City Research Online

City, University of London Institutional Repository

Citation: Asimit, V., Badescu, A., Chen, Z. & Zhou, F. (2025). Efficient and proper Generalised Linear Models with power link functions. Insurance: Mathematics and Economics, 122, pp. 91-118. doi: 10.1016/j.insmatheco.2025.02.005

This is the published version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/34775/

Link to published version: https://doi.org/10.1016/j.insmatheco.2025.02.005

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

City Research Online: http://openaccess.city.ac.uk/ publications@city.ac.uk

Insurance Mathematics and Economics 122 (2025) 91–118

Contents lists available at ScienceDirect

Insurance: Mathematics and Economics

journal homepage: www.elsevier.com/locate/ime

Efficient and proper generalised linear models with power link functions

Vali Asimit [a,*], Alexandru Badescu [b], Ziwei Chen [a], Feng Zhou [c]

[a] Bayes Business School, City St George's, University of London, 106 Bunhill Row, EC1Y 8TZ, UK
[b] Department of Mathematics and Statistics, University of Calgary, Calgary, AB T2N 1N4, Canada
[c] University of Cambridge, Cambridge, CB2 0SR, UK

ARTICLE INFO

JEL classification:
C13
C35
C44

Keywords:
Generalised linear model
Tweedie regression
Power link function
Self-concordance

ABSTRACT

The generalised linear model is a flexible predictive model for observational data that is widely used in practice as it extends linear regression models to non-Gaussian data. In this paper, we introduce the concept of a properly defined generalised linear model by requiring the conditional mean of the response variable to be properly mapped through the chosen link function and the log-likelihood function to be concave. We provide a comprehensive classification of proper generalised linear models for the Tweedie family and its popular subclasses under different link function specifications. Our main theoretical findings show that most Tweedie generalised linear models are not proper for canonical and log link functions, and identify a rich class of proper Tweedie generalised linear models with power link functions. We provide a novel interpretability methodology for power link functions that is mathematically sound and very simple, which could help the adoption of such a link function that has not been used much in practice for its lack of interpretability. Using self-concordant log-likelihoods and linearisation techniques, we provide novel algorithms for estimating several special cases of proper and not proper Tweedie generalised linear models with power link functions. The effectiveness of our methods is determined through an extensive numerical comparison of our estimates and those obtained using three built-in packages, **MATLAB** *fitglm*, **R** *glm2* and **Python** *sm.GLM* libraries, which are all implemented based on the standard Iteratively Reweighted Least Squares method. Overall, we find that our algorithms consistently outperform these benchmarks in terms of both accuracy and efficiency, the largest improvements being documented for high-dimensional settings. This is concluded for both simulated data and real data, which shows that our optimisation-based GLM implementation is a good alternative to the standard Iteratively Reweighted Least Squares implementations available in well-known software.

1. Introduction

1.1. Literature review and main goals

Generalised linear modelling (GLM) is a predictive model for observational data which creates a bridge between statistics and machine/statistical learning. That is, GLM provides not only statistical goodness of fit evidence (Nelder and Wedderburn, 1972; McCullagh et al., 1989; Bickel and Doksum, 2015) but also machine/statistical learning evidence such as feature/variable selection (Kuo and Mallick, 1998; Hastie et al., 2001).

GLMs have been successfully implemented in different research fields, and it is vastly used in insurance risk modelling; see e.g., (Debón et al., 2008) for mortality modelling, (Eling and Wirfs, 2019) for cyber risk modelling, (Delong et al., 2021) for insurance pricing, etc. Most of these applications assume independent observations, but insurance applications may require non-independent data, and one example is longitudinal data, and this setting is investigated in Antonio and Beirlant (2007).

The basic GLM requires assumptions about two key quantities, the underlying parametric distribution and the choice of *link function (LF)*. The estimation procedure is based on an optimisation algorithm if the most common estimation method is chosen, i.e. *maximum likelihood estimation (MLE)*. The asymptotic theory of M-estimators requires a concave log-likelihood function, which is the ideal setting so that efficient and stable estimates are obtained; the existence and uniqueness of the MLE

* Corresponding author.
E-mail addresses: asimit@city.ac.uk (V. Asimit), abadescu@math.ucalgary.ca (A. Badescu), Ziwei.Chen.3@bayes.city.ac.uk (Z. Chen), feng.zhou@mrc-bsu.cam.ac.uk (F. Zhou).

https://doi.org/10.1016/j.insmatheco.2025.02.005
Received 6 March 2024; Received in revised form 10 February 2025; Accepted 22 February 2025
Available online 26 February 2025
0167-6687/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

estimator is an essential assumption that requires some regularity conditions (Wedderburn, 1976; Mäkeläinen et al., 1981). Consequently, we introduce the concept of a *proper* GLM which requires the conditional mean of the response variable to be properly mapped through the chosen LF and for the log-likelihood function to be well-defined and concave. Since the GLM literature typically relies on exponential dispersion models (Jørgensen, 1987), *our first main goal* is to provide a classification of proper GLMs under this modelling assumption for different LF specifications. This allows the modeller to reduce the numerical issues and understand which combination of the parametric family and LF would provide the best possible setting for implementation purposes. The most common LFs belong to the class of *log* or *power* functions, see e.g. McCullagh et al. (1989) and Bickel and Doksum (2015), and thus the main focus will be on these choices.

The most popular algorithms for fitting exponential dispersion GLMs are *Iteratively Reweighted Least Squares (IRLS)*, *Broyden-Fletcher-Goldfarb-Shanno (BFGS)* and *Limited-memory BFGS (L-BFGS)*. IRLS is the standard algorithm which is reasonably scalable when the number of covariates/features is smaller than the sample size. However, IRLS requires inverting the Hessian matrix at every step, which is computationally challenging in non-sparse problems when either the number of features/covariates or the sample size is small. A remedy for this is given by either BFGS or L-BFGS, where the inverse of the Hessian is approximated so that it is feasible to solve higher-dimensional GLM Regressions. *The second main goal* of the paper is to identify viable alternative estimation algorithms to IRLS. Given that the underlying distribution of the response variable is parametrised according to an exponential dispersion family, the MLE could also be obtained via the vanilla Newton's method, which by design is the same as IRLS if the *canonical* LF is in place; the application of Newton's method is also known as the Fisher Scoring method in the GLM literature. Our aim is to improve this estimation method for both proper and not proper GLM settings, by making use of the mathematical properties of *power* LFs. For convex problems, Newton's algorithm can be further refined if the objective function is in addition *self concordant (SC)*, i.e. a convex function whose third derivative is bounded relative to the second derivative in the interior of its domain.[1] This property allows defining an augmented Newton's method which requires a fewer number of iterations for convergence to the optimal solution, see e.g. Boyd and Vandenberghe (2004) or Nesterov (2004) for further details on SC and their fast convergence iterative methods. Since the log-likelihood associated to special cases of Tweedie GLMs (e.g. Poisson and Gamma) equipped with some particular *power* LF specifications is an SC function, we rely on this method for implementing them.[2] For non-convex problems, which is typically the case for many exponential dispersion GLMs (e.g. Inverse Gaussian with *power* LFs), the use of standard IRLS-type algorithms leads to significant computational problems, as illustrated in the next subsection. In such cases, one could either construct bespoke optimisation algorithms designed to tackle a specific problem or rely on mainstream optimisation tools (e.g. generic interior-point methods) if the former is not available. In this paper, we also aim to identify tractable solutions for non-convex GLM instances by exploring linearisation techniques, see e.g. Boyd et al. (2011b).

[1] In a GLM context, a modified version of the SC property with a different control of the third derivative has been used by Bach (2010) for analysing the statistical properties of Logistic Regressions.

[2] We should note that the augmented Newton's method for SC objective functions still requires the inverse of the Hessian matrix, but in a much lower number of iterations, which reduces the computational time. If the size of the GLM is large, then one may need compromises like those given by BFGS and L-BFGS algorithms where the inverse of the Hessian is efficiently computed, although we do not recommend this choice unless the augmented Newton's method is overwhelmed by the size of the problem. In conclusion, the SC objective functions are expected to bring an improvement to IRLS, and large sized problems could be combined with the Hessian inverse approximations brought by L-BFGS or BFGS.

Finally, we would like to reiterate that the IRLS methodology approximates the MLE estimates, though IRLS is a very general method that is implemented in various forms in all well-known software. The differences between various implementations are given by bespoke solutions to overcome the lack of convergence that is not guaranteed by IRLS, and therefore, software engineers came up with different solutions; this explains why **R**/**Python**/**MATLAB** usually lead (if convergence is achieved) to different estimates even if the starting values and all other settings are the same.[3] Our numerical examples from Sections 5 and 6 show that the lack of convergence is not a negligible issue in GLM deployment for both simulated and real data.

1.2. Motivation and contributions

The impact of using standard IRLS-based built-in packages on fitting not proper exponential dispersion GLMs is illustrated in the following motivational example. Specifically, using synthetic data, we compare the estimates of an Inverse Gaussian GLM based on the *log* LF, which is an example of a not proper GLM due to the non-concavity of its log-likelihood function, obtained with either **MATLAB**'s *fitglm* library or the non-linear optimisation solver provided by **MATLAB**'s *fmincon* function. Fig. 1 displays box plots of the ratio between the L_1 distance (from the true value) of the estimates obtained with the latter method and those computed using **MATLAB**'s *fitglm* values. The results suggest that the *fmincon*-based estimation significantly outperforms the *fitglm* counterpart, especially for large size problems, which indicates that IRLS is not designed to perform well for not proper GLM settings.

To summarise, for any GLM implementation, one should not only consider a proper framework but also construct bespoke algorithms to deal with the optimisation problem when possible. *Our contributions* address these fundamental issues. *First*, we provide a comprehensive characterisation of proper MLE-based GLMs for a variety of exponential dispersion models, including the Tweedie family and its well-known special cases, under various LF specifications. Our main theoretical findings indicate that most of Tweedie generalised linear models are not proper for *canonical* and *log link* functions, and identify a rich class of proper Tweedie generalised linear models with power link functions. Consequently, using the Tweedie family for GLM implementation needs a careful approach, since, despite its very flexible parametrisation, the non-standard (Tweedie) models may lead to serious computational issues. *Second*, for a few standard Tweedie GLMs equipped with special cases of *power* LFs, we introduce efficient and accurate bespoke algorithms for solving high-dimensional problems which cannot be properly tackled with standard IRLS-type methods. Specifically, we propose the *Newton's method for Self-Concordant problems (NSC)* for solving Poisson and Gamma Regressions and the *Alternating Linearisation Methods (ALM)* algorithm for Inverse Gaussian Regressions. We provide a comprehensive comparison between these algorithms and those available in the standard built-in GLM libraries from various software, such as **MATLAB** *fitglm*, **R** *glm*2 and **Python** statsmodels *sm.GLM*. We find that our methods outperform these benchmarks in terms of both accuracy and efficiency, the largest improvements being documented for high-dimensional problems. *Third*, we propose a novel interpretability methodology for *power* LFs – which are omnipresent in this paper – that is simple and mathematically sound, which could further support the adoption of such an LF that has not been used much in practice because of its lack of interpretability; for details, see Section 2.2. The flexibility of GLM modelling is definitely enhanced if the user has access to more

[3] The IRLS lack of convergence led the developers of **MATLAB** *fitglm* and **R** *glm*2 to provide bespoke solutions for this problem by adding *step-halving* arguments that are convergence enablers. For example, the earlier version of **R** *glm* was modified for this reason, and **R** *glm*2 has an enhanced *step-halving* implementation that is a step ahead, but it does not resolve the issue in its totality; for details, see Marschner (2011) or the **R** *glm*2 documentation.

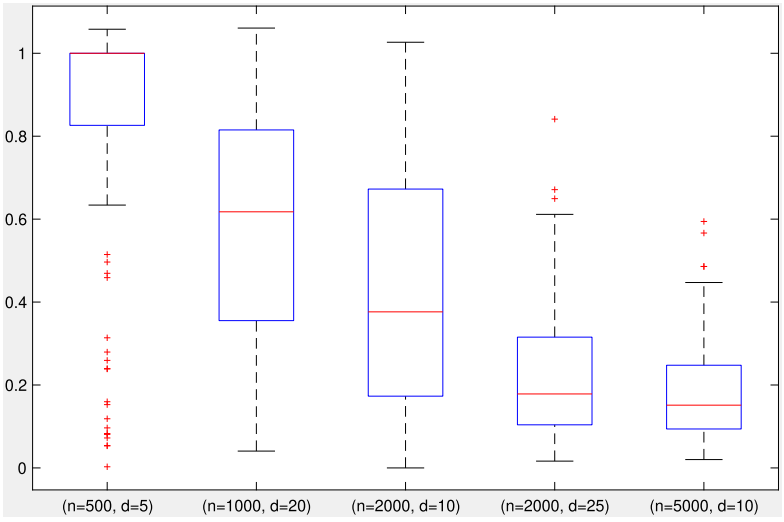

Fig. 1. Box plots of **MATLAB** *fmincon* vs *fitglm* for Inverse Gaussian GLM.
Notes: This figure shows the box plots of the ratio between the L_1 distance (from the true value) of the MLE-based GLM solutions obtained with **MATLAB**'s *fmincon* function and the IRLS-based GLM solution obtained with **MATLAB**'s *fitglm* library. Each box plot is constructed based on $N = 500$ simulations according to the DGP scheme outlined in Appendix C, for different specifications for the number of observations and the number of covariates. All GLMs are fitted with *log* LFs, i.e. a non-proper GLM.

LFs for which the pros and cons are well-understood. A similar objective is achieved in Delong et al. (2021), although the GLM model flexibility is obtained by varying the Tweedie's parameter, and one can do the same by varying the power LF parameter, so that data with complex structures could be handled by such GLM models.

The remainder of the paper is organised as follows. Section 2 introduces the notion of *proper* GLMs for exponential dispersion models and reviews the LF candidates. Section 3 provides a comprehensive classification of proper Tweedie GLMs and its subclasses. Section 4 introduces the NSC and ALM algorithms for solving Poisson and Gamma, and Inverse Gaussian Regressions, respectively. The numerical comparison between these algorithms and the standard built-in libraries from **MATLAB**, **R** and **Python** is illustrated in Section 5 for simulated data, while vast real data analysis is provided in Section 6 for two well-known insurance datasets; both sections show ample evidence for our optimisation-based implementation that is an alternative GLM implementation to the usual IRLS implementation. Section 7 concludes the paper.

2. Proper GLM and interpretability for exponential dispersion models

A univariate GLM setting assumes that the response variable Y, defined on $\mathcal{Y} \subseteq \mathfrak{R}$, is explained by covariates/features $\boldsymbol{X}$ defined on $\mathcal{X} \subseteq \mathfrak{R}^d$. Let $\{P_{\theta,\phi} : \theta \in \Theta \subseteq \mathfrak{R}, \phi \in \Phi \subseteq \mathfrak{R}\}$ be the parametric set of distributions for Y, which is assumed to be an *exponential dispersion model* characterised by the following probability density/mass function[4]:

$$\log\left(f_Y(y;\theta,\phi)\right) = \frac{\theta y - b(\theta)}{a(\phi)} + c(y,\phi). \tag{2.1}$$

Here, $a(\cdot)$, $b(\cdot)$ and $c(\cdot,\cdot)$ are real-valued functions defined on Φ, Θ and $\mathcal{Y}\times\Phi$, respectively, and ϕ is the dispersion parameter. When ϕ is fixed, (2.1) resembles an exponential family with *canonical* parameter θ. Under standard regularity conditions, the mean and variance of Y are

$$\mathrm{E}[Y] = b'(\theta) \quad \text{and} \quad \mathrm{Var}[Y] = a(\phi)b''(\theta). \tag{2.2}$$

[4] Although the univariate assumption for the response variable Y is not essential, it simplifies the exposition.

The GLM consists of n independent r.v.'s (observations) $Y_1, \cdots, Y_n$ with Y_i distributed according to (2.1) with parameters θ_i and ϕ, and functions $a_i(\phi), b(\theta_i)$ and $c(y_i,\phi)$, and conditional mean linked through a linear predictor $\eta_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}$ via a real-valued function h, so that

$$\mathrm{E}\left[Y_i \mid \boldsymbol{X}_i = \boldsymbol{x}_i\right] = h\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right). \tag{2.3}$$

Here, $\boldsymbol{x}_i$ is a d-dimensional vector of realised features/covariates for any $i = 1, \ldots, n$.[5]

The inverse function of h, provided that it exists, is known as the *link function (LF)* and it is denoted by $g = h^{-1}$. The standard GLM literature differentiates the GLMs by the parametric choice made in (2.1) and the preferred LF g. However, from the maximum likelihood estimation (MLE) perspective, the function h is more relevant than g, and thus, the remaining results are described in terms of the former. If the dispersion parameter ϕ is known (otherwise it is estimated through the variance function from (2.2)), the MLE associated with the GLM defined in Equations (2.1) and (2.3) is obtained by solving the following non-linear optimisation problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathfrak{R}^d}{\arg\max}\ \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{\theta_i y_i - b\left(\theta_i\right)}{a_i(\phi)} \ \text{ with } \ \theta_i = \left(b'^{-1} \circ h\right)\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right). \tag{2.4}$$

Without loss of generality, we let $a_i(\phi) = a(\phi)$.[6] The above optimisation problem is well-defined and admits a (unique) solution if the functions a, b and h satisfy certain regularity conditions. These constraints formalise the concept of a *proper GLM* and are summarised below.

[5] Note that although the linear predictor suggests observing d covariates/features (since $\mathcal{X} \subseteq \mathfrak{R}^d$), in fact we only assume $d-1$ covariates as we impose $x_{i,0} = 1$ for any $i = 1, \ldots, n$ almost surely. This convention simplifies the notation, so that the linear predictor becomes $\eta_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} = \beta_0 + \beta_1 x_{i,1} + \ldots + \beta_{d-1} x_{i,d-1}$.

[6] A popular choice in the GLM literature is to consider $a_i(\phi) = a(\phi)/w_i$ with $a(\phi) = \phi$ and w_i non-negative fixed weights for all $i = 1, \ldots, n$. Under this assumption, the non-linear optimisation from Equation (2.4) is equivalent to solving a weighted MLE for a GLM where the response variable follows a canonical one-parameter exponential family distribution. While this could simplify the estimation of $\boldsymbol{\beta}$, and some bespoke model adequacy is typically available to check whether the predefined weights w_i are acceptable, in reality, this is more like a trial error approach which is often resolved by relying on domain knowledge.

Definition 2.1. The GLM defined in Equations (2.1) and (2.3) is said to be *proper* if the following two conditions are satisfied:

C1. The conditional mean relationship from (2.3) is properly mapped, i.e. $h : \mathfrak{R} \to b'(\Theta) \subseteq Conv(\mathcal{Y})$ with $b' : \Theta \to b'(\Theta)$ an injective function.[7]

C2. Assume that the likelihood function is well-defined in (2.4). The individual likelihood contribution is a (strictly) concave function, i.e.

$$\begin{cases} \operatorname{sgn}(a(\phi)) \cdot \left(y \cdot \left(b'^{-1} \circ h\right)(\eta) - \left(b \circ b'^{-1} \circ h\right)(\eta)\right) \text{ is (strictly) concave} \\ \qquad\qquad \text{in } \eta \text{ on } \mathfrak{R} \text{ for any given } y \in \mathcal{Y}, \end{cases}$$

where sgn is the signum function.

