Inferring Tweedie Compound Poisson Mixed Models with Adversarial Variational Bayes

Yaodong Yang, Rui Luo, Reza Khorshidi, Yuanyuan Liu

Tweedie Distribution

Tweedie Compound Poisson models are heavily used for modelling non-negative continuous data with a discrete probability spike at zero.

Examples:

- total claim loss for an insurance policy
- total precipitation for a certain period

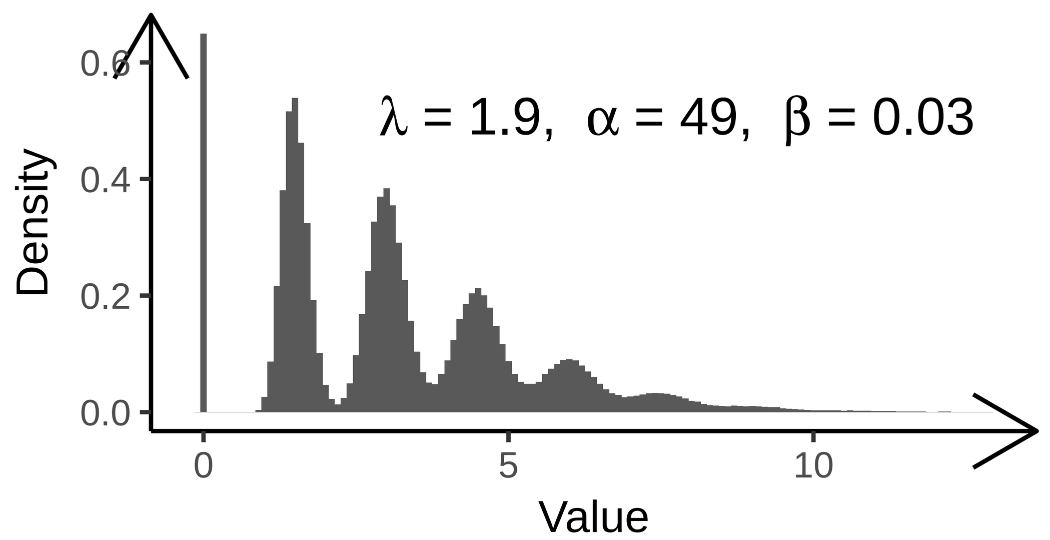

Figure 1: Tweedie Probability Density Function

- $N \sim \text{Poisson}(\lambda)$ – number of the events
- $G_i \overset{i.i.d}{\sim} \text{Gamma}(\alpha, \beta)$ – "value" of each event
- $Y = \sum_{i=1}^{N} G_i$ – total value of all events
- $\lambda > 0, \alpha > 0, \beta > 0$ – Tweedie parameters
- $\mu = \lambda\alpha\beta$ – Tweedie expected value
- $1 < p < 2 = \dfrac{\alpha+2}{\alpha+1}$ – Tweedie index parameter
- $Var(Y) = \mu^p$ – Tweedie power law

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

where $d_0(\cdot)$ is the Dirac Delta function at zero.

Affiliation

1 American International Group (AIG), Science Team, London

Mixed-Effects Models

Goal: to learn the distribution of Tweedie distribution parameters $(\lambda_i, \alpha_i, \beta_i)$ for each record x_i, y_i in the dataset D.

Model: $(\lambda_i, \alpha_i, \beta_i) = f(x_i, u_i|w, \Sigma_b)$, where

- $f(\cdot, w)$ – a fully connected neural network,
- w – its global parameters.

