution of the random effect. Advances in numerical approximation like Laplace approximation [21] and adaptive Gauss-Hermite quadrature (AGQ) [22] are then applied to solve the mixed model. Due to the complex density of the Tweedie model, the normalising term in Eq.(4) depends on both the dispersion parameter ϕ and the index parameter $\mathcal{P}$, which makes the profiled likelihood method intractable.

The expectation of a latent variable for an intractable likelihood function can typically be estimated using the EM algorithm in an empirical Bayes approach or MCMC methods in full Bayesian approach [39]. [3, 23] proposed solving the Tweedie mixed-effect models by explicitly exploiting the latent variable in the Bayesian formulation. Both MCMC and Monte Carlo EM methods have been applied. [23] compared the latent variable approach with the density function approximations aforementioned, and found that although MCMC methods allow for the estimation of variance function, without the need to directly evaluate the density function, they are computationally demanding on the high dimensional dataset, and tend to be subject to Monte Carlo errors, where dedicated supervisions are needed in order to adjust the Monte Carlo error and the sampling variance [40].

4 Implementation Details

• **GLM**. We fit the Tweedie GLM model with the *tweedie* function in the *statmod* package [41]. The index parameter p is set to 1.34 for AutoClaim dataset and 1.43 for FineRoot dataset, which is mean value of the posterior distribution estimated in MCMC result.

• **PQL**. Penalised quasi-likelihood method proposed by [42] is implemented using the *glmmPQL* function in the R package *BradleyTerry2* [43].

• **Laplace**. Laplacian approximation [21] likelihood-based methods that can are carried out with the R package *cplm* [23].

• **AGQ**. Adaptive Gauss–Hermite Quadrature [22]. The number of knots is set to 15 in the AGQ implementation using function *cpglmm* in R package *cplm* [23].

• **MCMC**. *bcplm* function in the *cplm* [23] package is used for the implementation of the Markov Chain Monte Carlo. To be consistent with studies in [23], we use the same settings of priors. They are specified as: $\beta_j \sim N(0, 1002)$, $j = 0, ..., 56$ for AutoClaim dataset and $j = 0, ..., 4$, $\phi \sim U(0, 100)$, $p \sim U(1, 2)$, and $\theta^2 \sim Ga(0.001, 1000)$. All the priors distributions are non-informative. The random-walk Metropolis-Hastings algorithm is used for parameter simulations except for the variance component due to conjugacy of its prior and posterior distribution. The proposal variances of the (truncated) Normal proposal distributions are tuned according to sample variances by running 5000 iterations before the MCMC so that the acceptance rate will be about 50% for the Metropolis-Hastings updates. The MCMC procedure contains 10000 iterations of three parallel chains with burin-in period set to be the first 1000 iterations.

• **TDBoost**. The TDBoost proposed by [44] is implemented by the R package *TDBoost* [45]. The number of trees is set to 1000 and the depth of interaction is 1 to be consistent with other methods. The tree is trained with early stopping by selecting the best iteration via a validation set.

• **AVB**. In the AutoClaim dataset. feature dimension is 56. For each observation, it estimates $\{\alpha, \beta, \lambda\}$. $Q_\phi(\boldsymbol{z})$ takes in 10 dimensional source of noise, initialised by $\text{Norm}(0, 1)$, 2 hidden layers with 10 neurons each, outputs vector of size 171 $(56 * 3 + 3)$. For prior distribution $P_\psi(\boldsymbol{z})$ network, input layer is size 171, initialise with $\text{Norm}(0, 1)$, outputs size 171 vector computed as element-wise computation *noise*weights + biases*. Discriminator input size 171, with 3 hidden layers of size 20, and output size is 1. The learning rate is 0.003 exponentially decay with the coefficient 0.99. For the implementation of FineRoot data, all the setting are based on 4 feature dimensions, other settings are changed accordingly.

5 Additional Results

5.1 AutoClaim Data

The dataset contains 10296 auto insurance policy records. Each record includes a driver's aggregated claim amount (y_i) for the policy he bought, as well as 56 features ($\boldsymbol{x}_i$). Features describe the general information about the driver and his insured vehicle, for example, distance to work, marriage status, whether driving license has been revoked in the past 7 years, *etc*. The definition of each column can be found in page 24 in [24]. We are interested in predicting the distribution of the aggregated claim loss for each insurance policy holder, i.e., the driver, based on the provided features. Unsurprisingly, the distribution of total claim amount is sparse and highly right-skewed, with 61% of the drivers having no claims at all, and 9% of the policy holders making over 65% of the total claims. The data are pre-processed following the same rules as [23].