Condition **C1** ensures that the GLM estimation is well-defined. More specifically, we require the function b' to be injective, so that it admits an inverse.[8] Condition **C2** implies that the likelihood function ℓ defined in (2.4) is a concave function in $\eta \in \mathfrak{R}$, since the composition of a concave function with an affine mapping is concave and the sum of concave functions is also concave; in other words, (2.4) is a concave programming instance. Consequently, under the constraints from Definition 2.1, the optimisation problem in (2.4) leads to solutions which are global maximum (see e.g. Boyd and Vandenberghe (2004)). Note that the asymptotic distribution of $\hat{\boldsymbol{\beta}}$ – like any M-estimator – requires Equation (2.4) to have a unique solution, which is not always guaranteed. However, this condition is always satisfied if the function from Condition **C2** is strictly concave. The technical conditions for the existence and uniqueness of the MLE estimate are well-known (see e.g. Wedderburn (1976) and Mäkeläinen et al. (1981)), and are standard in the literature, i.e. the log-likelihood function is strictly concave and some boundary conditions are satisfied. The MLE solutions could be on the boundary of the parameter space, which makes the estimation quite problematic, but we exclude such extreme cases from our analysis.

2.1. Link function candidates for proper GLMs

The standard choice for solving (2.4) is to assume the function h satisfies

$$h(\eta) = b'(\eta), \qquad \eta \in \mathfrak{R}. \tag{2.5}$$

Under the specification from (2.5), its equivalent LF g is known as the *canonical LF*. The sufficient conditions for a proper *canonical LF*-based GLM are summarised in the lemma below.

Lemma 2.2. *Let a GLM be equipped with its canonical LF. The MLE-based GLM is proper if $\Theta = \mathfrak{R}$ and b is strictly convex (concave) on Θ provided that $a(\phi) > 0$ ($a(\phi) < 0$) for all $\phi \in \Phi$.*

Although the *canonical* LF has useful mathematical/statistical properties, it does not always satisfy the conditions from Lemma 2.2, and therefore leads to not proper GLMs. Below, we briefly introduce two of the most popular alternative choices in the literature, namely the *log* and *power* classes of LFs.[9] The *log* LF is defined by taking

$$h(\eta) = e^{\eta}, \qquad \eta \in \mathfrak{R}. \tag{2.6}$$

Similar to the previous case, this choice may fail to produce a proper GLM in certain situations, but a general classification as in Lemma 2.2 for such models is not available. Moreover, *log* LFs have been further associated to computationally unstable MLE procedures, which leads us to considering the following family of LFs which could address some of these issues due to their appealing mathematical properties.[10] The *power* LF is defined via the following expression

$$h(\eta) = \eta^{\gamma}, \qquad \eta \in \mathfrak{R} \quad \text{and} \quad \gamma \in \mathfrak{R}^*. \tag{2.7}$$

Popular cases of *power* LFs used in numerical applications are the *identity*, *square* and *square-root* functions which are obtained by taking $\gamma = 1, 1/2$ and 2 in (2.7), respectively. Furthermore, the *reciprocal* versions of these cases (i.e. *reciprocal identity*, *reciprocal square* and *reciprocal square-root*) are obtained by letting $\gamma = -1, -1/2$ and -2, respectively.

Lemma 2.3 provides the sufficient conditions for **C1** to be satisfied under the choice from (2.7).[11]

Lemma 2.3. *Let a GLM with a power LF be chosen. Condition **C1** in Definition 2.1 is satisfied if either of the following conditions are satisfied:*

(i) *γ is a non-zero even integer and $b'(\Theta) = \mathfrak{R}_+ \subseteq Conv(\mathcal{Y})$ such that $b' : \Theta \to \mathfrak{R}_+$ is an injective mapping.*
(ii) *γ is an odd integer and $b'(\Theta) = Conv(\mathcal{Y}) = \mathfrak{R}$ such that $b' : \Theta \to \mathfrak{R}$ is an injective mapping.*

The above result helps us identify when a GLM is not proper due to Condition **C1** violation. For example, a direct consequence of Lemma 2.3 is that *power* LFs are not appropriate choices for GLMs where the function b' has a bounded image; this is the case of Logistic Regression (see Appendix B.2 for more details).

One way to tackle the not proper GLM issue for *power* LFs is to consider restrictions and/or modifications to these functions. For this purpose, we first introduce the class of *half-power* LFs which corresponds to taking

$$h(\eta) = \begin{cases} \eta^{\gamma}, & \eta > 0, \\ +\infty, & \eta \le 0, \end{cases} \tag{2.8}$$

with $\gamma \in \mathfrak{R}^*$.[12] Finally, one can consider the *negative* versions of the *power*/*half-power* functions, called *negative power*/*negative half-power*, respectively, which are obtained by multiplying h from (2.7)/(2.8) by -1.

2.2. Interpretability of GLM model outputs

Interpretability of (machine learning and statistical) model outputs is an emerging field that has been developing in recent years as it is crucial for building and ensuring trustworthy algorithms deployed in statistical and machine learning predictive modelling. Generally speaking, standard methods of global interpretation methods (e.g., the *Partial Dependence Plot* and *Accumulated Local Effect Plots*; for details, see Friedman (2001) and Apley and Zhu (2020)) make many prediction models (including GLM) more interpretable than these models were before. Such global interpretation methods describe average behaviour and become very useful when the modeller would like to understand the embedded data structure (e.g., whether linear or transformations of linear

[7] Note that *Conv* is the convex-hull of a set. In addition, $Conv(\mathcal{Y})$ should be read as $\mathcal{Y}$ when Y is continuously distributed, while the convex hull operator makes a difference when Y is a discrete random variable (see e.g. Bernoulli and Poisson families).

[8] The function b' is automatically surjective since the codomain coincides with its image $b'(\Theta)$.

[9] Note that both these functions are also *canonical* LFs for certain GLM cases. A detailed characterisation of these LFs within the context of a proper GLM is provided in Section 3 for several well-known cases of exponential dispersion models. Other classes of LFs such as *probit* and *complementary log-log* are introduced and discussed in Appendix B.3 for Logistic Regressions.

[10] Generally speaking, *power* LFs are useful for constructing convex optimisation algorithms for estimating GLMs in an accurate and efficient way. Examples of such algorithms are provided in Section 4.

[11] Note that a general characterisation for Condition **C2** cannot be provided for the *power* LF. The proofs of Lemmas 2.2 and 2.3 follow immediately from Definition 2.1.

[12] Note that the special cases for γ that we considered for the standard *power* LFs are defined in the same way for the *half-power* scenarios, e.g. we use the term *reciprocal half-square-root* for h following Equation (2.8) with $\gamma = -2$.

models are suitable for the specific dataset) or to improve the baseline model. Specifically, GLM modelling is known for its simplicity and is arguably preferred to *Generalised Additive Model (GAM)* and *Generalised Linear Mixed Models (GLMM)*, but all of them are interpretable based on the new and continuing discovery in this emergent, exciting and fast-growing interpretability of predictive (analytics) models. Therefore, such global interpretation methods apply very well to any GLM methodology, including our augmented IRLS, which could be used to interpret GLM model outputs.

As mentioned before, ensuring trustworthy algorithms is a priority to practitioners and academics with a keen interest on creating highly adoptable prediction models. Our algorithms ensure this and actually proper GLM ensures GLM trustworthiness by requiring the GLM model predictions to be as the end-user would expect to be, which is reflected in Definition 2.1 and motivated this work. It is well-known that GLM is widely used in insurance pricing and the omnipresent numerical solution is based on IRLS that is an approximation of the actual problem, i.e., maximising the data-driven MLE problem as given in (2.4). However, our proper GLM solutions enable using convex optimisation instead of using approximation methods such as IRLS; note that convex optimisation is computationally more reliable than general (non-convex) optimisation, and IRLS could lead to suboptimal solutions, and thus, proper GLM opens up a new strand of research of solving GLM in a more efficient way, and Algorithm 1 is an example of this kind. Note that the Logistic regression is a proper GLM, and no R or Python GLM solution is scalable if using IRLS, and thus, such packages use convex optimisation; the seminal paper (Boyd et al., 2011a) is an illustration of this point, while multiple implementations are available (e.g., *CVX* in **MATLAB**, *CVXPY* in **Python**, and *CVXR* in **R**, which could be retrieved from this link).[13] Thus, we may conclude that Algorithm 1 is a trustworthy algorithm.

Besides these off-the-shelf global interpretation methodologies that are mainly available and implemented in R and Python, modern bespoke models have their very own interpretability methodologies that help with enhancing the adoption of the model; for example, Zhang et al. (2023) illustrates this point for an actuarial application, namely, for their bespoke cause-of-death mortality model. In this section, we provide some simple and practical interpretable methods of GLM predictions for *power* link LFs, which are different than the global interpretation methods.

First, note that the *log* and *power* LFs defined in (2.6) and (2.7), respectively, play an essential role in our paper. The *log* LF is well-known for its interpretability, which explains why it is preferred by actuaries for insurance pricing models. Specifically, we have the following interpretation for an insured characterised by their observed or engineered $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$ covariates with a pure premium computed via the *log* LF and model estimates $\hat{\boldsymbol{\beta}}$

$$\frac{\partial e^{\eta}}{\partial x_k} = \frac{\partial e^{\boldsymbol{x}^\top \boldsymbol{\beta}}}{\partial x_k}\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} = \hat{\beta}_k \quad \text{for all } 2 \le k \le d, \tag{2.9}$$

which is a very neat interpretation of this insurance pricing model that it also has an obvious sensitivity interpretation connotation. Note that $k \neq 1$ in (2.9) since $x_1 = 1$ always holds in the GLM model and thus, the intercept cannot be interpreted in this way, but practitioners adjust the intercept so that the observed expected claim amount does not deviate too much from the average GLM predicted claim amount.

Following a similar approach, we now provide a novel interpretability for *power* LFs. As before, we assume an insured characterised by their observed or engineered $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$ covariates with a pure premium computed via the *power* LF and model estimates $\boldsymbol{\beta}$, i.e., the pure premium estimate is $\hat{y} = \left(\boldsymbol{x}^\top \hat{\boldsymbol{\beta}}\right)^{\gamma}$. We then have the following interpretation for such pure premium computed via a generic *power* LF

$$\begin{aligned} \hat{y} &= \sum_{k=1}^{d} \frac{\beta_k}{\gamma} \frac{\partial \hat{y}}{\partial \beta_k}\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} := \sum_{k=1}^{d} PC_k(\hat{\boldsymbol{\beta}}; \boldsymbol{x}), \\ PC_k(\boldsymbol{\beta}; \boldsymbol{x}) &:= \frac{\beta_k}{\gamma} \frac{\partial \hat{y}}{\partial \beta_k}\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} \text{ for all } 1 \le k \le d. \end{aligned} \tag{2.10}$$

The validity of (2.10) is ensured by the Euler's Homogeneous Function Theorem and similar arguments have been widely used in capital allocation (Denault, 2001; Asimit et al., 2019) and in portfolio theory (Asimit et al., 2024; Cetingoz et al., 2024). The interpretation of PC_k is very natural and represents the pure premium contribution of the k^{th} covariate to the pure premium $\hat{y}$. This interpretation is more general than the one used in (2.9), since the latter does not hold for a non-continuous covariate such as categorical/nominal covariate or ordinal covariate (e.g., on a Likert scale). On the contrary, our interpretability model in (2.10) does not exhibit such a drawback though the formulation in (2.10) would need a slight modification. That is, if the k^{th} observed covariate requires one-hot-encoding, then the PC corresponding to the original observed covariate (e.g., the k^{th} one) should be replaced by the sum of PC's corresponding to the one-hot-encoded covariates; that is, if the original covariate is an ordinal covariate with three possible outcomes (e.g., "low", "medium" and "high"), then the contribution of the k^{th} covariate to the pure premium is the sum of the three PCs corresponding to the one-hot-encoded covariates (corresponding to "low", "medium" and "high"). According to our knowledge, the novel interpretability of *power* LFs in (2.10) has not been discussed in the literature, and we believe that the practitioners would appreciate such very simple and natural interpretation of these GLM outputs. It might be more meaningful to report the percentage contribution vector as follows

$$\left(\frac{PC_1(\hat{\boldsymbol{\beta}}; \boldsymbol{x})}{\hat{y}}, \ldots, \frac{PC_d(\hat{\boldsymbol{\beta}}; \boldsymbol{x})}{\hat{y}}\right)^\top, \tag{2.11}$$

although a word of caution is that some of these percentages may be negative even though they sum up to 100%. An application of this interpretability is illustrated in our real data analyses from Section 6, and in particular in Section 6.1 for the health insurance dataset.

3. Special examples of GLMs and main results

This section provides a classification of proper MLE-based GLM for a variety of exponential dispersion models and discusses the potential issues associated with the use of the different LFs introduced in Section 2. Specifically, we focus on the more general Tweedie family, together with three of its most popular special cases, namely the Poisson, Gamma and Inverse Gaussian distributions.[14] A summary of proper GLMs is provided at the end of the section.

3.1. Poisson regression – Poisson family

We assume $Y \sim Poisson(\theta)$ with probability mass function given by

$$\log\left(f_Y(y; \theta, \phi)\right) = \theta y - e^{\theta} - \log(y!), \quad (y, \theta, \phi) \in \mathbb{N} \times \mathfrak{R} \times \{1\}.$$

The above expression is obtained as a special case of (2.1) by taking

$$a(\phi) = \phi = 1, \quad b(\theta) = e^{\theta}, \quad c(y, \phi) = -\log(y!).$$

In addition, $b'(\Theta) = \mathfrak{R}_+^*$ and $b'^{-1}(\mu) = \log(\mu)$. Proposition 3.1 provides a characterisation of a proper Poisson Regression model according to our Definition 2.1.

Proposition 3.1. *Assume that $Y \sim Poisson(\theta)$. The Poisson GLM is proper if and only if $h : \mathfrak{R} \to \mathfrak{R}_+^*$, and*

[13] Available at: https://stanford.edu/~boyd/software.html.

[14] In addition, the Linear and Logistic Regression models are also illustrated in Section Appendix B though we mention that only the Linear Regression is a special case of the Tweedie Regression.

$$-y\log(h(\eta))+h(\eta) \quad \textit{is convex in } \eta \textit{ on } \mathfrak{R} \textit{ for any given } y\in\mathbb{N}. \tag{3.1}$$

The Poisson *canonical* LF is the *log* function and this choice leads to a proper GLM due to either Lemma 2.2 or Proposition 3.1. The *power* LF does not satisfy the conditions from Proposition 3.1 unless $\gamma=2k$ with $k\in\mathbb{N}^*$; specifically, Condition **C1** does not hold unless γ is a non-zero even integer, while Condition **C2** requires $\gamma\geq 1$. The *half-power* LF satisfies the conditions stated in Proposition 3.1 for any $\gamma\in[1,\infty)$. Thus, the simplified Poisson regression (i.e. $\phi=1$) with a proper *half-power* LF, obtained by taking any $\gamma\geq 1$ in (2.8), leads to solving

$$\hat{\boldsymbol{\beta}}=\underset{\boldsymbol{\beta}\in\mathfrak{R}^d}{\arg\max}\ \ell(\boldsymbol{\beta})=\sum_{i=1}^{n}\left(\gamma\, y_i\log\left(\boldsymbol{x}_i^\top\boldsymbol{\beta}\right)-\left(\boldsymbol{x}_i^\top\boldsymbol{\beta}\right)^{\gamma}\right). \tag{3.2}$$

While these *half-power* LFs lead to proper GLMs that could be solved via a general convex programming algorithm, the *half-identity* and *half-square-root* cases can be solved via a computationally efficient algorithm, as outlined in Section 4.1. Finally, note that the *half-square-root* and the standard *square-root* LFs are closely related, but the latter does not satisfy (3.1) because Condition **C2** does not hold in this case. Essentially, the *half-square-root* case optimises the strictly concave instance in (3.2) on the $\mathfrak{R}^d$ cone such that $\boldsymbol{x}_i^\top\boldsymbol{\beta}>0$ for all $i=1,\dots,n$, while the *square-root* solves a similar problem to (3.2) (where $\log\left(\boldsymbol{x}_i^\top\boldsymbol{\beta}\right)$ is replaced by $\log\left|\boldsymbol{x}_i^\top\boldsymbol{\beta}\right|$) on $\mathfrak{R}^d$, but its objective function is not concave on the entire feasibility set, namely $\mathfrak{R}^d$. An analogous differentiation between the *half-identity* and *identity* LFs can be formulated as well. Finally, Condition **C1** is not satisfied for any *negative power* LF or *negative half-power* LF, which are not proper for Poisson GLM.

3.2. Gamma Regression – Gamma family

We assume $Y\sim Gamma(\theta,\phi)$ with probability distribution function given by

$$\log\left(f_Y(y;\theta,\phi)\right)=\frac{\theta y+\log(-\theta)}{\phi}+\frac{1-\phi}{\phi}\log(y)-\log\left(\phi^{\frac{1}{\phi}}\Gamma\left(\frac{1}{\phi}\right)\right),\ (y,\theta,\phi)\in\mathfrak{R}_+^*\times\mathfrak{R}_-^*\times\mathfrak{R}_+^*.$$

The above expression is obtained as a special case of (2.1) by taking

$$a(\phi)=\phi,\quad b(\theta)=-\log(-\theta),$$
$$c(y,\phi)=\frac{1-\phi}{\phi}\log(y)-\log\left(\phi^{\frac{1}{\phi}}\Gamma\left(\frac{1}{\phi}\right)\right).$$

In addition, $b'(\Theta)=\mathfrak{R}_+^*$ and $b'^{-1}(\mu)=-\mu^{-1}$. Proposition 3.2 provides a characterisation of a proper Gamma Regression model according to our Definition 2.1.

Proposition 3.2. *Assume that $Y\sim Gamma(\theta,\phi)$. The Gamma GLM is proper if and only if $h:\mathfrak{R}\to\mathfrak{R}_+^*$, and*

$$\frac{y}{h(\eta)}+\log(h(\eta)) \quad \textit{is convex in } \eta \textit{ on } \mathfrak{R} \textit{ for any given } y\in\mathfrak{R}_+^*. \tag{3.3}$$

The *canonical* LF associated with the Gamma GLM is the *reciprocal identity* function. This function does not satisfy the conditions stated in Lemma 2.2 or Proposition 3.2, since Condition **C1** does not hold, and therefore, unlike in the Poisson case, the *canonical* GLM is not proper. A popular alternative for Gamma GLM is represented by the *log* LF; this choice satisfies the conditions stated in Proposition 3.2 and is thus appropriate for Gamma GLM. As in Section 3.1, we now discuss the impact of using *power*/*half-power* LFs in Gamma GLM. First, a *power* LF does not satisfy the conditions from Proposition 3.2 unless $\gamma=-2k$, with $k\in\mathbb{N}^*$; specifically, Condition **C1** does not hold unless γ is a non-zero even integer, while Condition **C2** requires $\gamma\leq -1$. Second, one could find that *half-power* LFs always satisfy Condition **C1**, but Condition **C2** holds if and only if $\gamma\leq -1$, leading to proper Gamma GLM in this case. Note that the simplified Gamma GLM (i.e. $\phi=1$) with such proper *half-power* LF is equivalent to solving

$$\hat{\boldsymbol{\beta}}=\underset{\boldsymbol{\beta}\in\mathfrak{R}^d}{\arg\max}\ \ell(\boldsymbol{\beta})=\sum_{i=1}^{n}\left(-\gamma\log\left(\boldsymbol{x}_i^\top\boldsymbol{\beta}\right)-y_i\left(\boldsymbol{x}_i^\top\boldsymbol{\beta}\right)^{-\gamma}\right), \tag{3.4}$$

where $\gamma\leq -1$. While *half-power* LFs with $\gamma\leq -1$ lead to proper GLMs that could be solved via a general convex programming algorithm, the *half-reciprocal identity* and *half-reciprocal-square-root* cases could be solved via a computationally efficient algorithm, as outlined in Section 4.1. Finally, Condition **C1** is not satisfied for any *negative power* LF or *negative half-power* LF, which are not proper for Poisson GLM.