Thus, we are looking for the posterior distribution $P(w|D)$, given the prior distribution $P(w)$. Once it is found, inference of y_i distribution is done by:

- sampling w and Σ_b,
- inferring $(\lambda_i, \alpha_i, \beta_i) = f(x_i, u_i|w, \Sigma_b)$,
- sampling y_i from Tweedie$(x|\lambda_i, \alpha_i, \beta_i)$

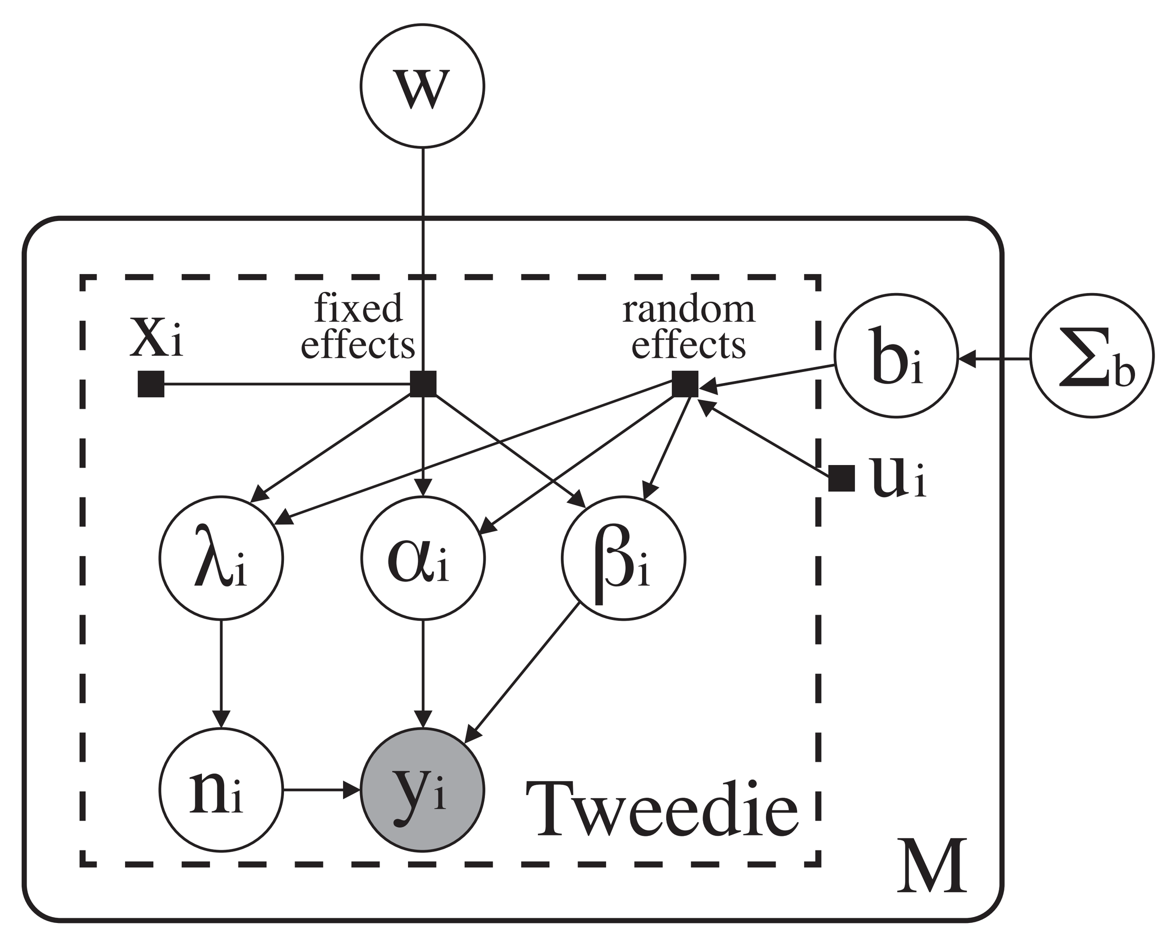

Figure 2: Tweedie Mixed-Effects Models

We find $P(w|D)$ using **Variational Inference**:

- Approximate $P(w|D)$ by $Q_\theta(w)$
- Minimize KL divergence $KL(Q_\theta(w)||P(w|D))$ by minimizing ELBO $= \mathrm{E}_{Q_\theta(\mathbf{w})}\left[\log \dfrac{P(D, \mathbf{w})}{Q_\theta(\mathbf{w})}\right]$

Therefore, the optimization problem is to find

$$\theta^* = \arg\max_\theta \mathrm{E}_{Q_\theta(\mathbf{w})}\left[-\log \frac{Q_\theta(\mathbf{w})}{P(\mathbf{w})} + \log P(D|\mathbf{w})\right]$$

Training details

We employ **Adversarial Variational Bayes** method.