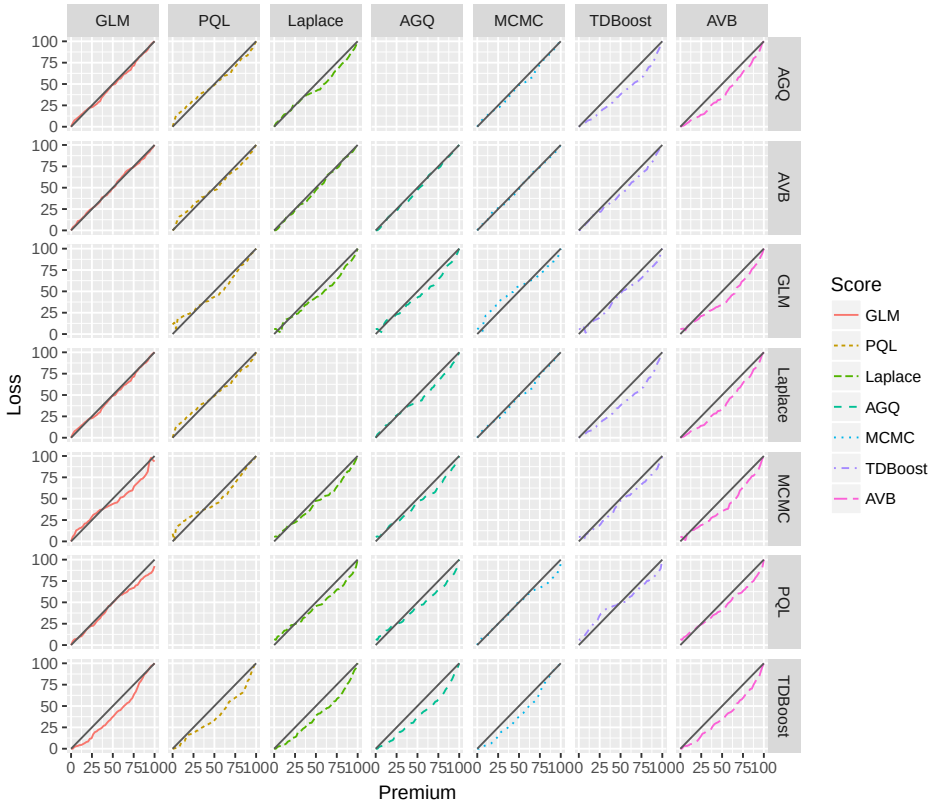


Figure 3: Pairwise ordered Lorenz curve for the experiment on AutoClaim Dataset. Each row shows the results when the algorithm is set as the baseline model. For example, the first row of subplots show the comparative performance of 6 other algorithms over AGQ. Twice the area between the Lorenz curve and the 45° line is the Gini index in Table.1. The larger the Gini index, the more comparatively better predictive power the model of that column has. AVB has larger Gini than all other algorithms.

5.2 FineRoot Data

The FineRoot dataset from ecology experiments conducted by [26] models the impact factors of the length density of fine roots. The response variable, root length density (RLD) is considered following highly right-skewed Tweedie distribution with 37.76% of the 511 total records being exactly zero. Four potential factors are included as covariates. The factor "plant" represents the randomized block design in the study of [26]. Hence, it is analyzed as random effect in later studies.

.