3.3. Inverse Gaussian Regression – Inverse Gaussian (IG) family

We assume $Y\sim IG(\theta,\phi)$ with probability distribution function given by

$$\log\left(f_Y(y;\theta,\phi)\right)=\frac{\theta y-\sqrt{2\theta}}{-1/\phi}+\frac{1}{2}\left(\log\left(\frac{\phi}{2\pi y^3}\right)-\frac{\phi}{y}\right),$$
$$(y,\theta,\phi)\in\mathfrak{R}_+^*\times\mathfrak{R}_+^*\times\mathfrak{R}_+^*.$$

The above function is also a special case of (2.1) where

$$a(\phi)=-\frac{1}{\phi},\quad b(\theta)=\sqrt{2\theta},\quad c(y,\phi)=\frac{1}{2}\left(\log\left(\frac{\phi}{2\pi y^3}\right)-\frac{\phi}{y}\right).$$

In addition, $b'(\Theta)=\mathfrak{R}_+^*$ and $b'^{-1}(\mu)=\frac{1}{2}\mu^{-2}$. Proposition 3.3 provides the characterisation of a proper Inverse Gaussian Regression model according to our Definition 2.1.

Proposition 3.3. *Assume that $Y\sim IG(\theta,\phi)$. The Inverse Gaussian GLM is proper if and only if $h:\mathfrak{R}\to\mathfrak{R}_+^*$, and*

$$\frac{y}{2h^2(\eta)}-\frac{1}{h(\eta)} \quad \textit{is convex in } \eta \textit{ on } \mathfrak{R} \textit{ for any given } y\in\mathfrak{R}_+^* \tag{3.5}$$

The *canonical* LF for the GLM-based on the IG distribution is the *reciprocal square* function. Similar to the Gamma scenario, this function does not satisfy the conditions stated in Lemma 2.2 or Proposition 3.3, namely Condition **C1**, and therefore, it is not a proper GLM. Under the *log* LF assumption Condition **C1** is satisfied, but Condition **C2** is violated since (3.5) does not hold. The effect of non-convexity is depicted in our motivational example from Fig. 1.

As before, we also investigate the *power* and *half-power* LFs in the context of an IG GLM. *First*, we notice that there is no *power* LF that satisfies the conditions in Proposition 3.3; specifically, Condition **C1** does not hold unless γ is a non-zero even integer, while Condition **C2** is satisfied if and only if $\gamma\in[-1,-1/2]$. *Second*, one could find that *half-power* LFs always satisfy Condition **C1**, but Condition **C2** holds if and only if $\gamma\in[-1,-1/2]$, concluding that *half-power* LF leads to a proper GLM only in this case. Given the previous findings, running IG Regressions with *power* or *half-power* LFs would require a compromise. That is, the *power* LF with $\gamma=2k$, $k\in\mathbb{Z}^*$ is the best possible choice so that constrained programming is avoided (for proper IG GLM with *half-power* LFs such that $\gamma\in[-1,-1/2]$ for which n linear inequality constraints are needed), which is computationally undesirable for large samples. Such choice requires an efficient algorithm to solve the non-concave log-likelihood function optimisation. We show how to achieve this in Section 4.2 for the *reciprocal-square-root* LF.

3.4. Main results on Tweedie regression – Tweedie family

In this section, we focus our analysis on a more general class of GLMs based on the Tweedie family, which includes the previous distributions as special/limiting cases. As before, our main goal is to in-

vestigate if the Tweedie distribution leads to proper GLMs. Assume that $Y \sim Tweedie(\theta, \phi)$ with probability distribution function defined below

$$\log\left(f_Y(y;\theta,\phi)\right) = \frac{\theta y - K_p(\theta)}{\phi} + \log\left(\mu'_\phi((-\infty, y])\right),$$
$$(y,\theta,\phi) \in \mathcal{Y} \times \Theta \times \mathfrak{R}^*_+, \tag{3.6}$$

where $\Theta \subseteq \mathfrak{R}$, μ_ϕ is a Radon measure on $\mathcal{Y} \subseteq \mathfrak{R}$ and the function K_p is given by

$$K_p(\theta) := \begin{cases} \frac{\alpha-1}{\alpha}\left(\frac{\theta}{\alpha-1}\right)^\alpha, & p \in (-\infty, 0] \cup (1,\infty) \setminus \{2\}, \\ e^\theta, & p = 1, \\ -\log(-\theta), & p = 2, \end{cases}$$

with $\alpha = \frac{p-2}{p-1}$. The expression from (3.6) is obtained as a special case of (2.1) by taking

$$a(\phi) = \phi, \quad b(\theta) = K_p(\theta), \quad c(y,\phi) = \log\left(\mu'_\phi((-\infty, y])\right).$$

Moreover, the Poisson, Gamma and Inverse Gaussian families are obtained as special cases by taking $p = 1$ with $\mathcal{Y} = \mathbb{N}$ and $\Theta = \mathfrak{R}$, $p = 2$ with $\mathcal{Y} = \mathfrak{R}^*_+$ and $\Theta = \mathfrak{R}^*_-$, and $p = 3$ with $\mathcal{Y} = \mathfrak{R}^*_+$ and $\Theta = \mathfrak{R}^*_-$, respectively.[15]

Without loss of generality, we henceforth assume that $p \neq \{1, 2\}$, since these two cases have already been investigated in Sections 3.1 and 3.2. Note that one should carefully choose $\Theta, \mathcal{Y}$ and p so that $K_p(\cdot)$ is well-defined on Θ. In this section, we assume that $\Theta \in \{\mathfrak{R}, \mathfrak{R}^*, \mathfrak{R}^*_+, \mathfrak{R}^*_-\}$, and thus, the function b' is well-defined and bijective on Θ only under the three settings considered in the theorem below. Extensions to subsets of these sets are obtainable at the expense of the exposition, and for this reason, we proceed with this simplification.

We now provide a characterisation of proper Tweedie GLMs, where we exclude the previous cases investigated in Sections 3.1 and 3.2 and Appendix B.1. First, we identify in Theorem 3.4 all possible settings under which Condition **C1** from Definition 2.1 is satisfied.

Theorem 3.4. *Let $Y \sim Tweedie(\theta, \phi)$ parameterised as in (3.6) with $p \in (-\infty, 0) \cup (1, 2) \cup (2, \infty)$ (or equivalently, $\alpha \in (-\infty, 2) \setminus \{0, 1\}$) such that $\mathcal{Y}, \Theta \in \{\mathfrak{R}, \mathfrak{R}^*, \mathfrak{R}^*_+, \mathfrak{R}^*_-\}$. Then, Condition **C1** is only satisfied for the following settings:*

a) $\Theta = b'(\Theta) = \mathfrak{R}^*_+$ *(or $\mathfrak{R}_+$)*, $\mathcal{Y} \in \{\mathfrak{R}^*_+, \mathfrak{R}\}$ *(or $\mathcal{Y} \in \{\mathfrak{R}_+, \mathfrak{R}\}$) and $1 < \alpha < 2$ (which is equivalent to $p < 0$), with $h : \mathfrak{R} \to \mathfrak{R}^*_+$ (or $h : \mathfrak{R} \to \mathfrak{R}_+$);*
b) $\Theta = \mathfrak{R}^*_-$, $b'(\Theta) = \mathfrak{R}^*_+$, $\mathcal{Y} \in \{\mathfrak{R}^*_+, \mathfrak{R}_+, \mathfrak{R}\}$ *and $\alpha \in (-\infty, 1) \setminus \{0\}$ (which is equivalent to $p \in (1, \infty) \setminus \{2\}$), with $h : \mathfrak{R} \to \mathfrak{R}^*_+$;*
c) $\Theta = \mathfrak{R}$, $b'(\Theta) = \mathfrak{R}^*_+$, $\mathcal{Y} \in \{\mathfrak{R}^*_+, \mathfrak{R}_+, \mathfrak{R}^*\}$, $\alpha \in \{-2l + 1 : l \in \mathbb{N}^*\}$, *with $h : \mathfrak{R} \to \mathfrak{R}^*_+$.*
d) $\Theta = \mathfrak{R}$, $b'(\Theta) = \mathfrak{R}^*$, $\mathcal{Y} \in \{\mathfrak{R}^*, \mathfrak{R}\}$, $\alpha \in \{-2l : l \in \mathbb{N}^*\}$, *with $h : \mathfrak{R} \to \mathfrak{R}^*$.*

Setting a) includes a pedantic reference on whether the response variable could or could not include $y = 0$, and thus, we made a difference between the cases $\Theta = \mathfrak{R}^*_+$ and $\Theta = \mathfrak{R}_+$. Note that the generic Condition **C1** in Definition 2.1 requires the range of $\mathrm{E}[Y]$, namely $b'(\Theta)$, to be a subset of $Conv(\mathcal{Y})$, though a more practical condition would be $b'(\Theta) = Conv(\mathcal{Y})$, which we assume henceforth. Setting c) is a subcase of setting b) from the implementation point of view, since the modeller chooses the Tweedie models so that $\mathcal{Y}$ matches the data range of values. However, our classification in Theorem 3.4 has to differentiate between models with different parameter sets Θ. The next results focus on the validity of Condition **C2** from Definition 2.1 for the above Tweedie GLM settings under the LF specifications introduced in Section 2. The *power* LF class, together with its restrictions/modifications, is investigated in Theorem 3.5 below.

Theorem 3.5. *Let $Y \sim Tweedie(\theta, \phi)$ parameterised as in (3.6) with $b'(\Theta) = \mathcal{Y}$, for which condition **C1** is satisfied. Then, Condition **C2** is not satisfied by settings a)–d), for any*

(i) *power LF, except for the following cases:*
  – *setting b) with $0 < \alpha < 1$ and $\gamma = -2k$, for any $k \in \mathbb{N}^*$, with $(1-\gamma)\alpha \leq 1$,*
  – *setting b) with $\alpha < 0$ and $\gamma = 2k$, for any $k \in \mathbb{Z}^*$,*
  – *setting c) and $\gamma = 2k$, for any $k \in \mathbb{Z}^*$.*
(ii) *half-power LF, except for the following cases:*
  – *setting a) with $1 < \alpha < 2$ and $\frac{\alpha-1}{\alpha} \leq \gamma \leq \alpha - 1$,*
  – *setting b) with $0 < \alpha < 1$ and $\frac{\alpha-1}{\alpha} \leq \gamma \leq \alpha - 1$,*
  – *setting b) with $\alpha < 0$ and $\gamma \leq \alpha - 1$ or $\frac{\alpha-1}{\alpha} \leq \gamma$,*
  – *setting c) with $\alpha \in \{-2l + 1 : l \in \mathbb{N}^*\}$ and $\gamma \leq \alpha - 1$ or $\frac{\alpha-1}{\alpha} \leq \gamma$.*
(iii) *negative power or negative half-power LF.*

We notice that the above results are in agreement with our previous findings. For example, one could recover our discussion from Section 3.3 on proper IG GLMs, which is a special case of Theorem 3.5 if we take $p = 3$ (or equivalently $\alpha = 1/2$), where we found that proper IG GLMs with *half-power* LF are achieved if and only if $\gamma \in [-1, -1/2]$. In addition, Theorem 3.5 provides necessary and sufficient conditions for proper GLMs under other distributional assumptions. For example, Tweedie GLMs based on Positive stable distributions (i.e. $p > 2$ or equivalently $0 < \alpha < 1$) are proper only for *power LFs* with $\gamma = -2k$, $k \in \mathbb{N}^*$, with $(1-\gamma)\alpha \leq 1$ and *half-power LFs* with $\frac{\alpha-1}{\alpha} \leq \gamma \leq \alpha - 1$. Similarly, the Compound Poisson-Gamma GLM (i.e. $1 < p < 2$ or equivalently $\alpha < 0$) is proper only for *power* LFs with $\gamma = 2k$, $k \in \mathbb{Z}^*$ or *half-power* LFs with $\gamma \leq \alpha - 1$ or $\frac{\alpha-1}{\alpha} \leq \gamma$. A complete summary of proper Tweedie GLMs is illustrated in Table 1 of Section 3.5.

Note that if $p \in (-\infty, 0] \cup (1, \infty) \setminus \{2\}$, which is equivalent to $\alpha \in (-\infty, 2] \setminus \{0, 1\}$, then the simplified Tweedie regression (i.e. $\phi = 1$) with LF h is equivalent to solving

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathfrak{R}^d}{\arg\max}\ \ell(\boldsymbol{\beta})$$
$$= \sum_{i=1}^n \left(y_i(\alpha-1)\left(h\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)\right)^{\frac{1}{\alpha-1}} - \frac{\alpha-1}{\alpha}\left(h\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)\right)^{\frac{\alpha}{\alpha-1}} \right). \tag{3.7}$$

A few comments on (3.7) would help understanding the issues with deploying Tweedie GLMs. *First*, one may discard Condition **C2** at the expense of losing all useful properties of the M-estimators (MLE is only a special case), such as the asymptotic distribution, which questions the asymptotic bias and variance of these estimators. If that is the case, one can only hope for the numerical optimisation to behave well, but this is possible from case to case, and one would need to perform extensive numerical implementations to check whether the optimisation algorithm shows a reasonable performance for specific choices of $(\alpha, \mathcal{Y}, h)$. Such compromise is done in Algorithm 2 for solving (4.5), where $\alpha = \frac{1}{2}$ as $p = 3$, $\mathcal{Y} = \mathfrak{R}_+$, and *reciprocal square-root* LF; one could recover (4.5) from (3.7) for this particular choice of $(\alpha, \mathcal{Y}, h)$. *Second*, there are other parametrisations other than the one in Algorithm 2 for which Condition **C2** is not satisfied while all other regularity conditions in Definition 2.1 hold. In these instances, one has to rely on non-convex optimisation, but more importantly, one has to accept that some (possibly all) statistical properties of the MLE estimator may not hold. The modeller needs to identify stable computational methods (as in Algorithm 2) instead of assuming that the general purpose GLM solvers are indeed computationally stable. Finally, we notice that the proper GLMs identified in Theorem 3.5 (ii) require solving a constrained optimisation

[15] Other notable examples are Gaussian ($p = 0$ with $\mathcal{Y} = \Theta = \mathfrak{R}$), Compound Poisson-Gamma ($1 < p < 2$ with $\mathcal{Y} = \Theta = \mathfrak{R}_+$) and Positive stable ($p > 2$ with $\mathcal{Y} = \Theta = \mathfrak{R}_+$).

Table 1
Summary of proper GLMs and violations of Conditions **C1** and **C2**.

Regression model	LF	Predictor $\left(\hat{y}=h\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)\right)$	Violations
Gaussian/Linear	*identity* (canonical)	$\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}$	No
Logistic	*logit* (canonical)	$\left(1+\exp\left(-\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)\right)^{-1}$	No
	probit	$\Phi\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)$	No
	complementary log-log	$1-\exp\left(-\exp\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)\right)$	No
Poisson	*log* (canonical)	$\exp\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)$	No
	power	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}$	No, if $\gamma=2k,\ k\in\mathbb{N}^{*}$
	half-power	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}\cdot I_{\{\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}>0\}}$	No, if $\gamma\geq 1$
Gamma	*reciprocal identity* (canonical)	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{-1}$	**C1**
	log	$\exp\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)$	No
	power	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}$	No, if $\gamma=-2k,\ k\in\mathbb{N}^{*}$
	half-power	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}\cdot I_{\{\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}>0\}}$	No, if $\gamma\leq -1$
Inverse Gaussian	*reciprocal square* (canonical)	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{-1/2}$	**C1**
	log	$\exp\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)$	**C2**
	power	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}$	**C1**, if $\gamma\neq 2k,\ k\in\mathbb{Z}^{*}$, and **C2**, if $\gamma\notin[-1,-1/2]$
	half-power	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}\cdot I_{\{\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}>0\}}$	No, if $\gamma\in[-1,-1/2]$
Tweedie (except of some of the above special cases: Gaussian, Poisson and Gamma)	*canonical*	$\left((1-p)\cdot\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{1/(1-p)}$	see Theorem 3.6
	log	$\exp\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)$	see Theorem 3.6
	power or *negative power*	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}$ or $-\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}$, $\gamma\in\Re^{*}$	see Theorem 3.5
	half-power	$\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}\cdot I_{\{\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}>0\}}$, $\gamma\in\Re^{*}$	see Theorem 3.5
	negative half-power	$-\left(\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}\right)^{\gamma}\cdot I_{\{\boldsymbol{x}^{\top}\hat{\boldsymbol{\beta}}>0\}}$, $\gamma\in\Re^{*}$	see Theorem 3.5

Notes: This table presents a summary of proper GLMs equipped with the LFs discussed in Section 3 and Appendix B, and the potential violations of Conditions **C1** and **C2** from Definition 2.1 associated with these regressions. Φ stands for the $\mathcal{N}(0,1)$ cumulative distribution functionand I_A represents the indicator function for set A.

problem on the convex cone $\left\{\boldsymbol{\beta}\in\Re^{d}:\boldsymbol{x}_i^{\top}\boldsymbol{\beta}\geq 0, i=1,\ldots,n\right\}$. Unfortunately, this is computationally expensive for large values of n, which is a negative attribute. These optimisations could be solved via convex programming and not via off-the-shelf GLM packages that rely on IRLS which cannot be adapted when such constraints are needed.

The classification of proper Tweedie GLMs based on *canonical* and *log* LF is illustrated below.

Theorem 3.6. *Let $Y\sim Tweedie(\theta,\phi)$ parameterised as in (3.6) with $b'(\Theta)=\mathcal{Y}$, for which condition **C1** is satisfied. Then, Condition **C2** is not satisfied by settings a)–d), for any*

(i) *canonical LF.*
(ii) *log LF, except for setting b) with $\alpha<0$ or setting c).*

Theorem 3.6 shows that there are no proper Tweedie GLMs if the *canonical* LF is chosen. In addition, we notice that the Compound Poisson-Gamma GLM is proper for any *log* LF.

3.5. Summary results

Table 1 summarises our findings discussed in Section 3 and Appendix B. *First*, we recall that the *canonical* LFs, which are the standard choices in all built-in GLM implementations (available in **MATLAB**, **Python**, **R**, etc.), lead to not proper Tweedie GLMs, except for the Gaussian and Poisson cases. *Second*, *log* LFs tend to have the similar limitations to *canonical* LFs for Tweedie modelling. *Third*, the *power* and *half-power* LFs allow more flexibility than *log* LFs to GLM modelling when proper GLM are sought.

4. Alternative algorithms for GLMs with power LFs

The goal of this section is to not only provide efficient methods for solving high-dimensional problems while addressing the potential numerical issues in the optimisation stage, but to also create tractable models for dealing with non-convex instances, which cannot be tackled with standard built-in GLM algorithms. In this sense, we introduce the *Newton's method for Self-Concordant problems (NSC)* for Poisson and Gamma regressions equipped with some bespoke *half-power* LFs, and the *Alternating Linearisation Method (ALM)* for solving Inverse Gaussian regressions based on the *reciprocal-square-root* LF.[16]

4.1. The NSC algorithm for Poisson and Gamma Regressions

The explicit structure of such *self-concordant* functions allows for defining a refined Newton's method which is generally more efficient due to a reduced number of iterations.[17] First, we introduce the definition of a *self-concordant* function, which was first provided by Nesterov

[16] This is also known as *inverse-square-root* LF, but we avoid referring to 'inverse' since the GLM uses the inverse of a function to identify the functional estimator h with the LF g.

[17] For further details on SC problems and their fast convergence iterative methods, see Boyd and Vandenberghe (2004); Nesterov (2004).

(2004), although a simplified version is provided in Boyd and Vandenberghe (2004), which we follow in this paper.

Definition 4.1. Let $f : \Omega \to \Re$ be a closed convex function[18] where $\Omega = dom(f)$ is an open set in $\Re^d$ and $f \in C^3(dom(f))$. The function f is self-concordant on Ω if the function $g(t) := f(\boldsymbol{u} + t\boldsymbol{v})$ satisfies $|g'''(t)| \leq 2\left(g''(t)\right)^{3/2}$ for any $t \in dom(g) \subseteq \Re$, $\boldsymbol{u} \in dom(f)$, and $\boldsymbol{v} \in \Re^d$ such that $\boldsymbol{u} + t\boldsymbol{v} \in dom(f)$.