- Discriminator $T_\phi(\cdot)$ learns to differentiate prior weights $P(\mathbf{w})$ and posterior weights $Q_\theta(\mathbf{w})$. Using the optimal discriminator parameters ϕ^* we can compute the distribution log-ratio as:
$$\mathrm{E}_{Q_\theta(\mathbf{z})}\left[\log \frac{Q_\theta(\mathbf{z})}{P(\mathbf{z})}\right] = \mathrm{E}_{Q_\theta(\mathbf{z})}\left[T_{\phi^*}(\mathbf{z})\right]$$
- Prior distribution $P(\mathbf{w})$ tunes its hyperparameters to decrease the accuracy of $T_\phi(\cdot)$. We use

$$\mathbf{w} \sim z_{\mu,\sigma}(\epsilon) = \mu + \sigma \odot \epsilon, \epsilon \sim \mathcal{N}(0, 1)$$

- Generator $Q_\theta(\cdot)$ is itself a neural network with parameters θ, which learns to simultaneously:
  1) decrease the accuracy of $T_\phi(\cdot)$,
  2) maximize Tweedie likelihood $P(Y|\lambda, \alpha, \beta)$

In order to avoid sampling n_j, Tweedie Likelihood is approximated by a limited summation over n_j:

$$P(y_i|\mathbf{w}; \mathbf{x_i}) = \sum_{j=1}^{T} P(y_i|n_j, \mathbf{w}; \mathbf{x_i}) \cdot P(n_j|\mathbf{w}; \mathbf{x_i})$$

Results

We evaluate our method by predicting losses for auto insurance policies.

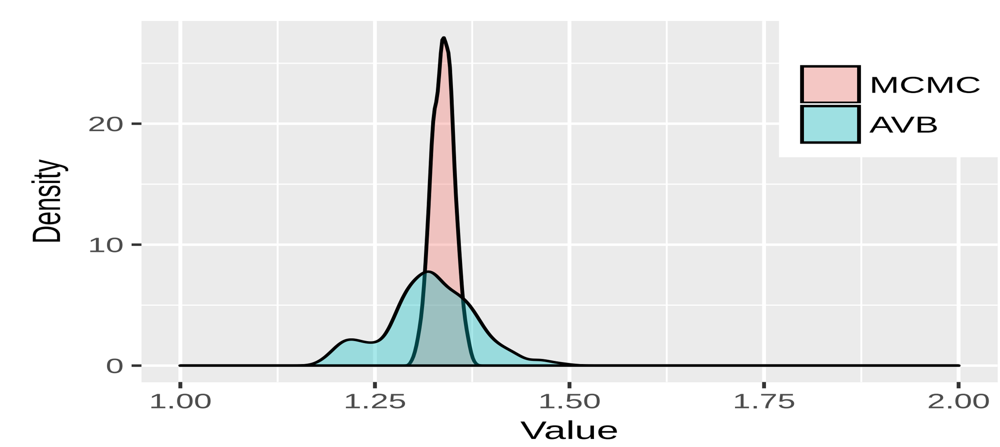

Figure 3: Posterior estimation of the index parameter $\mathcal{P}$ estimation on the AutoClaim dataset.

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

Table 2: The pairwise Gini index comparison with standard error based on 20 random splits

Baseline \ Model	GLM	PQL	Laplace	AGQ	MCMC	TDBoost	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	$-8.261_{7.66}$	$-10.886_{8.98}$	$2.13_{7.28}$	$2.13_{7.28}$	$-8.26_{7.66}$	$11.00_{7.74}$	/

Summary: AVB method shows the state-of-the-art performance, even compared with the TDBoost method which is considered to be the strongest.