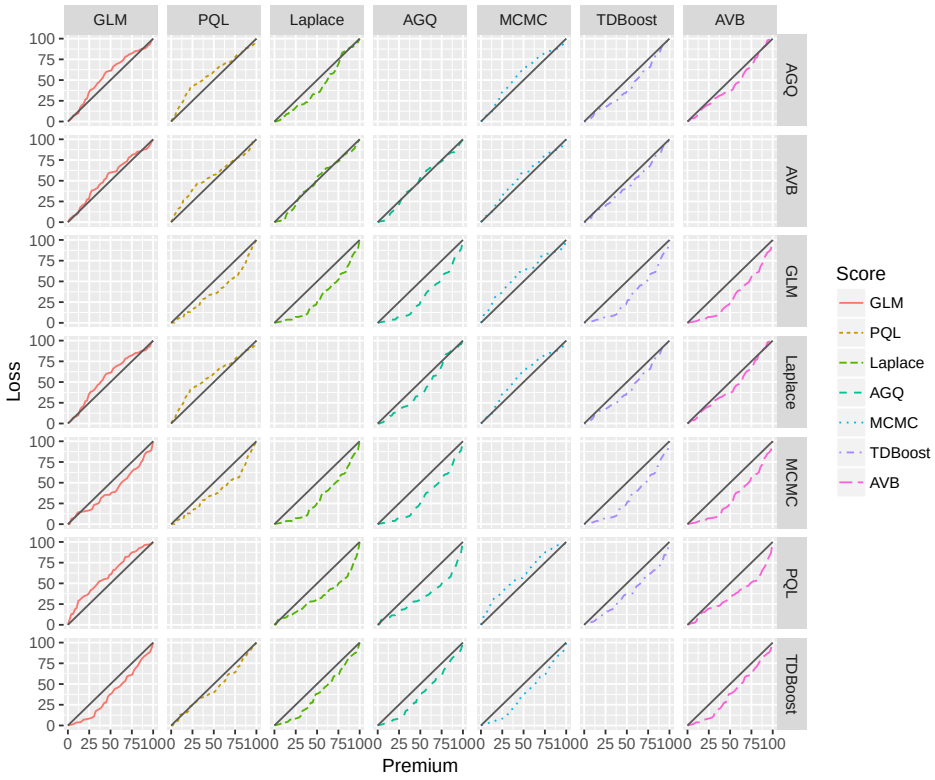


Figure 4: Pairwise ordered Lorenz curve for the experiment on FineRoot Dataset. Each row shows the results when the algorithm is set as the baseline model. For example, the first row of subplots show the comparative performance of 6 other algorithms over AGQ. Twice the area between the Lorenz curve and the 45° line is the Gini index in Table.1. The larger the Gini index, the more comparatively better predictive power the model of that column has. AVB has larger Gini than all other algorithms.

.

5.3 Simulation Study

Table 3: Simulation study on $\mathcal{P}$ and σ_b with mean and estimation bias (in percentage %)

Parameter / Algorithm	$\mathcal{P}_1 = 1.5$	$\sigma_{b_1} = 0.5$	$\mathcal{P}_2 = 1.5$	$\sigma_{b_2} = 0.05$	$\mathcal{P}_3 = 1.5$	$\sigma_{b_3} = 0.5$	$\mathcal{P}_4 = 1.5$	$\sigma_{b_4} = 0.05$
PQL	/	$0.4379_{-12.42}$	/	$0.0268_{+46.36}$	/	$0.4534_{-18.60}$	/	$0.0313_{-37.36}$
LAPLACE	$1.4956_{-0.28}$	$0.4385_{-12.28}$	$1.4958_{-0.27}$	$0.0260_{-47.93}$	$1.4956_{+0.29}$	$0.4541_{-9.16}$	$1.4956_{-0.29}$	$0.0304_{-39.15}$
AGQ	$1.4958_{-0.28}$	$0.4381_{-12.36}$	$1.4958_{-0.27}$	$0.0260_{-47.93}$	$1.4956_{+0.29}$	$0.4534_{-9.30}$	$1.4956_{-0.29}$	$0.0304_{-39.10}$
MCMC	$\underline{1.4997_{-0.02}}$	$0.0326_{+34.75}$	$\underline{1.5001_{+0.00}}$	$0.0781_{+56.23}$	$\underline{1.4997_{-0.01}}$	$\underline{0.4961_{-0.77}}$	$\underline{1.4999_{+0.00}}$	$0.0745_{+49.06}$
AVB	$1.4988_{-0.08}$	$\underline{0.4456_{-10.88}}$	$1.4992_{-0.08}$	$\underline{0.0403_{-19.42}}$	$1.4986_{-0.09}$	$0.4667_{-6.66}$	$1.4987_{-0.08}$	$\underline{0.0407_{+18.83}}$

The number of groups in random effects (g), the variance index parameters $\mathcal{P}$, and σ_b^2 have great impacts on the performance of the estimation. We then design four groups of simulation data by $g_1 = 5, g_2 = 5, g_3 = 10, g_4 = 10$ to test the estimation robustness. Five dimensional fixed-effect covariates are generated from standard normal distribution. The true coefficients generated are all ones for each of the covariates with a global intercept set to -1. Random effects are generated according to the number of groups and the true σ_b value (listed in the head). The fixed-effect covariates and random effects are added together for μ, and for each μ, the response variable is generated with EDM $\phi = 1$ and $\mathcal{P} = 1.5$ under a log-link. For each simulation scenario S = 100 data sets are generated with each having 500 data points. The average estimation of parameter, θ is calculated as $\hat{\theta} = \sum_{i=1}^{S} \theta_i / S$ and relative bias with $(\hat{\theta} - \theta)/\theta$ where θ is the true parameter value. Note that PQL method is not capable of estimating $\mathcal{P}$, TDBoost is by nature not designed for mixed-effects model,

therefore we do not present them. In Table.3, we see that the performance of the estimation on $\mathcal{P}$ is accurate for all inference methods; there is little difference among different inference methods even though MCMC has the most accurate estimation. However the estimation of σ_b is challenging in essence. With a larger number of groups or bigger random effect variance, the relative accuracy of the estimations of all algorithms except MCMC increases. AVB enjoys the smallest bias in most of the cases, especially when σ_b is relatively small, it largely outperforms other algorithms.