Note that the constant 2 in Definition 4.1, see $|g'''(t)| \leq 2\left(g''(t)\right)^{3/2}$, is chosen for convenience and helps to identify an explicit upper bound for the total number of iterations required by the Newton's method for SC functions. If constant 2 is replaced by M, i.e. $|g'''(t)| \leq M\left(g''(t)\right)^{3/2}$, then we say that its equivalent function f is SC with constant M; e.g., if f is SC with constant M, then it is not difficult to show that $\tilde{f}(\cdot) := \frac{M^2}{4} f(\cdot)$ is SC with constant 2.

We explore the Poisson and Gamma Regressions based on some special choices of *half-power* LFs by solving (3.2) and (3.4), since the associated negative log-likelihoods are not only convex (actually strictly convex in those two cases), but also *self-concordant*. This is illustrated in Theorem 4.2 below, where the *half-identity* and *half-square-root* LFs for Poisson Regression are explored in Theorem 4.2 a), while the *half-reciprocal identity* and *half-reciprocal-square-root* LFs for Gamma Regression are explored in Theorem 4.2 b).

Theorem 4.2. *Let* $\{(y_i, \boldsymbol{x}_i) : 1 \leq i \leq n\}$ *be a sample of size* n *drawn from* $(Y, \boldsymbol{X})$*, where* $\boldsymbol{X} = (X_1, X_2, \ldots, X_d)$ *with* $d \geq 1$ *and define* $\Omega := \bigcup_{i=1}^{n} \{\boldsymbol{\beta} \in \Re^d : \boldsymbol{x}_i^\top \boldsymbol{\beta} > 0\}$*. The following statements hold:*

a) *The MLE-based Poisson GLM equipped with the half-power LF from (2.8) with either* $\gamma = 2$ *(and* $\gamma = 1$*) is self-concordant, and it leads to an optimisation problem with a self-concordant objective function* f_P *(*$\check{f}_P$*) on* Ω*, where*

$$\min_{\boldsymbol{\beta} \in \Omega} f_P(\boldsymbol{\beta}) := \sum_{i=1}^{n} \left(\frac{1}{2} \left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2 - y_i \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right) \right), \tag{4.1}$$

$$\min_{\boldsymbol{\beta} \in \Omega} \check{f}_P(\boldsymbol{\beta}) := \sum_{i=1}^{n} \left(\boldsymbol{x}_i^\top \boldsymbol{\beta} - y_i \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right) \right). \tag{4.2}$$

b) *The MLE-based Gamma GLM equipped with the half-power LF from (2.8) with* $\gamma = -2$ *(and* $\gamma = -1$*) is self-concordant, and it leads to an optimisation problem with a self-concordant objective function* f_G *(*$\check{f}_G$*) on* Ω*, where*

$$\min_{\boldsymbol{\beta} \in \Omega} f_G(\boldsymbol{\beta}) := \sum_{i=1}^{n} \left(\frac{y_i}{2} \left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2 - \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right) \right), \tag{4.3}$$

$$\min_{\boldsymbol{\beta} \in \Omega} \check{f}_G(\boldsymbol{\beta}) := \sum_{i=1}^{n} \left(y_i \cdot \boldsymbol{x}_i^\top \boldsymbol{\beta} - \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right) \right). \tag{4.4}$$

As previously mentioned, the constant of an SC function does not have any impact on the actual iterative algorithm, and could change only the upper bound of the total number of steps (that is in an explicit form for SC functions; for details, see the Newton's step in Algorithm 1). One may show that a tighter bound could be obtained for (4.1) and (4.2), i.e. the objective function is SC with constant M_P and $\check{M}_P$, respectively, where

$$M_P = \check{M}_P := 2 \max_{1 \leq i \leq n} \left\{ y_i^{-1/2} I_{\{y_i > 0\}} + I_{\{y_i = 0\}} \right\},$$

which satisfies $M_P \leq 2$. However, no tighter bound (tighter than 2) is possible for the Gamma GLMs in either (4.3) and (4.4).

Theorem 4.2 allows us to use the standard SC algorithm which is detailed in (Nesterov, 2004; Boyd and Vandenberghe, 2004), and is provided here as Algorithm 1.

Algorithm 1: Standard SC algorithm for solving (4.1) and (4.3).

Result: $\boldsymbol{z}^{(k^*)}$ which approximates $\boldsymbol{z}^*$, the global optimum of $\min_{\boldsymbol{z} \in \Omega} f(\boldsymbol{z})$ with $f(\cdot)$ being SC on Ω, where k^* is the termination step.

Choose $\boldsymbol{z}^{(0)} \in dom(f)$, $\epsilon > 0$, and $\lambda^* \in (0, \tilde{\lambda})$ where $\tilde{\lambda} = \frac{3-\sqrt{5}}{2}$;
Let $\nabla f(\cdot)$ and $\nabla^2 f(\cdot)$ be the gradient and Hessian, respectively, of f on Ω;
Define the *step/search direction* function $\Delta(\cdot) := \left[\nabla^2 f(\cdot)\right]^{-1} \nabla f(\cdot)$ on Ω;
Define $\lambda_f(\cdot) := \left(\nabla f(\cdot)^\top \left[\nabla^2 f(\cdot)\right]^{-1} \nabla f(\cdot)\right)^{1/2}$ on Ω;

Step 1: Damped phase
(i) If $\lambda_f\left(\boldsymbol{z}^{(0)}\right) < \lambda^*$ go to Step 2;
(ii) While $\lambda_f\left(\boldsymbol{z}^{(k)}\right) \geq \lambda^*$ do $\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} - \frac{1}{1 + \lambda_f\left(\boldsymbol{z}^{(k)}\right)} \Delta\left(\boldsymbol{z}^{(k)}\right)$ for all $k \geq 0$;

Step 2: Newton (or quadratically convergence) phase
While $\lambda_f\left(\boldsymbol{z}^{(k)}\right) > \epsilon$ do $\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} - \Delta\left(\boldsymbol{z}^{(k)}\right)$ for all $k \geq k^*_{DP}$, where k^*_{DP} is the termination step in Step 1.

This algorithm can be viewed as a modification of the Newton's method and consists of two phases that help reduce the number of iterations. More specifically, Step 1, called the *damped phase*, guarantees that $f\left(\boldsymbol{z}^{(k)}\right) - f\left(\boldsymbol{z}^{(k+1)}\right) \geq \omega(\lambda^*)$ and in turn, the number of iterations in Step 1, denoted by N_{DP}, is bounded with

$$N_{DP} \leq \frac{f\left(\boldsymbol{z}^{(0)}\right) - f(\boldsymbol{z}^*)}{\omega(\lambda^*)}, \quad \text{where } \omega(\lambda) := \lambda - \log(1 + \lambda) \text{ on } \Re_+.$$

This represents the advantage of Algorithm 1 as compared to relying only on the Newton's method, see Theorem 4.1.10 of Nesterov (2004) or Section 9.6.4 of Boyd and Vandenberghe (2004) for further details on this issue.[19] The total number of iterations in Step 2 is $\log_2 \log_2(1/\epsilon)$ if an accuracy of $f\left(\boldsymbol{z}^{(k^*)}\right) - f(\boldsymbol{z}^*) \leq \epsilon$ is sought. The latter bound is very small, e.g., 4.32 and 5.82 for $\epsilon = 10^{-6}$ and $\epsilon = 10^{-17}$, respectively. Note that $\epsilon = 10^{-17}$ is the **MATLAB** machine epsilon, which is the top end tolerance level benchmark in **MATLAB**.

Remark 4.3. Inverting the Hessian is often challenging, and an alternative solution to computing the step/search direction, i.e. computing $\Delta(\boldsymbol{z}) := \left[\nabla^2 f(\boldsymbol{z})\right]^{-1} \nabla f(\boldsymbol{z})$ for a given $\boldsymbol{z}$, is to solve $\nabla^2 f(\boldsymbol{z})\, \boldsymbol{t} = \nabla f(\boldsymbol{z})$ in $\boldsymbol{t}$, which is a linear system of equations. If we denote by $t^*_f(\boldsymbol{z})$ the latter solution, we have $\Delta\left(\boldsymbol{z}^{(k)}\right) = t^*_f\left(\boldsymbol{z}^{(k)}\right)$ and

$$\lambda_f\left(\boldsymbol{z}^{(k)}\right) = \sqrt{\nabla f\left(\boldsymbol{z}^{(k)}\right)^\top \left[\nabla^2 f\left(\boldsymbol{z}^{(k)}\right)\right]^{-1} \nabla f\left(\boldsymbol{z}^{(k)}\right)}$$
$$= \sqrt{\nabla f\left(\boldsymbol{z}^{(k)}\right)^\top t^*_f\left(\boldsymbol{z}^{(k)}\right)}.$$

4.2. The ALM algorithm for the Inverse Gaussian Regression

We showed in Section 3.3 that the Inverse Gaussian Regression model is not proper for any *power* LF. However, it is still possible to create a tractable model for this parametric family for a particular *power* LF. Indeed, we assume a *reciprocal-square-root* LF (i.e. *power* LF from

[18] A function $f : A \subseteq \Re^d \to B$ is closed convex if f is convex and closed on A, where f is closed if for any $\alpha \in \Re$, $\{\boldsymbol{x} \in dom(f) : f(\boldsymbol{x}) \leq \alpha\}$ is a closed set.

[19] More formal convergence measures for Step 1 that are compared to the equivalent convergence measures of the standard Newton's method are available in Theorems 4.1.11 and 4.1.12 of Nesterov (2004).

(2.7) with $\gamma = -2$) which satisfies Condition **C1** but not Condition **C2** of Definition 2.1. This choice leads to solving the following (non-linear) optimisation problem:

$$\min_{\boldsymbol{\beta}\in\Omega} \; f_{IG}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(\frac{y_i}{2} \left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^4 - \left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2 \right). \tag{4.5}$$

The advantage of using the *reciprocal-square-root* LF is that (4.5) has a tractable solution via the *Alternating Linearisation Method (ALM)*, see e.g. Boyd et al. (2011b) for further details. More specifically, the variable β can be split into two variables, so that the *ALM* reformulation of (4.5) is given by:

$$\min_{(\boldsymbol{z},\boldsymbol{t})\in\Re^d\times\Re^d} \; G(\boldsymbol{z},\boldsymbol{t}) = \sum_{i=1}^{n} \left(\frac{y_i}{2} \left(\boldsymbol{x}_i^\top \boldsymbol{z}\right)^2 \left(\boldsymbol{x}_i^\top \boldsymbol{t}\right)^2 - \left(\boldsymbol{x}_i^\top \boldsymbol{z}\right) \left(\boldsymbol{x}_i^\top \boldsymbol{t}\right) \right)$$
$$\text{so that } \boldsymbol{z} = \boldsymbol{t}. \tag{4.6}$$

The iterative algorithm that efficiently solves (4.6) is given as Algorithm 2 and is an *Alternating Linearisation Method with backtracking (ALM-bktr)*, i.e. a bespoke *ALM* algorithm. This algorithm provides an approximation for $\boldsymbol{\beta}^*$, which denotes a local optimum of (4.5), by generating two sequences $\{\boldsymbol{z}_s : s \geq 0\}$ and $\{\boldsymbol{t}_s : s \geq 0\}$ such that $\boldsymbol{z}_s \to \boldsymbol{\beta}^*$ and/or $\boldsymbol{t}_s \to \boldsymbol{\beta}^*$. The main idea is to solve a two-block variant of (4.6), which is a *convex quadratic programming (QP)* instance in $\boldsymbol{z}$ for any given $\boldsymbol{t}$ that could be efficiently solved, and the same holds if $\boldsymbol{z}$ and $\boldsymbol{t}$ are interchanged. The *ALM* algorithm relies on replacing the function G by their linearisation and an additional regularisation factor in order to obtain an approximation to the initial objective function f_{IG} from (4.5). Thus, we define the following functions

$$H_1(\boldsymbol{z},\boldsymbol{t};\mu) := G(\boldsymbol{z},\boldsymbol{t}) + \langle G_2(\boldsymbol{t},\boldsymbol{t}), \boldsymbol{z}-\boldsymbol{t}\rangle + \frac{1}{2\mu}\|\boldsymbol{z}-\boldsymbol{t}\|_2^2,$$
$$H_2(\boldsymbol{z},\boldsymbol{t};\mu) := G(\boldsymbol{z},\boldsymbol{t}) + \langle G_1(\boldsymbol{z},\boldsymbol{z}), \boldsymbol{t}-\boldsymbol{z}\rangle + \frac{1}{2\mu}\|\boldsymbol{z}-\boldsymbol{t}\|_2^2,$$

where $\|\cdot\|_2$ is the L^2 norm on $\Re^d$, μ is a positive constant, and G_1 and G_2 are the partial derivatives of G given below:

$$G_1(\boldsymbol{z},\boldsymbol{t}) := \frac{\partial G}{\partial \boldsymbol{z}} = \sum_{i=1}^{n} \left(y_i \left(\boldsymbol{x}_i^\top \boldsymbol{z}\right) \left(\boldsymbol{x}_i^\top \boldsymbol{t}\right)^2 - \left(\boldsymbol{x}_i^\top \boldsymbol{t}\right) \right) \boldsymbol{x}_i,$$
$$G_2(\boldsymbol{z},\boldsymbol{t}) := \frac{\partial G}{\partial \boldsymbol{t}} = \sum_{i=1}^{n} \left(y_i \left(\boldsymbol{x}_i^\top \boldsymbol{z}\right)^2 \left(\boldsymbol{x}_i^\top \boldsymbol{t}\right) - \left(\boldsymbol{x}_i^\top \boldsymbol{z}\right) \right) \boldsymbol{x}_i.$$

Algorithm 2 for solving (4.5), and therefore (4.6), is described below.[20]

Algorithm 2: Standard ALM algorithm for solving (4.5).

Result: $(\boldsymbol{z}_{s^*}, \boldsymbol{t}_{s^*})$ that approximates $\boldsymbol{\beta}^*$, a local optimum of (4.5), where s^* is the termination step.

Choose $\mu_{1,0} = \mu_{2,0} = \mu_0 > 0$, $b \in (0,1)$, and $\boldsymbol{z}_0 = \boldsymbol{t}_0 \in \Re^d$;
for $s \in \{0, 1, \ldots\}$ **do**
  $\boldsymbol{z}_{s+1} := \underset{\boldsymbol{z}\in\Re^d}{\arg\min}\; H_1(\boldsymbol{z}, \boldsymbol{t}_s; \mu_{1,s})$;
  if $f_{IG}(\boldsymbol{z}_{s+1}) \leq H_1(\boldsymbol{z}_{s+1}, \boldsymbol{t}_s; \mu_{1,s})$ **then**
    choose $\mu_{1,s+1} \geq \mu_{1,s}$;
  else
    find the lowest $n_{1,s} \geq 1$ such that $f_{IG}(\boldsymbol{u}_{1,s}) \leq H_1\left(\boldsymbol{u}_{1,s}, \boldsymbol{t}_s; \mu_{1,s}^*\right)$,
      where $\mu_{1,s}^* = \mu_{1,s} b^{n_{1,s}}$ and $\boldsymbol{u}_{1,s} := \underset{\boldsymbol{z}\in\Re^d}{\arg\min}\; H_1\left(\boldsymbol{z}, \boldsymbol{t}_s; \mu_{1,s}^*\right)$;
    $\mu_{1,s+1} := \mu_{1,s}^*/b$ and $\boldsymbol{z}_{s+1} := \boldsymbol{u}_{1,s}$;
  end
  $\boldsymbol{t}_{s+1} := \underset{\boldsymbol{t}\in\Re^d}{\arg\min}\; H_2(\boldsymbol{z}_{s+1}, \boldsymbol{t}; \mu_{2,s})$;
  if $f_{IG}(\boldsymbol{t}_{s+1}) \leq H_2(\boldsymbol{z}_{s+1}, \boldsymbol{t}_{s+1}; \mu_{2,s})$ **then**
    choose $\mu_{2,s+1} \geq \mu_{2,s}$;
  else
    find the lowest $n_{2,s} \geq 1$ such that
      $f_{IG}(\boldsymbol{u}_{2,s}) \leq H_2\left(\boldsymbol{z}_{s+1}, \boldsymbol{u}_{2,s}; \mu_{2,s}^*\right)$, where $\mu_{2,s}^* = \mu_{2,s} b^{n_{2,s}}$ and
      $\boldsymbol{u}_{2,s} := \underset{\boldsymbol{t}\in\Re^d}{\arg\min}\; H_2\left(\boldsymbol{z}_{s+1}, \boldsymbol{t}; \mu_{2,s}^*\right)$;
    $\mu_{2,s+1} := \mu_{2,s}^*/b$ and $\boldsymbol{t}_{s+1} := \boldsymbol{u}_{2,s}$;
  end
end

5. Simulation study

This section presents several numerical experiments to determine the efficiency and accuracy of the proposed algorithms and investigates to what extent they can improve the standard built-in GLM libraries from various software. Specifically, we implement the *NSC* Algorithm 1 introduced in Section 4.1 for the Poisson (with *half-square-root* LF) and Gamma Regressions (with *half-reciprocal-square-root* LF), and the *ALM* Algorithm 2 introduced in Section 4.2 for solving Inverse Gaussian Regressions (with *reciprocal-square-root* LF).

Before discussing our numerical analyses, we would like to mention that the performance of our *NSC* Algorithm 1 is compared to standard IRLS implementations in **MATLAB**, **R** and **Python**, i.e., built-in functions for non-penalised GLM, since prediction through GLM is the main aim of this paper. Moreover, we also illustrate that it is worth considering our *NSC* Algorithm 1 as an alternative to IRLS to perform GLM modelling. It is well-known that predictive models benefit from adding penalisations to the objective function in order to improve the prediction error, and many off-the-shelf pieces of software offer such an option. We have not included a comparison with penalised GLM models since the penalisation functions are SC functions, and thus, the augmented objective function would be SC as well and Theorem 4.2 holds whenever such penalisation is added. Specifically, the objective functions in (4.1)–(4.4) are SC functions if such penalisations are added, and our theoretical results would hold. In addition, since the efficiency of a penalised predictive model heavily depends upon the range of values for the penalisation parameters, it would be more informative to focus on comparing our *NSC* Algorithm 1 to IRLS in settings without penalisations and raise awareness that advanced optimisation techniques could enhance the standard IRLS deployment for GLM estimation, which is the main message of our numerical implementation.

For each specification of the number of observations n and number of covariates d, we synthetically construct N data generating processes (henceforth called DGP) and perform the above GLM estimations using both algorithms.[21] The effectiveness of our methods is determined by comparing our estimates with the "true" regression coefficients $\boldsymbol{\beta}_k$, for any $k = 1, \ldots, N$, obtained by using three standard built-in packages: **MATLAB** *fitglm*, **R** *glm2* and **Python** statsmodels *sm.GLM* libraries.[22] To assess the accuracy of Algorithms 1 and 2 relative to these benchmarks we consider two performance indicators. First, we compute the Absolute Error Ratio (*AER*) and its mean (*MAER*), defined as:

[20] The algorithm stops whenever $\sum \frac{|\boldsymbol{z}_{s+1} - \boldsymbol{t}_{s+1}|}{|\boldsymbol{z}_{s+1}|}$ reaches the user's defined value (e.g. the default value in our numerical examples is taken to be 10^{-4} to balance the speed and precision with other benchmark algorithms). Once the process is stopped, we use $\boldsymbol{z}_{s+1}$ (or $\boldsymbol{t}_{s+1}$) if H_1 is smaller (or larger) than H_2.

[21] Note that unlike in the theoretical presentation, d represents here the number of covariates excluding the trivial one corresponding to the intercept β_0, so that the full matrix of explanatory variables is obtained by adding the n-dimensional unit vector to $\boldsymbol{X}$. Details on the DGP simulation are illustrated in Appendix C.

[22] We remark that all three software rely on the IRLS method to estimate the regression coefficients. Generally speaking, **R** *glm2* provides an improvement over the standard **R** *glm* package by using the step-halving approach in order to improve the convergence properties of IRLS (see e.g. Marschner (2011)).

$$MAER = \frac{1}{N}\sum_{k=1}^{N} AER_k \quad \text{with}$$

$$AER_k = \frac{AE\left(\hat{\boldsymbol{\beta}}_k^{alg}\right)}{AE\left(\hat{\boldsymbol{\beta}}_k^{benchmark}\right)},\ k = 1, \ldots, N. \tag{5.1}$$

Here, the Absolute Error (AE) associated with each estimator $\hat{\boldsymbol{\beta}}_k$ is defined by the L^1-norm:

$$AE\left(\hat{\boldsymbol{\beta}}_k\right) = \sum_{j=1}^{d} |\hat{\beta}_{k,j} - \beta_{k,j}^{true}|, \tag{5.2}$$

where $\beta_{k,j}^{true}$ is the j^{th} component of the k^{th} simulated "true" regression coefficient according to the DGP scheme outlined in Appendix C, and $\hat{\beta}_{k,j}^{alg}$ and $\hat{\beta}_{k,j}^{benchmark}$ are their corresponding estimated values obtained with Algorithms 1 and 2, and the three software benchmark packages, respectively. The performance of our approach is further evaluated by computing the log-likelihood ratio statistics, which compare the GLM with the saturated model. Thus, we introduce below the Deviance Ratio (DR) and its mean (MDR):

$$MDR = \frac{1}{N}\sum_{k=1}^{N} DR_k \quad \text{with} \quad DR_k = \frac{D\left(\hat{\boldsymbol{\beta}}_k^{alg}\right)}{D\left(\hat{\boldsymbol{\beta}}_k^{benchmark}\right)},\ k = 1, \ldots, N. \tag{5.3}$$

Here, the Deviance (D) of each GLM is defined by:

$$D\left(\hat{\boldsymbol{\beta}}_k\right) = -2\phi\left(\ell\left(\hat{\boldsymbol{\beta}}_k\right) - \ell_s\right), \tag{5.4}$$

where $\ell\left(\hat{\boldsymbol{\beta}}_k\right)$ is the log-likelihood function corresponding to the fitted GLM for the k^{th} simulated DGP scenario, while ℓ_s is the maximum value of the log-likelihood of the saturated model that is computed using the same function as in (2.4) with $\theta_i = b'^{-1}\left(y_i\right)$. Explicit expressions for the deviance of all GLMs considered in our numerical experiments are provided in Appendix D. Note that an $MAER$ or MDR value smaller than 1 indicates that our approach is more accurate on average than the benchmark with respect to the corresponding performance measure.

The efficiency of our algorithms relative to their benchmarks is also investigated by reporting the Mean Computational Time Ratio ($MCTR$) introduced as:

$$MCTR = \frac{1}{N}\sum_{k=1}^{N} CTR_k \quad \text{with}$$

$$CTR_k = \frac{CT\left(\hat{\boldsymbol{\beta}}_k^{alg}\right)}{CT\left(\hat{\boldsymbol{\beta}}_k^{benchmark}\right)},\ k = 1, \ldots, N. \tag{5.5}$$

Here, $CT\left(\hat{\boldsymbol{\beta}}_k^{alg1}\right)$ and $CT\left(\hat{\boldsymbol{\beta}}_k^{benchmark}\right)$ are the Algorithm 1 and benchmark computational times recorded for the k^{th} simulated DGP scenario, respectively. It follows that our algorithms are faster on average whenever $MCTR < 1$.[23] For a consistent and fair comparison of the computational time efficiency, all benchmarks have been implemented using their corresponding default starting values, and the same specifications in the optimisation procedure, i.e. maximum number of iterations = 10,000 and tolerance level = 10^{-6}. Since Algorithms 1 and 2 are coded in **MATLAB**, we use the **MATLAB** fitglm starting values for our estimations.

The performance indicators $MAER$ and MDR (both in **bold**), and $MCTR$ are computed based on $N = 500$ replicates. Note that Algorithms 1 and 2 always converge within a very reasonable number of iterations, which is not the case for the three benchmarks. Therefore, the number of replicates (out of 500 simulations) for which the optimisation problem (associated with the benchmarks) does not converge within the allocated maximum number of iterations is illustrated as #NaN in our tables. Consequently, these cases are discarded from the computation of our performance indicators so that the benchmarks' performance is computed in the most advantageous possible to those benchmarks.

Table 2 presents the results for the Poisson GLM regression. We first notice that in terms of accuracy, Algorithm 1 consistently outperforms both **MATLAB** *fitglm* and **Python** *sm.GLM* libraries for all cases considered. The improvements are relative to **Python** *sm.GLM* is quite significant with respect to both $MAER$ and MDR with the largest augmentations being noticed for larger scale settings when the ratio between the sample size and the number of covariates/features decreases; for example, when $n/d = 5$, the improvements for both indicators are on average of around 15%, 37% and 53% for $n = 100, 500$ and 1,000, respectively. The $MAER$ and MDR for the **MATLAB** *fitglm* benchmark are closer to 1, but unlike in the previous case, there are many scenarios when the *fitglm* MLE does not converge. This typically happens for the bigger scale problems, as it is the case when $n = 1{,}000$ and $d = 200$ (our largest setting) where convergence was not achieved in half of the cases. Unlike the **MATLAB** and **Python** libraries, **R** *glm*2 seems to perform very similarly to our Algorithm 1 for the Poisson GLM, the $MAER/MDR$ values being typically slightly above/below 1. The $MCTR$ values indicate that Algorithm 1 is always more efficient than both **Python** *sm.GLM* and **MATLAB** *fitglm*, with the largest improvements observed for small dimension settings. The smallest differences in runtime happen when $n = 1{,}000$ and $d = 50$, when our algorithm is five and seven times faster than the aforementioned benchmarks, respectively. However, while **R** *glm*2 is also slower when $n = 100$ than our Algorithm 1, it becomes more efficient for larger values of n.

The Gamma GLM results are illustrated in Table 3. First, we notice that Algorithm 1 consistently outperforms all benchmarks in terms of both accuracy and efficiency. Unlike in the Poisson case, our method performs significantly better than **R** *glm*2 with respect to both accuracy indicators, with an average improvement ranging from $40\% - 77\%$ and $40\% - 68\%$ for $MAER$ and MDR, respectively, when $n = 1{,}000$. We further notice a reverse situation regarding the **MATLAB** *fitglm* and **Python** *sm.glm* GLM libraries when compared to the results from Table 2. Specifically, on the one hand, the MLE procedure from **Python** *sm.glm* does not converge in many instances, but when it converges, the estimates are very close to those obtained via Algorithm 1. On the other hand, despite always converging, the **MATLAB** *fitglm* optimisation produces $MAER$ and MDR values which are significantly lower than 1, with the lowest values recorded when $n = 1{,}000$. The reported average computational times favour again our methodology; only $MCTR$ values greater than 1 are spotted for the larger scale settings for **R** *glm*2, which provided inaccurate estimates in all these cases.

In summary, based on our DGP for Poisson and Gamma GLMs, we can argue that overall, our Algorithm 1 provides the most accurate and efficient estimation approach relative to the three benchmarks, while **R** *glm*2 is the second best, generally speaking being more stable than the **Python** *sm.GLM* and **MATLAB** *fitglm* counterparts.[24]

We next turn our attention to the implementation results of the *ALM* Algorithm 2 for solving Inverse Gaussian Regressions based on the *reciprocal-square-root* LF. The benchmark chosen in our analysis is the **MATLAB** *fitglm* package and we only focus on the accuracy of our methodology. Fig. 2 illustrates the box plots of the **MATLAB** *fitglm*-based AER and DR for the same values of n and d as in the previous tables. First, we notice (in all nine cases) that the AER indicators are more or less

[23] Note that for streamline purposes we only report the $MCTR$ values for Algorithm 1.

[24] Note that these conclusions are drawn solely based on our DGP and a limited number of experiments, so further implementations may be needed to further investigate this problem.

Table 2
MAER, MCTR and MDR for Poisson GLM.

		n = 100			n = 500			n = 1,000		
		d = 5	d = 10	d = 20	d = 25	d = 50	d = 100	d = 50	d = 100	d = 200
MATLAB	**MAER**	**0.9730**	**0.9620**	**0.9523**	**0.9685**	**0.9721**	**0.9713**	**0.9758**	**0.9782**	**0.9816**
fitglm	**MDR**	**0.9947**	**0.9935**	**0.9883**	**0.9977**	**0.9986**	**0.9970**	**0.9998**	**1.0002**	**1.0021**
	MCTR	0.0134	0.0169	0.0272	0.0630	0.0625	0.0762	0.1446	0.1012	0.1069
	#NaN	16	32	58	37	67	182	46	87	256
Python	**MAER**	**0.9393**	**0.8998**	**0.8431**	**0.9002**	**0.8463**	**0.6227**	**0.8838**	**0.8131**	**0.4723**
sm.GLM	**MDR**	**0.9177**	**0.8972**	**0.8518**	**0.9093**	**0.8553**	**0.6268**	**0.8915**	**0.8166**	**0.4721**
	MCTR	0.0065	0.0082	0.0129	0.0551	0.0531	0.0340	0.2016	0.1022	0.0531
	#NaN	0	0	0	0	0	0	0	0	0
R	**MAER**	**0.9999**	**0.9967**	**1.0014**	**1.0082**	**1.0085**	**1.0161**	**1.0087**	**1.0168**	**1.0376**
*glm*2	**MDR**	**0.9579**	**0.9708**	**0.9858**	**0.9832**	**0.9882**	**0.9950**	**0.9870**	**0.9911**	**1.0057**
	MCTR	0.2553	0.2815	0.5043	1.5819	1.5695	1.0513	3.3093	2.0093	1.3328
	#NaN	0	0	0	0	0	0	0	0	0

Notes: This table reports the **Mean Absolute Error Ratio (***MAER***)**, **Mean Deviance Ratio (***MDR***)** and Mean Computational Time Ratio (*MCTR*) of Algorithm 1 from Section 4.1 relative to its benchmarks, **MATLAB** *fitglm*, **Python** *sm.GLM* and **R** *glm*2, for the Poisson GLM equipped with the *half-power* LF from (2.8) with $\gamma = 2$. These indicators are computed based on the MLE values obtained from $N = 500$ simulations according to the DGP scheme outlined in Appendix C, for different specifications for the number of observations n and number of covariates d. The number of replicates (out of 500 simulations) that the benchmarks cannot converge is shown as #NaN. All benchmarks are implemented using the same starting values with a maximum of 10,000 iterations and 10^{-6} tolerance level.

Table 3
MAER, MCTR and MDR for Gamma GLM.

		n = 100			n = 500			n = 1,000		
		d = 5	d = 10	d = 20	d = 25	d = 50	d = 100	d = 50	d = 100	d = 200
MATLAB	**MAER**	**0.9216**	**0.9449**	**0.9722**	**0.6554**	**0.7141**	**0.8469**	**0.5547**	**0.5734**	**0.7167**
fitglm	**MDR**	**0.9534**	**0.9511**	**0.9687**	**0.6713**	**0.6753**	**0.8061**	**0.5065**	**0.4424**	**0.6202**
	MCTR	0.0579	0.0270	0.0404	0.2549	0.1142	0.0995	0.5530	0.1991	0.1954
	#NaN	0	0	0	0	0	0	0	0	0
Python	**MAER**	**0.9831**	**0.9930**	**0.9989**	**0.9962**	**0.9999**	**1.0000**	**0.9932**	**1.0000**	**1.0000**
sm.GLM	**MDR**	**0.9953**	**0.9980**	**1.0000**	**0.9998**	**1.0000**	**1.0000**	**0.9997**	**1.0000**	**1.0000**
	MCTR	0.0700	0.2049	0.2314	1.6705	0.9847	0.5505	3.7401	2.0635	0.8492
	#NaN	78	55	21	406	268	124	471	373	206
R	**MAER**	**0.9450**	**0.9608**	**0.9850**	**0.5843**	**0.7216**	**0.8928**	**0.4018**	**0.5434**	**0.7679**
*glm*2	**MDR**	**0.9496**	**0.9621**	**0.9878**	**0.5887**	**0.6859**	**0.8585**	**0.3944**	**0.4643**	**0.6840**
	MCTR	0.2945	0.5550	0.5101	6.5451	3.4493	1.6073	12.2574	5.1892	1.5737
	#NaN	0	0	0	0	0	0	0	0	0

Notes: This table reports the **Mean Absolute Error Ratio (***MAER***)**, **Mean Deviance Ratio (***MDR***)** and Mean Computational Time Ratio (*MCTR*) of Algorithm 1 from Section 4.1 relative to its benchmarks, **MATLAB** *fitglm*, **Python** *sm.GLM* and **R** *glm*2, for the Gamma GLM equipped with the *half-power* LF from (2.8) with $\gamma = -2$. These indicators are computed based on the MLE values obtained from $N = 500$ simulations according to the DGP scheme outlined in Appendix C, for different specifications for the number of observations n and number of covariates d. The number of replicates (out of 500 simulations) that the benchmarks cannot converge is shown as #NaN. All benchmarks are implemented using the same starting values with a maximum of 10,000 iterations and 10^{-6} tolerance level.

symmetrically distributed around 1, with a median value smaller (but closer) to 1, suggesting that our Algorithm 2 slightly outperforms **MATLAB** *fitglm* relative to this performance measure. However, our method performs much better in terms of the deviance measure, as almost all DR values are below 1, with the most significant differences being documented for larger dimension problems and the smallest n/d ratio (i.e. $n/d = 5$). Furthermore, for each value of n, we notice a decreasing trend in the median of DRs as the number of covariates increases. These observations are consistent with the previous findings on Algorithm 1 regarding the significant improvements in accuracy for bigger datasets.

6. Real data analyses

This section replicates some of the numerical illustrations in Section 5 for two insurance datasets: i) *health insurance* and ii) *flood insurance*. The statistical description of these two datasets and their preprocessing steps are detailed in Appendix E. Specifically, we compare the out-of-sample performances of Algorithm 1 (from Section 4.1) against benchmark GLM implementations in **MATLAB** *fitglm*, **Python** *sm.GLM*, and **R** *glm*2.

For the health insurance dataset, Algorithm 1 is applied to a Poisson GLM with the *half-power* LF defined in (2.8) with $\gamma = 2$. Benchmarks are implemented using Poisson GLMs with either the *half-power* LF in (2.8)

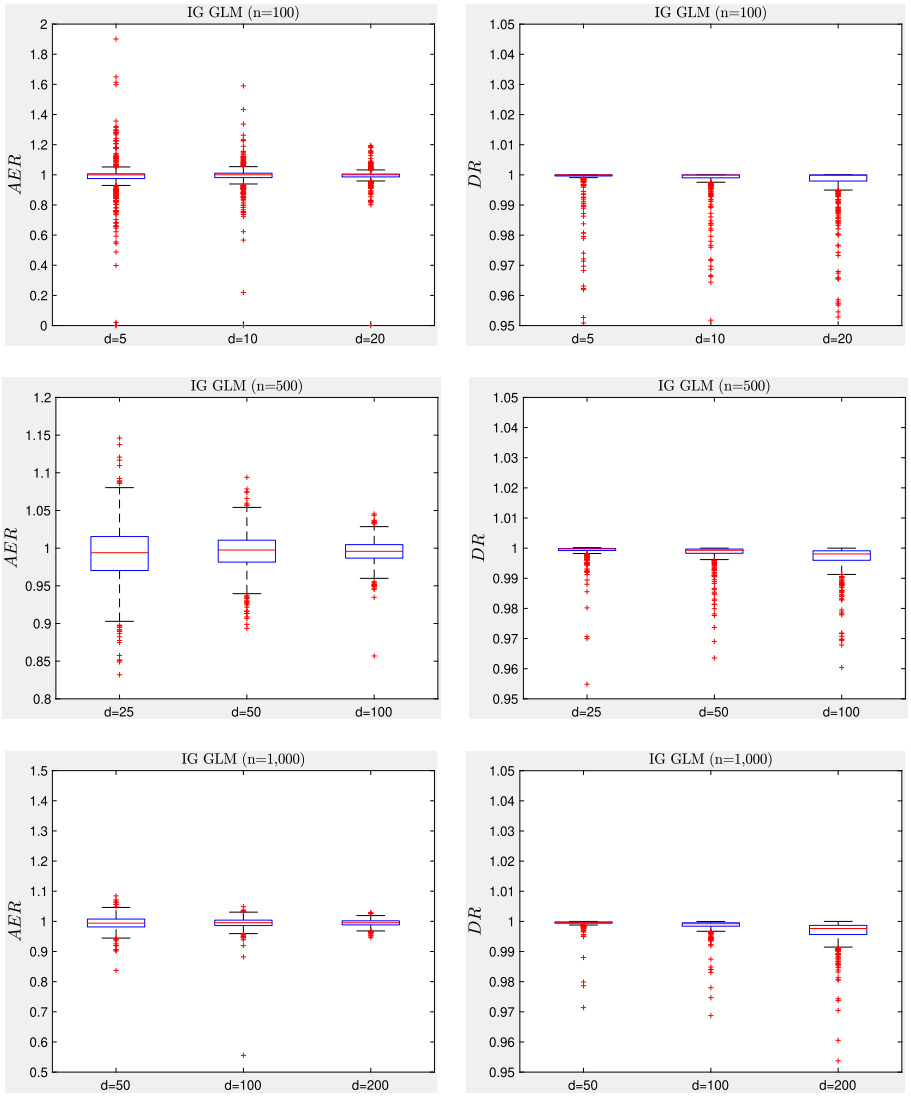

Fig. 2. Absolute Error Ratio (*AER*) and Deviance Ratio (*DR*) for Inverse Gaussian GLM based on **MATLAB** *fitglm*.
Notes: This figure shows the box plots of Absolute Error Ratio (*AER*) in the left panel and Deviance Ratio (*DR*) in the right panel of Algorithm 2 from Section 4.2 relative to the **MATLAB** *fitglm* benchmark for the Inverse Gaussian GLM based on the *reciprocal-square-root* LF. Each box plot is constructed using *AER*s and *DR*s computed based on MLE values obtained from $N = 500$ simulations according to the DGP scheme outlined in Appendix C, for different specifications for the number of observations n and the number of covariates d. All implementations use the same starting value with a maximum of 10,000 iterations and 10^{-6} tolerance level.

with $\gamma = 2$ or the *log* LF. For the flood insurance dataset, Algorithm 1 is applied to a Gamma GLM with the *half-power* LF in (2.8) with $\gamma = -2$, while the benchmarks use Gamma GLMs with either the *half-power* LF defined in (2.8) with $\gamma = -2$ or the *log* LF. Results referring to GLM benchmarks with the *log* LF are labelled, for example, as "**R** *glm*2 Log", and those with the *half-power* LF are labelled as "**MATLAB** *fitglm* Hp".

For both datasets, we compute ratios of the *mean squared error (MSE)* of Algorithm 1 predictions and those corresponding to the six benchmarks considered. An MSE ratio less than one indicates that Algorithm 1 outperforms the benchmark, with smaller values signifying better performance. For the health insurance dataset, the data is split into 70% training and 30% testing sets, and this process is repeated 100 times. The training data is used to estimate model parameters, and predictions are made on the testing data to evaluate the MSE ratios. For the flood insurance dataset, we analyse claims from three US states, including Florida, Texas, and Louisiana, which are prone to severe flooding. In this case, we use the previous year's data (e.g., 2010) as the training set to predict claims for the next year (e.g., 2011), repeating this process annually from 2011 to 2023.

6.1. U.S. health insurance dataset

The results for the *first* U.S. health insurance dataset are summarised in Table 4. Among 100 data splits, Algorithm 1 consistently outperformed the other two GLM benchmark methods (*Log* LF and *half-power* LF) across all software platforms. From Panel A of Table 4, we notice that Algorithm 1 achieved the lowest MSE in 77 scenarios for both **MATLAB** *fitglm* and **R** *glm*2, and in all 100 cases for **Python** *sm.GLM*. The "wins" counts further highlight that Algorithm 1 achieved lower MSEs than GLMs with *Log* LF and *half-power* LF in 86 and 82 cases, respectively, for both **MATLAB** *fitglm* and **R** *glm*2, and in all 100 scenarios for **Python** *sm.GLM*.

Panel B of Table 4 provides the average MSE ratios and quantiles for the benchmark methods relative to Algorithm 1. The average MSE ratios for GLMs with *Log* LF and *half-power* LF for both **MATLAB** *fitglm* and **R** *glm*2 are slightly above 0.98 (0.9829 and 0.9882, respectively), indicating a limited improvement of around 1.7% and 1.2% for Algorithm 1. In contrast, the average MSE ratios in **Python** *sm.GLM* are much smaller (0.0578 for *Log* LF and 0.2180 for *half-power* LF), demonstrating a more significant improvement for this software. This trend is consistent across

Table 4
Poisson GLM for Health data.

	MATLAB *fitglm*			**Python** *sm.GLM*			**R** *glm2*		
	Alg_1	Log	Hp	Alg_1	Log	Hp	Alg_1	Log	Hp
Panel A: Model Counts									
"Best" Model Counts	77	16	7	100	0	0	77	7	16
Alg_1 "Wins" Counts	-	86	82	-	100	100	-	86	82
Panel B: MSE Ratios									
Average	-	0.9829	0.9882	-	0.0578	0.2180	-	0.9829	0.9882
25% quantile	-	0.9735	0.9781	-	0.0544	0.2064	-	0.9735	0.9785
50% quantile	-	0.9826	0.9886	-	0.0577	0.2162	-	0.9826	0.9886
75% quantile	-	0.9938	0.9961	-	0.0619	0.2301	-	0.9936	0.9961

Notes: Panel A: Model Counts summarises the results of 100 data splits for GLM implementations of Algorithm 1 from Section 4.1, which uses the *half-power* LF with $\gamma = 2$, compared to six benchmarks that use both the *half-power* LF with $\gamma = 2$ and the *log* LF, across **MATLAB**, **Python**, and **R**. The first row (*"Best" Model Counts*) shows how many times each model achieved the lowest MSE in each software. The second row (*Alg_1 "Wins" Counts*) indicates how many times Algorithm 1 (**Alg_1**) achieved a lower MSE compared to the *Log* LF and *half-power* LF models. *Panel B: MSE Ratios* presents the average MSE ratios along with the 25%, 50%, and 75% quantiles, calculated across 100 data splits for each method. All methods used default starting values provided by the software packages, with a maximum of 10,000 iterations and a tolerance level of 10^{-6}. For Algorithm 1, the starting values were taken from the first iteration in each respective software package.

Table 5
Original Data and Percentage Contributions for Selected Examples.

Rows	Charges	Original Data Information						Percentage Contributions (%)					
		Age	Sex	BMI	Children	Smoker	Region	Age	Sex	BMI	Children	Smoker	Region
Insured 1	12,829.46	18	M	17.29	2	Yes	NE	0.98	-0.37	58.22	2.46	38.71	0.00
Insured 2	42,303.69	55	M	30.69	0	Yes	NE	11.28	-0.31	57.30	0.00	31.72	0.00
Insured 3	36,580.28	61	F	33.33	4	No	SE	24.82	0.00	74.67	3.85	0.00	-3.34

Notes: This table presents the results for three selected examples from the U.S. health insurance dataset. The first part (*Original Data Information*) includes the characteristics of each selected individual, such as age (18-64), sex (Male or Female), BMI (15.96-53.13), number of children (0-5), smoker status (Yes or No), and region (NE for Northeast, NW for Northwest, SE for Southeast, and SW for Southwest). The second part (*Percentage Contributions (%)*) reports the percentage contribution of each covariate to the predicted premium. The contributions are calculated based on the estimated coefficients obtained from Algorithm 1 with a power LF ($\gamma = 2$). The contributions for one-hot encoded variables such as **Region** and **Age** are summed for all corresponding categories. Some contributions may appear negative due to the interpretability model formulation, but they are consistent with the theoretical results in the Section 2.2. The negative percentage contributions for some covariates (i.e. Sex and Region) for a given insured indicate that these covariates are in fact decreasing the overall value of the corresponding predicted premium (when combining with other available covariates). A relevant domain knowledge or a further detailed investigation would help researchers to understand better about these covariates and their contributions.

all quantiles (25%, 50%, and 75%) for **Python** *sm.GLM*, highlighting that Algorithm 1 is particularly effective in reducing errors when compared to these benchmarks.

In addition to the out-of-sample performance summarised in Table 4, while Table 5 illustrates the interpretability of Algorithm 1 through the percentage contribution vectors in (2.11) for three selected examples from the U.S. health insurance dataset. The three examples correspond to three insureds that were chosen to capture diverse characteristics of the data. The first part of the table reports the original data information for each example, such as age, sex, BMI, number of children, smoker status, and region. The second part presents the percentage contributions of each covariate to the predicted premium, calculated based on the interpretability framework introduced in Section 2.2.

The results emphasise the varying influence of covariates across individuals, showing that some contributions are substantial (e.g., BMI and smoking status), while others may be negligible or even negative (e.g., sex and region). For example, the negative contribution of the Region feature for Insured 3 shows that, given this individual's age (higher than the average age amongst the cohort) and high BMI, the Southeast region might slightly lower the predicted premium. This could be because age and BMI have a much stronger impact, while the effect of the region is relatively smaller and interacts differently with the other factors. Additionally, the minimal contribution of gender indicates that Algorithm 1 is not highly sensitive to this factor, demonstrating its robustness and fairness in the context of insurance pricing. The age column highlights a clear trend where the percentage contribution increases as the individual gets older. For instance, the percentage contribution of age is 0.98% for the youngest individual (18 years old), 11.28% for the middle-aged individual (55 years old), and 24.82% for the oldest individual (61 years old). This progression is consistent with the expected influence of age on health insurance premiums, as older individuals are generally associated with higher risks and, therefore, higher premiums. These patterns align with the theoretical model, in which all contributions sum to 100%, although some contributions are negative. This table shows that the Algorithm 1 not only improves predictive performance but also provides a clear and practical interpretation on the role of individual covariates in insurance pricing.

6.2. Flood insurance dataset

The results for the *second* flood insurance dataset are summarised in Table 6, which compares the performance of Algorithm 1 relative to the six benchmark methods across 13 years of out-of-sample predictions for Florida, Texas, and Louisiana in the U.S. Algorithm 1 demonstrates strong performance in most cases. For instance, in the Panel B of Table 6, Algorithm 1 achieves 23 wins with the *log* LF and 29 wins with the *half-power* LF in **MATLAB** *fitglm*. Additionally, Algorithm 1 has the highest number of times achieving the best performance across all methods in **MATLAB** *fitglm*, with 19 cases for the *log* LF and 39 cases for the

Table 6
Gamma GLM for Flood data.

Training Year	Predicted Year	**MATLAB** *fitglm*		**Python** *sm.GLM*		**R** *glm*2	
		Log	Hp	Log	Hp	Log	Hp
Florida (FL)							
2010	2011	24.6196	1.0144	0.0142	0.0142	24.6136	1.0144
2011	2012	31.5413	1.0675	0.0151	0.0150	6.9684	0.9777
2012	2013	3.0440	1.0176	0.0039	0.0039	3.0432	1.0178
2013	2014	14.6594	0.9948	0.0540	0.0538	14.6575	0.9946
2014	2015	3.8898	0.9993	0.0048	0.0048	3.8897	0.9993
2015	2016	0.0146	0.9982	NaN	0.0146	0.3764	0.9985
2016	2017	0.0528	0.6760	NaN	0.0068	0.2913	1.0588
2017	2018	0.1236	0.3160	NaN	0.0233	0.2364	1.1073
2018	2019	0.0194	0.5719	0.0199	0.0199	0.4000	0.9852
2019	2020	0.0537	0.9486	0.0531	0.0529	5.6086	0.9487
2020	2021	0.0106	0.9769	NaN	0.0106	0.1947	0.9773
2021	2022	0.1697	0.9718	0.1678	0.1679	0.7980	0.9741
2022	2023	0.8357	1.0013	0.0857	0.0857	0.8351	1.0015
Texas (TX)							
2010	2011	5.8798	1.0557	0.0022	0.0022	5.8782	1.0556
2011	2012	6.9224	0.6053	0.0249	0.0248	6.9144	0.6054
2012	2013	8.5670	0.9641	0.0239	0.0239	8.5657	0.9638
2013	2014	3.3014	0.9643	0.0023	0.0023	3.3009	0.9643
2014	2015	4.7864	0.9913	0.0479	0.0477	4.7857	0.9911
2015	2016	18.5862	0.9181	0.0110	0.0110	4.5940	1.0097
2016	2017	0.0291	0.9522	0.0272	0.0270	0.3640	0.9882
2017	2018	0.0488	0.8019	NaN	0.0301	0.4833	1.9563
2018	2019	0.1961	0.6751	0.0541	0.0543	0.3690	0.9855
2019	2020	0.0046	0.9463	NaN	0.0046	0.1951	0.9462
2020	2021	0.0179	0.7292	0.0112	0.0112	0.5730	1.0259
2021	2022	0.8210	0.9790	0.0510	0.0510	0.7251	0.9812
2022	2023	0.7062	1.0305	0.0642	0.0642	0.7320	1.0312
Louisiana (LA)							
2010	2011	3.0166	0.9570	0.0364	0.0364	3.0166	0.9569
2011	2012	0.0100	0.5317	0.0120	0.0119	0.2573	0.9533
2012	2013	10.2523	1.3220	0.0137	0.0137	10.2518	1.3224
2013	2014	2.0511	0.8257	0.0100	0.0101	2.0509	0.8257
2014	2015	7.5954	1.0932	0.0118	0.0118	7.5945	1.0933
2015	2016	3.9441	0.7310	0.0686	0.0682	3.9542	0.7311
2016	2017	0.0471	0.9164	NaN	0.0062	0.2124	1.1408
2017	2018	0.0096	1.0854	0.0095	0.0095	0.2488	1.0863
2018	2019	0.0445	0.5151	0.0307	0.0311	0.5027	0.9928
2019	2020	0.0600	0.3827	0.0103	0.0103	0.2688	0.9917
2020	2021	0.0157	1.0746	0.0157	0.0157	0.2491	1.0746
2021	2022	0.0115	0.9746	0.0115	0.0115	0.3276	0.9760
2022	2023	0.6846	0.4587	0.0374	0.0374	0.6143	0.9206
Panel A: Statistics Summary (FL, TX, LA)							
2011-2023	Average	4.0165	0.8727	0.0314	0.0282	3.3062	1.0160
	25% quantile	0.0368	0.7301	0.0112	0.0105	0.3458	0.9692
	50% quantile	0.6846	0.9641	0.0178	0.0150	0.7320	0.9917
	75% quantile	4.3652	1.0003	0.0487	0.0369	4.6898	1.0286
Panel B: Counts Summary (FL, TX, LA)							
2011-2023	Alg_1 "wins"	23	29	39	39	22	24
	Alg_1 "best"	**19**		**39**		12	
	Log "best"	16		0		**18**	
	Hp "best"	4		0		9	

Notes: This table compares the **MSE** values for Gamma GLM implementations of Algorithm 1 (**Alg_1**) from Section 4.1 using the *half-power* LF from (2.8) with $\gamma = -2$ to six benchmark methods. The benchmarks use both the *half-power* LF from (2.8) with $\gamma = -2$ and the *log* LF. All implementations rely on default starting values provided by the respective software packages, with a maximum of 10,000 iterations and a tolerance level of 10^{-6}. For Algorithm 1, the starting values are initialised from the *first iteration* results in **MATLAB**, **Python**, and **R**. The first three panels present the results of 13 out-of-sample evaluations (2011 to 2023) for three states (FL, TX, and LA) with the most frequent flood claims. Failures to converge are marked as "NaN" in the ***Python*** *sm.GLM Log* column, while Algorithm 1 successfully converges in all cases. Panel A displays the average and the 25%, 50%, and 75% quantiles of the 39 ratios computed for each column. Panel B shows the number of times Algorithm 1 outperforms the other methods in all settings and shows the best implementation of each method within each respective software.

half-power LF in **Python** *sm.GLM*. This highlights the algorithm's consistent ability to deliver optimal results in these software environments. However, its performance in **R** *glm*2 is mixed. While the *log* LF achieves the best results in 18 cases, Algorithm 1 still outperforms the *half-power* LF, which is similar in formulation, in 12 instances. These results indicate that while Algorithm 1 is generally effective, the *log* LF can perform better under specific conditions. Nevertheless, Algorithm 1 achieves notable success in delivering the best results across a significant portion of the tests.

Panel A of Table 6 provides additional insights into the relative performance of the methods. For the *log* LF, Algorithm 1 has an average MSE ratio of 4.02 in **MATLAB** *fitglm* and 3.31 in **R** *glm*2, indicating that the benchmark methods perform better for certain years. However, for the *half-power* LF, Algorithm 1 achieves an average MSE ratio of 0.87 in **MATLAB** *fitglm* and 1.02 in **R** *glm*2, showing comparable or better performance. These results suggest that the *half-power* LF is better aligned with the properties of Algorithm 1, leading to improved predictions in many scenarios.

One key advantage of Algorithm 1 is its stability. As shown in the first three panels in Table 6, convergence failures occur frequently for the *log* LF in **Python** *sm.GLM*, resulting in missing values "NaN"). In contrast, Algorithm 1 converges successfully in all cases, demonstrating its robustness across different datasets and link functions. This reliability makes it a better candidate for practical predictive modelling, particularly when convergence is a concern. Therefore, Algorithm 1 performs well overall, particularly with the *half-power* LF, and offers consistent convergence in all cases. While it does not always achieve the lowest MSE, especially when compared with the *log* LF in certain years, its stability and competitive performance make it a reliable option for modelling flood insurance claims.

Focusing on extreme events recorded by the *National Center for Environmental Information (NCEI)*,[25] several significant flooding disasters impacted Texas, Louisiana, and Florida during the study period. In 2015, 2016, 2017, and 2019, extreme rainfall and subsequent flooding caused billions of dollars in losses in Texas and Louisiana. Notably, the 2016 Louisiana flood, a historic event, destroyed over 50,000 homes. Additionally, Hurricanes *Laura* and *Delta* in 2021 brought widespread damage to homes in Texas and Louisiana. These events provide context for interpreting the results in Table 6.

For Texas, Algorithm 1 generally performs well during years affected by extreme events, showing better overall predictability compared to the *log* and *half-power* LFs. However, exceptions exist. For instance, in **MATLAB** *fitglm* and **R** *glm*2, the *log* LF achieves the best performance for the predicted years 2015 and 2016, and the *half-power* LF performs better in 2016 and 2021. For Louisiana, the results reflect mixed performance during extreme events, with Algorithm 1 showing comparable predictive ability to other methods across many years. For Florida, NCEI records highlight events such as Hurricane *Ian* in 2022 and historical rainfall with flash flooding in 2023. In these years, Algorithm 1 outperforms all other methods in the predicted year 2022 and remains a strong choice for 2023 across most scenarios. These findings suggest that Algorithm 1 can adapt well to extreme conditions, particularly in Florida while maintaining competitive performance in Texas and Louisiana. Overall, the results indicate that Algorithm 1 is a reliable choice for predictive modelling, particularly in years with extreme flooding events.

7. Conclusions

This paper makes two important contributions to the GLM literature. First, we provide a general characterisation of proper GLMs for various exponential dispersion models, including the Tweedie family. The main finding is that although most Tweedie GLMs are not proper for *canonical* and *log* LFs, a rich class of proper Tweedie GLMs can be identified for *power* LFs. Second, we propose specialised optimisation algorithms for implementing several instances of Tweedie GLMs under *power* LFs. These algorithms outperform standard methods in terms of accuracy and efficiency, particularly in high-dimensional scenarios, as demonstrated via a thorough comparison with existing libraries like **MATLAB** *fitglm*, **R** *glm2*, and **Python** *sm.GLM*.

[25] Available at: https://www.ncei.noaa.gov/access/billions/.

CRediT authorship contribution statement

Vali Asimit: Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Alexandru Badescu:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Ziwei Chen:** Writing – review & editing, Investigation, Formal analysis. **Feng Zhou:** Writing – review & editing, Visualization, Software, Formal analysis.

Declaration of competing interest

There is no competing interest.

Appendix A. Proofs

A.1. Proof of Propositions 3.1-3.3

The proofs follow easily by verifying the conditions in Definition 2.1 for the Poisson, Gamma and Inverse Gaussian families, respectively.

A.2. Proof of Theorem 3.4

The identification of the three classes of Tweedie GLM that are well-defined is not difficult, and thus, we only outline some arguments without further details that are quite obvious. Clearly, $b'(\theta) = \big(\theta/(\alpha-1)\big)^{\alpha-1}$ for all $\theta \in \mathfrak{R}$. Since $\alpha < 2$, then setting a) is readily true and we require $\alpha \in (1,2)$, which is equivalent to $p < 0$, whenever $\Theta \in \{\mathfrak{R}_+^*, \mathfrak{R}_+\}$. Setting b) is the mirror case of setting a), and the proof is very similar. Settings c) and d) are similar to the previous ones, and the analysis depends if $\alpha - 1$ is an odd or even negative integer.

A.3. Proof of Theorem 3.5

First, we investigate parts (i) and (iii) (the *negative power* LF case) together, and therefore assume only *power* or *negative power* LFs. Condition **C2** requires

$$y(\alpha-1)\big(h(\eta)\big)^{\frac{1}{\alpha-1}} - \frac{\alpha-1}{\alpha}\big(h(\eta)\big)^{\frac{\alpha}{\alpha-1}} \text{ to be concave in } \eta \text{ on } \mathfrak{R} \text{ for all } y \in \mathcal{Y}. \tag{A.1}$$

Setting a) is first justified, but only for *power* LFs since the image of h is $\mathfrak{R}_+^*$, and in turn, $\gamma = 2k, k \in \mathbb{Z}^*$. Denote $a_1 = y(\alpha-1)$, $a_2 = \frac{1-\alpha}{\alpha}$ and $\gamma' = \frac{1}{\alpha-1}$. Equation (A.1) is equivalent to

$$\xi(\eta; y) := a_1\eta^{\gamma\gamma'} + a_2\eta^{\gamma(\gamma'+1)} \quad \text{is concave in } \eta \text{ on } \mathfrak{R} \text{ for all } y \in \mathcal{Y}. \tag{A.2}$$

Note that $\mathcal{Y} = \mathfrak{R}_+^*$ is assumed. Since $1 < \alpha < 2$ and $y > 0$, then $a_1 > 0$ and $a_2 < 0$, and in turn, (A.2) holds if and only if $\gamma\gamma' \in [0,1]$ and $\gamma(\gamma'+1) \notin (0,1)$. This is equivalent to having $\gamma \geq 0$, $\gamma\gamma' \leq 1$ and $\gamma(\gamma'+1) \geq 1$, since $\gamma' > 1$ in this case, which is further equivalent to $\frac{\alpha-1}{\alpha} \leq \gamma \leq \alpha - 1$. The latter cannot hold since $\alpha - 1 \in (0,1)$, $\alpha - 1 - \frac{\alpha-1}{\alpha} \in (0,1/2)$ and $\gamma = 2k, k \in \mathbb{Z}^*$, which concludes that no proper GLM model is possible for setting a).

Setting b) is now justified, but only for *power* LFs since the image of h is $\mathfrak{R}_+^*$, and thus, $\gamma = 2k, k \in \mathbb{Z}^*$. We split this in two subcases, setting b1) and setting b2) for $0 < \alpha < 1$ and $\alpha < 0$, respectively.

Setting b1) holds if and only if $\gamma\gamma' \notin (0,1)$ and $\gamma(\gamma'+1) \in [0,1]$, since $a_1 < 0$ and $a_2 > 0$, which is equivalent to having $\gamma \leq 0$, $\gamma\gamma' \geq 1$

and $\gamma(\gamma'+1) \geq 1$ as $\gamma' < -1$, and in turn, $\frac{\alpha-1}{\alpha} \leq \gamma \leq \alpha - 1$. The later is true if and only if $\gamma = -2k$ for any $k \in \mathbb{N}^*$ and $(1-\gamma)\alpha \leq 1$ since $0 < \alpha < 1$, which concludes setting b1).

Setting b2) implies that $a_1, a_2 < 0$ and $\gamma' \in (-1, 0)$. Therefore, setting b2) holds if and only if $\gamma\gamma' \notin (0,1)$ and $\gamma(\gamma'+1) \notin (0,1)$, which is equivalent to having $\gamma \geq 0$ and $\gamma(\gamma'+1) \geq 1$ or $\gamma \leq 0$ and $\gamma\gamma' \geq 1$, and in turn, $\frac{\alpha-1}{\alpha} \leq \gamma$ or $\gamma \leq \alpha - 1$ must hold, which concludes setting b2).

Setting c) is similar to setting b2), and we thus skip its proof. Setting d) requires for *power* and *negative power* LFs to having $\gamma' \in \mathbb{Z}$ so that the likelihood function is well-defined in (2.4) (and thus, in (A.1)), but also γ to be an odd integer so that the image of h is $\mathfrak{R}^*$. These do not hold since $\gamma' \in (-1, 0)$, which justifies our claim for setting d). This concludes parts (i) and (iii) (the *negative power* LF case).

The proof of parts (ii) and (iii) (the *negative half-power* LF case) follows in a similar way, with one small difference. That is, *half-power* LFs require $\gamma \in \mathfrak{R}^*$ instead of $\gamma = 2k, k \in \mathbb{Z}^*$, but everything else does not significantly change. For these reasons, we do not provide additional details on this proof.

A.4. Proof of Theorem 3.6

We first show part (i), and assume *canonical* LFs. Note first $h(\eta) = b'(\eta) = \left(\eta/(\alpha-1)\right)^{\alpha-1}$, which implies that $\alpha \in \mathbb{Z} \setminus \{1\}$. This implies that amongst settings a)–c), only setting b2), which was introduced in Appendix A.3, might hold while all other settings are clearly infeasible. The image of h is $\mathfrak{R}_+^*$ and therefore, α is an odd negative integer, which is a *power* LF with an odd parameter γ. This contradicts our findings in the proof of part (i) from Theorem 3.4 for setting b2), and concludes that no *canonical* LF leads to proper GLM in settings a)–c). Setting d) requires α to be an even negative integer and $\gamma' \in \mathbb{Z}$ as explained in the previous proof, which is infeasible conditions. Thus, no *canonical* LF leads to proper GLM in setting d). This concludes part (i).

We now show part (ii) and assume *log* LFs. Using the same notations as in Appendix A.3, Equation (A.1) is equivalent to

$$\xi(\eta; y) := a_1 e^{\eta\gamma'} + a_2 e^{\eta(\gamma'+1)} \quad \text{is concave in } \eta \text{ on } \mathfrak{R} \text{ for all } y \in \mathcal{Y}, \tag{A.3}$$

which requires $a_1, a_2 \leq 0$ due to the convexity property of $e^{\eta\gamma}$ in η on $\mathfrak{R}$, for any $\gamma \in \mathfrak{R}$. The latter explains that only setting b2) is feasible amongst settings a)–c). Setting d) is infeasible since the image of h is $\mathfrak{R}^*$, which is impossible for a *log* LF. The proof is now complete.

A.5. Proof of Theorem 4.2

We proceed by showing part a), but only for (4.1), since (4.2) could be argued similarly. Let

$$f_{i,P}(\boldsymbol{\beta}) = \left(\frac{1}{2}\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2 - y_i \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)\right) \quad \text{for all } 1 \leq i \leq n, \tag{A.4}$$

so that $f_P(\boldsymbol{\beta}) = \sum_{i=1}^n f_{i,P}(\boldsymbol{\beta})$. First, we show that f_P is a closed convex function on Ω. From (A.4), $f_{i,P}$ is convex (and therefore, continuous) on Ω, and since $dom\left(f_P\right) = \Omega$ is an open set and $\lim_{\boldsymbol{\beta} \to \boldsymbol{\beta}_0} f_{i,P}(\boldsymbol{\beta}) = \infty$ for all $\boldsymbol{\beta}_0 \in \partial\, dom\left(f_P\right)$, it follows that $f_{i,P}$ is closed convex on Ω. The closed convex property of f_P follows from the fact that it is a sum of closed convex functions.

We next prove that f_P is self-concordant on Ω. For any $t \in \mathfrak{R}$, $\boldsymbol{u} \in \Omega$ and $\boldsymbol{v} \in \mathfrak{R}^d$, such that $\boldsymbol{u} + t\boldsymbol{v} \in \Omega$, we define the function $g_{i,P}(t) = f_{i,P}(\boldsymbol{u} + t\boldsymbol{v})$, or any $i = 1, \ldots, n$, and let $g_P(t) = \sum_{i=1}^n g_{i,P}(t)$. Next, we show that

$$\left|g_{i,P}'''(t)\right| \leq 2\left(g_{i,P}''(t)\right)^{3/2}. \tag{A.5}$$

Note that

$$g_{i,P}''(t) = \left(\boldsymbol{x}_i^\top \boldsymbol{v}\right)^2 + \frac{y_i\left(\boldsymbol{x}_i^\top \boldsymbol{v}\right)^2}{\left(\boldsymbol{x}_i^\top \boldsymbol{u} + t\boldsymbol{x}_i^\top \boldsymbol{v}\right)^2} \quad \text{and} \quad g_{i,P}'''(t) = -\frac{2y_i\left(\boldsymbol{x}_i^\top \boldsymbol{v}\right)^3}{\left(\boldsymbol{x}_i^\top \boldsymbol{u} + t\boldsymbol{x}_i^\top \boldsymbol{v}\right)^3}.$$

Clearly, (A.5) holds whenever $\boldsymbol{x}_i^\top \boldsymbol{v} = 0$, and thus, we further assume that $\boldsymbol{x}_i^\top \boldsymbol{v} \neq 0$. Now,

$$\left|g_{i,P}'''(t)\right| \left(g_{i,P}''(t)\right)^{-3/2} = 2y_i\left(y_i + \left(\boldsymbol{x}_i^\top \boldsymbol{u} + t\boldsymbol{x}_i^\top \boldsymbol{v}\right)^2\right)^{-3/2} \leq 2,$$

since $y_i \leq y_i^{3/2} \leq \left(y_i + \epsilon_i\right)^{3/2}$ for any non-negative integer y_i and any $\epsilon_i \geq 0$ (recall that $y_i \in \mathbb{N}$ as the sampling distribution is Poisson). The self-concordant property of f_P follows from

$$\begin{aligned}\left|g_P'''(t)\right| &= \left|\sum_{i=1}^n g_{i,P}'''(t)\right| \leq \sum_{i=1}^n \left|g_{i,P}'''(t)\right| \leq 2\sum_{i=1}^n \left(g_{i,P}''(t)\right)^{3/2} \\ &\leq 2\left(\sum_{i=1}^n g_{i,P}''(t)\right)^{3/2} = 2\left(g_P''(t)\right)^{3/2}.\end{aligned}$$

Note that the first inequality follows from the triangle inequality, the second from (A.5), and the last one from the fact that the p-norm on $\mathfrak{R}^n$, $||\boldsymbol{x}||_p := \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$ is a decreasing function in p on $\mathfrak{R}_+^*$ for any $\mathbf{x} \in \mathfrak{R}^n$, and thus, $||\boldsymbol{x}||_1 \leq ||\boldsymbol{x}||_{2/3}$. This completes the proof for part a).

The proof of part b) follows in a similar way, and thus, we only provide the main steps. As before, we only show (4.3) since its proof is very similar to the proof of (4.4). We denote

$$f_{i,G}(\boldsymbol{\beta}) = \left(\frac{y_i}{2}\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2 - \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)\right) \quad \text{for all } 1 \leq i \leq n,$$

so that $f_G(\boldsymbol{\beta}) = \sum_{i=1}^n f_{i,G}(\boldsymbol{\beta})$. Following the same arguments as in part a), we may show that f_G is a closed convex function on Ω. The proof that f_G is self-concordant on Ω follows in a similar way by defining the function $g_{i,G}(t) = f_{i,G}(\boldsymbol{u} + t\boldsymbol{v})$ and $g_G(t) = \sum_{i=1}^n g_{i,G}(t)$ for any $t \in \mathfrak{R}$, $\boldsymbol{u} \in \Omega$ and $\boldsymbol{v} \in \mathfrak{R}^d$, such that $\boldsymbol{u} + t\boldsymbol{v} \in \Omega$, and showing that $\left|g_{i,G}'''(t)\right| \leq 2\left(g_{i,G}''(t)\right)^{3/2}$. The second and third order derivatives of $g_{i,G}$ are given by

$$g_{i,G}''(t) = y_i\left(\boldsymbol{x}_i^\top \boldsymbol{v}\right)^2 + \frac{\left(\boldsymbol{x}_i^\top \boldsymbol{v}\right)^2}{\left(\boldsymbol{x}_i^\top \boldsymbol{u} + t\boldsymbol{x}_i^\top \boldsymbol{v}\right)^2} \quad \text{and} \quad g_{i,G}'''(t) = -\frac{2\left(\boldsymbol{x}_i^\top \boldsymbol{v}\right)^3}{\left(\boldsymbol{x}_i^\top \boldsymbol{u} + t\boldsymbol{x}_i^\top \boldsymbol{v}\right)^3}.$$

Clearly, the required inequality holds if $\boldsymbol{x}_i^\top \boldsymbol{v} = 0$, and thus, $\boldsymbol{x}_i^\top \boldsymbol{v} \neq 0$ is further assumed. Now,

$$\left|g_{i,G}'''(t)\right| \left(g_{i,G}''(t)\right)^{-3/2} = 2\left(y_i\left(\boldsymbol{x}_i^\top \boldsymbol{u} + t\boldsymbol{x}_i^\top \boldsymbol{v}\right)^2 + 1\right)^{-3/2} \leq 2,$$

since $\left(1 + y_i\epsilon_i\right)^{-3/2} \leq 1$ for any $y_i > 0$ and $\epsilon_i \geq 0$ (recall that $y_i \in \mathfrak{R}_+^*$ as the sampling distribution is Gamma). This completes the proof.

Appendix B. Other special cases of GLMs

B.1. Linear regression – Gaussian family

Assume that $Y \sim N(\theta, \phi^2)$ with probability distribution function given by

$$\log\left(f_Y(y; \theta, \phi)\right) = \frac{\theta y - \frac{\theta^2}{2}}{\phi} - \frac{1}{2}\left(\frac{y^2}{\phi} + \log(2\pi\phi)\right),$$
$$(y, \theta, \phi) \in \mathfrak{R} \times \mathfrak{R} \times \mathfrak{R}_+^*.$$

The above pdf is obtained as a special case of (2.1) by taking

$$a(\phi) = \phi, \quad b(\theta) = \frac{\theta^2}{2}, \quad c(y, \phi) = -\frac{1}{2}\left(\frac{y^2}{\phi} + \log(2\pi\phi)\right).$$

In addition, $b'(\Theta)=\mathfrak{R}$ and $b'^{-1}(\mu)=\mu$. Proposition Appendix B.1 provides a characterisation of the LFs under which the Gaussian GLM is properly defined according to Definition 2.1.

Proposition Appendix B.1. *Assume that $Y\sim N(\theta,\phi^2)$. The Gaussian GLM is proper if and only if $h:\mathfrak{R}\to\mathfrak{R}$ and*

$$-yh(\eta)+\frac{h^2(\eta)}{2} \quad \text{is convex in } \eta \text{ on } \mathfrak{R} \text{ for any given } y\in\mathfrak{R}. \tag{B.1}$$

Proof. The proof follows from verifying conditions **C1** and **C2** from Definition 2.1. □

Corollary Appendix B.2 identifies the only class of LFs which satisfies Equation (B.1).

Corollary Appendix B.2. *The Gaussian GLM is proper if and only if the LF is linear.*

Proof. Since any convex real function defined on a finite open set I is continuous with non-decreasing left (and right) derivatives, then (B.1) implies that

$$h'_+(\eta_1)\,h(\eta_1)-yh'_+(\eta_1)\le h'_+(\eta_2)\,h(\eta_2)-yh'_+(\eta_2) \quad \text{for all } y\in\mathfrak{R}, \tag{B.2}$$

and any reals $\eta_1<\eta_2$ from I, where h'_+ is the right derivative of h. Assume now that h is not linear on $\mathfrak{R}$, and thus, not linear on I. Then, there exists $\eta_1<\eta_2$ from I such that $h'_+(\eta_2)-h'_+(\eta_1)\neq 0$. The latter contradicts (B.2), and in turn, we must have h linear on $\mathfrak{R}$, and no other possible LF leads to a MLE-based Gaussian GLM. □

The *canonical* LF for Gaussian GLMs is the *identity* function. Corollary Appendix B.2 implies the *canonical* LF leads to a proper GLM and it is the only *power* function with this property.

B.2. Logistic Regression – Bernoulli family

Assume that $Y\sim Bernoulli(\theta)$ with probability mass function given by

$$\log\left(f_Y(y;\theta,\phi)\right)=\theta y-\log\left(1+e^{\theta}\right) \quad \text{with } (y,\theta,\phi)\in\{0,1\}\times\mathfrak{R}\times\{1\}.$$

The above function is obtained as a special case of (2.1) by taking

$$a(\phi)=1,\quad b(\theta)=\log\left(1+e^{\theta}\right),\quad c(y,\phi)=0.$$

In addition, $b'(\Theta)=(0,1)$ and $b'^{-1}(\mu)=\log\frac{\mu}{1-\mu}$. Proposition Appendix B.3 provides a brief characterisation of a proper Logistic regression model.

Proposition Appendix B.3. *Assume that $Y\sim Bernoulli(\theta)$. The Bernoulli GLM is proper if and only if $h:\mathfrak{R}\to(0,1)$, and*

$$y\log(h(\eta))+(1-y)\log(1-h(\eta))$$
$$\text{is concave in } \eta \text{ on } \mathfrak{R} \text{ for any given } y=\{0,1\}. \tag{B.3}$$

Proof. The proof follows easily by verifying the conditions **C1** and **C2** from Definition 2.1. □

A direct consequence of the above is that the MLE-based Bernoulli GLM is proper if and only if $h(\eta)$ and $h(1-\eta)$ are log-concave functions[26] on $\mathfrak{R}$. Three standard choices for h have been proposed for this family in the literature, and all of them lead to proper GLMs:

(i) *logit* LF, which corresponds to having $h(\eta)=\frac{1}{1+e^{-\eta}}$, which is also the Bernoulli *canonical* LF that satisfies the conditions in Proposition 2.2 since b is strictly convex on $\mathfrak{R}$.
(ii) *probit* LF, which corresponds to having $h(\eta)=\Phi(\eta)$, where Φ is the cdf of a standard Gaussian random variable. In this case, it is not difficult to show that h satisfies the characterisation from Proposition Appendix B.3.
(iii) *complementary log-log* LF, which corresponds to having $h(\eta)=1-\exp(-\exp(-\eta))$. It is not difficult to show that h satisfies the conditions in Proposition Appendix B.3.

Finally, it is clear that no power LF satisfies the conditions in Proposition Appendix B.3.

Appendix C. Data generation process

This section briefly outlines the DGPs for the Poisson, Gamma and Inverse Gaussian GLMs.

- **Step 1**: Generate the matrix of covariates $\boldsymbol{X}=\{X_{i,j}\}_{i=1,j=1}^{n,d}$, from a Gaussian distribution with mean μ and unit standard deviation, $X_{i,j}\sim\mathcal{N}(\mu,1)$. Note that for each GLM, we let μ to be a function of d, such that the expected value of the response variable is within reasonable bounds in order to avoid exaggerating the parameter values when generating $\boldsymbol{Y}$ in Step 3, which typically affect the estimation procedure for the benchmarks.[27]
- **Step 2**: Generate the regression coefficient $\boldsymbol{\beta}=\{\beta_j\}_{j=0}^{d}$ by setting $\beta_j=j/d$.
- **Step 3**: For any $i=1,\dots,n$, let $\theta_i=\beta_0+\sum_{j=1}^{d}\beta_j x_{i,j}$ and generate the response variable $\boldsymbol{Y}=\{Y_i\}_{i=1}^{n}$ by simulating each Y_i from $Poisson\left(\theta_i^2\right)$ for the Poisson GLM, $Gamma\left(\theta_i^2,1\right)$ for the Gamma GLM and $IG\left(\theta_i^{-2},1\right)$ for the Inverse Gaussian GLM.

Appendix D. Deviance for Poisson, Gamma and Inverse Gaussian GLMs

- Poisson GLM with *half-square-root* LF

$$D\left(\hat{\boldsymbol{\beta}}\right)=\sum_{i=1}^{n}\left(4y_i\log\left(\frac{\sqrt{y_i}}{\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}}\right)+2\left(\left(\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}\right)^2-y_i\right)\right)\cdot I_{\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}>0}$$
$$+0\cdot I_{\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}=y_i=0}+\infty\cdot I_{else}$$

- Gamma GLM with *half-reciprocal-square-root* LF

$$D\left(\hat{\boldsymbol{\beta}}\right)=\sum_{i=1}^{n}\left(2y_i\left(\left(\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}\right)^2-y_i^{-1}\right)-2\log\left(y_i\right)\right.$$
$$\left.-4\log\left(\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}\right)\right)\cdot I_{\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}>0}+\infty\cdot I_{\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}\le 0}$$

- Inverse Gaussian GLM with *reciprocal-square root* LF

$$D\left(\hat{\boldsymbol{\beta}}\right)=\phi^2\sum_{i=1}^{n}\left(y_i\left(\left(\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}\right)^4-y_i^{-2}\right)-2\left(\left(\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}\right)^2-y_i^{-1}\right)\right)$$
$$\cdot I_{\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}>0}+\infty\cdot I_{\boldsymbol{x}_i^{\top}\hat{\boldsymbol{\beta}}\le 0}$$

[26] A function $f:A\to B$ is log-concave on A if $\log(f(\alpha x+(1-\alpha)y))\ge\alpha\log(f(x))+(1-\alpha)\log(f(y))$ for all $x,y\in A$ and $0<\alpha<1$.

[27] Note that in such cases all standard benchmarks fail to converge in most scenarios.

Table E.7
Summary Statistics of Raw Health Insurance Data.

Variable	Obs.	Min	Max	Median	Mean	Std.Dev.
Medical costs (charges, USD)	1,338	1,122	63,770	9,382	13,270	12,110
Age	1,338	18	64	39	39	14
Sex (female: 662 vs. male: 676)	1,338	N/A	N/A	N/A	N/A	N/A
BMI	1,338	15.96	53.13	30.4	30.66	6.1
No. of Children	1,338	0	5	1	1.1	1.21
Smoker (yes: 274 vs. no: 1,064)	1,338	N/A	N/A	N/A	N/A	N/A
Region	1,338	N/A	N/A	N/A	N/A	N/A

Notes: This table summarises the variables in the U.S. health insurance dataset. The dataset includes 1,338 observations. Character variables such as *Sex*, *Smoker*, and *Region* are shown with counts. For *Region*, NE refers to the Northeast (324 observations), NW refers to the Northwest (325 observations), SE refers to the Southeast (364 observations), and SW refers to the Southwest (325 observations). Numeric variables include medical costs (*charges*), *age*, *BMI*, and the *number of children*, summarised by their minimum, maximum, median, mean, and standard deviation. Character variables are listed as "N/A" for numerical summaries.

Table E.8
Summary of Data Preprocessing for Health Insurance Dataset.

Variable	Type	Regrouped	Bounded	One-hot Encoded	Resulting Columns
Charges	Numeric	Yes	No	Yes	4
Age	Integer	Yes	No	Yes	6
Sex	Character	No	No	Yes	2
BMI	Numeric	No	No	No	N/A
Children	Integer	No	Yes	Yes	N/A
Smoker	Character	No	No	Yes	2
Region	Character	No	No	Yes	4

Notes: This table summarises the processing steps applied to the U.S. health insurance dataset. The *Charges* variable, used as the dependent variable, was regrouped into four categories: (1) Low ($\leq$ 10,000 USD), (2) Mid (10,001 – 20,000 USD), (3) Upper-Mid (20,001 – 40,000 USD), and (4) High ($>$ 40,000 USD). The remaining variables were used as features in the analysis. For *Age*, values were regrouped into six categories: (1) Ages 18-21, (2) Ages 22-26, (3) Ages 27-35, (4) Ages 36-45, (5) Ages 46-55, and (6) Ages 56-64. The *Children* variable was capped at 3, meaning any number greater than 3 was set to 3. The *BMI* variable underwent a logarithmic transformation to reduce skewness. Binary variables such as *Smoker* and *Sex* were encoded into two categories. The *Region* variable, which includes four geographic areas (Northeast, Northwest, Southeast, and Southwest), was one-hot encoded. The *Resulting Columns* column indicates the number of columns created after one-hot encoding for character variables. For numeric variables (*BMI* and *Children*), "N/A" is shown under the *Resulting Columns* column, as they were not transformed into additional columns during preprocessing.

Appendix E. Data description

This section describes the two real-world datasets used in the analysis: the U.S. health insurance dataset and the FEMA NFIP claims dataset. The first focuses on medical costs and related demographic and lifestyle factors, while the second provides information on flood insurance claims from 2010 to 2023 in Florida, Texas, and Louisiana. Both datasets are preprocessed to ensure the variables are suitable for statistical modelling, including transformations, binning, and encoding as needed. Summary statistics and visualisations are provided to give an overview of the datasets and highlight key features. The preprocessing steps and descriptive analyses ensure the data is prepared for evaluating the performance of Algorithm 1.

E.1. U.S. health insurance dataset

The *first* U.S. health insurance dataset[28] contains 1,338 observations and is widely used in the machine learning research community to study the relationships between medical costs and various demographic and lifestyle factors. The dataset includes one dependent variable, *Medical Costs (Charges)*, and six independent variables: *Age, Sex, BMI, Number of Children, Smoker*, and *Region*. Table E.7 provides summary statistics for the raw dataset, which highlights the distribution of the numerical variables and the counts for categorical variables. The dependent variable, *Charges*, exhibits a wide range from 1,122 to 63,770 USD with a mean of 13,270 USD, showing a right-skewed distribution.

To prepare the dataset for analysis, preprocessing steps were applied to both dependent and independent variables, as summarised in Table E.8. The *Charges* variable, representing medical costs, was regrouped into four categories: Low ($\leq$ 10,000 USD), Mid (10,001 – 20,000 USD), Upper-Mid (20,001 – 40,000 USD), and High ($>$ 40,000 USD). Binning the *Charges* variable simplifies the interpretation of medical costs and makes it suitable for modelling with a Poisson GLM. For *Age*, values were grouped into six categories based on predefined age ranges to facilitate modelling. The *BMI* variable underwent a logarithmic transformation to reduce skewness and improve its interpretability. Binary variables, such as *Sex* (Male or Female) and *Smoker* (Yes or No), were one-hot encoded. The *Region* variable was also one-hot encoded to represent four geographic areas: Northeast, Northwest, Southeast, and Southwest. Table E.9 presents the cleaned dataset, grouped by binned *Charges*, with numerical averages and distributions of categorical variables. Similar preprocessing techniques, including binning age groups and applying

[28] "U.S. Health Insurance" dataset is available at https://github.com/stedy/Machine-Learning-with-R-datasets or www.kaggle.com/datasets/teertha/ushealthinsurancedataset.

Table E.9
Summary Statistics of Cleaned Health Insurance Data by Binned Charges.

Charges_binned	Low	Mid	Upper-Mid	High
Avg_charges	5,207.23	13,513.71	29,222.34	45,681.02
Avg_age	33.09	49.03	39.90	48.75
Avg_log(BMI)	3.39	3.39	3.39	3.60
Sex	**F:** 355, **M:** 357	**F:** 192, **M:** 161	**F:** 83, **M:** 111	**F:** 32, **M:** 47
Smoker	**N:** 712, **Y:** 0	**N:** 291, **Y:** 62	**N:** 61, **Y:** 133	**N:** 0, **Y:** 79
Children Count	**0:** 292, **1:** 201, **2:** 126, **3:** 93	**0:** 174, **1:** 59, **2:** 53, **3:** 67	**0:** 82, **1:** 46, **2:** 40, **3:** 26	**0:** 26, **1:** 18, **2:** 21, **3:** 14
Age_binned	**1:** 140, **2:** 100, **3:** 168 **4:** 190, **5:** 114, **6:** 0	**1:** 27, **2:** 16, **3:** 36 **4:** 19, **5:** 93, **6:** 162	**1:** 27, **2:** 21, **3:** 30 **4:** 36, **5:** 50, **6:** 30	**1:** 0, **2:** 3, **3:** 6 **4:** 19, **5:** 27, **6:** 24
Region	**NE:** 161, **NW:** 176, **SE:** 193, **SW:** 182	**NE:** 95, **NW:** 88, **SE:** 83, **SW:** 87	**NE:** 53, **NW:** 47, **SE:** 52, **SW:** 42	**NE:** 15, **NW:** 14, **SE:** 36, **SW:** 14

Notes: This table summarises the U.S. health insurance dataset grouped by binned charges. *Avg_charges*, *Avg_age*, and *Avg_log(BMI)* represent the averages for numerical variables. The distributions of Character variables are provided for *Sex*, *Smoker*, *Age_binned*, and *Region*. For *Sex*, F indicates Female, and M indicates Male. For *Smoker*, Y indicates Yes (smoker) and N indicates No (non-smoker). While *Children* is treated as a numeric variable, it is displayed in the table as counts for 0, 1, 2, and capped at 3 children to enhance clarity. *Age_binned* represents grouped age ranges labelled from 1 to 6. *Region* refers to geographic locations, with NE for Northeast, NW for Northwest, SE for Southeast, and SW for Southwest.

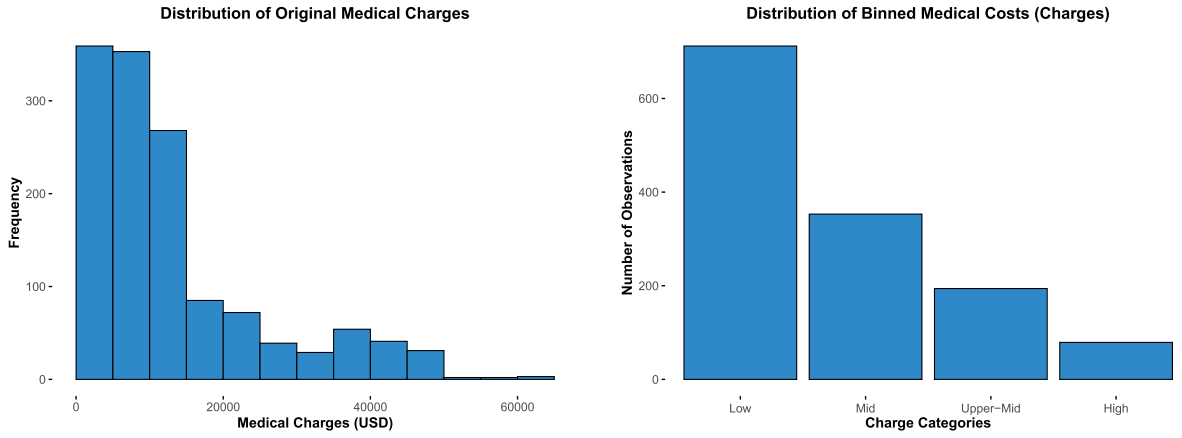

Fig. E.3. Distribution of Medical Costs (Charges).
Notes: The left plot shows the histogram of the dependent variable, medical charges, which exhibits a right-skewed distribution due to a few high-cost outliers. The right plot depicts the histogram of the same variable after binning into four categories for analysis: Low ($\leq$ 10,000 USD), Mid (10,001 – 20,000 USD), Upper-Mid (20,001 – 40,000 USD), and High ($>$ 40,000 USD). Binning simplifies the interpretation of the dependent variable by grouping observations into discrete categories, which are used in subsequent modelling and analysis. These plots represent data from the U.S. health insurance dataset used in the study.

logarithmic transformations to handle skewness, can be found in the analysis of the *French Motor Third-Party Liability Claims* dataset.[29]

Fig. E.3 visualises the dependent variable, *Charges*, with two histograms. The left plot shows the original distribution, which is right-skewed due to high-cost outliers, while the right plot depicts the binned *Charges* distribution categorised into four levels. These preprocessing steps ensure the dataset is structured and ready for a 70% training and 30% testing split to evaluate the out-of-sample performance of Algorithm 1 and compare it to other methods implemented in **Matlab**, **Python**, and **R**.

E.2. Flood insurance dataset

The *second* FEMA NFIP Claims dataset[30] includes flood insurance claim records from the NFIP. The data provide information on flood-related claims across the United States, with sensitive details redacted to protect policyholders. We focus on claim data from 2010 to 2023 for three states: Florida, Texas, and Louisiana. These states are particularly vulnerable to flood-related losses and were selected due to their prominence in flood insurance claims. The dataset captures financial, structural, and geographical attributes of flood claims, making it suitable for analysing coverage efficiency and model-based predictions.

[29] Tutorials for similar preprocessing steps can be accessed at https://github.com/actuarial-data-science/Tutorials.

[30] "OpenFEMA" dataset is available at www.fema.gov/openfema-data-page/fima-nfip-redacted-claims-v2.

Table E.10
Summary of Data Preprocessing for Flood Insurance Dataset.

Variable	Type	Regrouped	Bounded	One-hot Encoded	Resulting Columns
ratioCoverage	Numeric	No	Yes	No	N/A
amountPaidOnBuildingClaim	Numeric	No	Yes	No	N/A
totalBuildingInsuranceCoverage	Integer	No	Yes	No	N/A
buildingPropertyValue	Numeric	No	Yes	No	N/A
buildingDamageAmount	Integer	No	Yes	No	N/A
numberOfFloorsInTheInsuredBuilding	Integer	No	No	No	N/A
waterDepth	Integer	No	Yes	No	N/A
buildingDeductibleCode	Character	No	Yes	No	N/A
elevatedBuildingIndicator	Integer	No	No	Yes	2
postFIRMConstructionIndicator	Character	No	No	Yes	2
ratedFloodZone	Character	Yes	No	Yes	2
buildingDescriptionCode	Integer	Yes	No	Yes	2
originalConstructionDate	Character	Yes	No	Yes	3
replacementCostBasis	Character	Yes	No	Yes	2
causeOfDamage	Character	Yes	No	Yes	2
latitude	Numeric	Yes	No	Yes	4
longitude	Numeric	Yes	No	Yes	4

Notes: This table summarises the dependent variable and features used in the flood insurance dataset. The dependent variable, *ratioCoverage*, is calculated as the ratio of *amountPaidOnBuildingClaim* to *totalBuildingInsuranceCoverage*, representing the proportion of insurance coverage paid for building claims. To mitigate the effect of outliers, *ratioCoverage* was capped at the 0.99 quantile. Features include all variables except *amountPaidOnBuildingClaim*. Variables such as *totalBuildingInsuranceCoverage*, *buildingPropertyValue*, *buildingDamageAmount*, and *buildingDeductibleCode* were bounded to be strictly greater than 0 to avoid numerical issues during logarithmic transformation and to ensure meaningful values. Specifically, *buildingDeductibleCode* was transformed from its original character representation into numerical amounts based on descriptions provided in the *FEMA NFIP Claims* dataset. *waterDepth* was capped at the 0.99 quantile to handle outliers and was lower bounded by 0 (including 0). *latitude* and *longitude* were used together and grouped into clusters to represent the location of buildings. Binary variables, including *replacementCostBasis*, *elevatedBuildingIndicator*, and *postFIRMConstructionIndicator*, were retained in their original format. Categorical variables such as *ratedFloodZone*, *buildingDescriptionCode*, and *causeOfDamage* were simplified into two groups based on the meaning being described to avoid rare categories that may not appear in all years. *originalConstructionDate* was converted to *originalConstructionYears* to calculate building age and grouped into three categories. The *Resulting Columns* column indicates the number of columns created after one-hot encoding or clustering. For numeric variables that were not transformed into additional columns, "N/A" is shown to indicate no additional columns were created.

The dependent variable, *ratioCoverage*, is defined as the ratio of *amountPaidOnBuildingClaim* to *totalBuildingInsuranceCoverage*. This variable represents the proportion of coverage utilised in claims and is used to evaluate the efficiency and adequacy of flood insurance coverage. To ensure meaningful values and reduce the impact of outliers, *ratioCoverage* was capped at the 0.99 quantile. In addition to this dependent variable, the dataset includes 15 features: *totalBuildingInsuranceCoverage*, *buildingPropertyValue*, *buildingDamageAmount*, *numberOfFloorsInTheInsuredBuilding*, *waterDepth*, *buildingDeductibleCode*, *elevatedBuildingIndicator*, *postFIRMConstructionIndicator*, *ratedFloodZone*, *buildingDescriptionCode*, *originalConstructionDate*, *replacementCostBasis*, *causeOfDamage*, *latitude*, and *longitude*. These features reflect various aspects of the claims, including financial details, building characteristics, and geographical location. Table E.10 outlines the features, their types, and the preprocessing steps applied, which include bounding numerical variables, simplifying categorical variables, and encoding binary variables as needed.

Tables E.11, E.12, and E.13 present the annual and overall summary statistics for *amountPaidOnBuildingClaim*, *totalBuildingInsuranceCoverage*, and *ratioCoverage*, respectively, across Florida, Texas, and Louisiana. They also include the number of observations, minimum, maximum, median, mean, and standard deviation for each variable, which helps to understand variability and trends over time. For example, *amountPaidOnBuildingClaim* (Table E.11) underscores variations in flood-related claims between states and across years, while *totalBuildingInsuranceCoverage* (Table E.12) shows differences in insured amounts. *ratioCoverage* (Table E.13) summarises the proportion of claims relative to coverage, illustrating patterns in insurance utilisation.

Figs. E.4, E.5, and E.6 provide visualisations of the data for Florida, Texas, and Louisiana, respectively. These include histograms of *amountPaidOnBuildingClaim*, *totalBuildingInsuranceCoverage*, and *ratioCoverage*. Logarithmic transformations of financial variables are included to address skewness and enhance interpretability. The *ratioCoverage* histograms are also overlaid with a Gamma distribution fit, illustrating its suitability for modelling purposes.

These preprocessing steps ensure that the extracted data from 2010 to 2023 are ready for an out-of-sample testing approach to evaluate the forecasting performance of different methods. Specifically, each year's dataset is used as a test set, and the data from the previous year is used as a training set. This approach exploits temporal patterns while avoiding data leakage. The main goal of the analysis is to compare the performance of Algorithm 1 with alternative methods implemented in **Matlab**, **Python**, and **R**, focusing on the ability to predict *ratioCoverage* using the selected features.

Table E.11
Summary Statistics for Flood Insurance Dataset (Building Claim).

Year	Obs.	Min	Max	Median	Mean	Std.Dev.
Florida (FL)						
2010	241	265.50	151,336.40	6,275.89	14,310.32	20,469.76
2011	1,027	19.13	483,061.70	8,099.52	17,327.53	31,953.73
2012	2,741	0.20	494,292.70	8,836.92	19,712.75	34,265.14
2013	1,101	31.58	1,070,279.00	13,373.94	22,732.65	43,857.98
2014	2,352	10.65	1,649,712.00	24,377.75	36,088.53	54,672.15
2015	1,080	45.04	433,906.00	10,214.19	20,090.91	30,566.75
2016	5,044	44.64	1,011,369.00	26,107.08	39,331.90	47,977.24
2017	16,216	3.88	2,395,029.00	20,934.88	42,705.43	67,122.00
2018	2,864	46.06	4,504,105.00	27,212.72	52,764.74	120,814.80
2019	492	41.66	868,788.00	17,669.28	40,740.05	73,328.56
2020	6,322	24.13	908,923.50	21,511.22	39,308.67	52,624.37
2021	384	386.67	248,750.00	18,294.27	34,779.68	41,193.29
2022	31,061	2.68	9,899,055.00	69,509.74	105,434.20	219,265.70
2023	7,754	34.25	1,084,439.00	61,062.64	69,173.08	66,755.06
All Years	78,679	0.20	9,899,055.00	35,001.34	67,897.07	150,038.60
Texas (TX)						
2010	1,642	8.60	357,937.80	13,903.04	28,785.46	37,879.04
2011	82	11.53	131,565.20	6,346.45	15,411.13	24,837.49
2012	1,336	25.81	244,527.90	10,579.21	19,815.68	24,443.59
2013	1,241	23.68	230,043.00	24,929.10	35,464.10	37,020.18
2014	654	102.38	287,634.50	9,395.23	16,695.70	22,481.69
2015	8,972	12.69	2,250,779.00	25,606.37	43,018.55	58,203.41
2016	10,509	26.62	480,806.20	35,234.87	47,613.32	48,642.23
2017	61,229	5.68	3,718,845.00	66,255.77	78,802.76	68,702.77
2018	2,091	4.00	490,812.20	25,102.79	39,657.50	44,727.73
2019	8,998	18.70	515,975.00	48,697.10	56,777.53	48,429.79
2020	1,191	9.34	361,616.30	15,578.69	26,336.46	31,344.37
2021	1,237	153.67	428,758.30	17,248.87	29,623.56	38,413.29
2022	346	78.30	259,506.50	21,284.90	36,452.89	42,293.05
2023	285	200.29	339,217.60	22,280.11	37,757.51	47,990.23
All Years	99,813	4.00	3,718,845.00	51,192.40	65,386.98	64,221.07
Louisiana (LA)						
2010	177	269.71	186,889.00	6,663.98	12,771.22	18,402.82
2011	1,789	6.21	225,524.70	10,635.26	18,323.00	24,697.03
2012	9,195	4.80	454,036.50	23,601.94	34,873.10	38,161.29
2013	702	45.16	166,523.50	10,805.19	20,788.63	23,292.28
2014	486	194.17	209,857.40	18,746.75	27,304.86	27,819.45
2015	488	80.82	213,694.10	10,720.24	20,341.93	27,430.32
2016	26,704	9.54	692,480.00	66,080.45	71,927.44	52,991.48
2017	1,741	21.75	361,308.00	20,654.59	30,409.53	34,754.57
2018	346	89.23	215,838.50	18,381.26	26,296.47	28,477.97
2019	2,031	133.29	433,805.00	24,241.07	35,923.32	41,337.63
2020	2,737	8.03	434,323.20	27,130.69	39,524.83	42,882.96
2021	11,615	14.68	499,648.60	51,518.71	58,998.37	52,799.65
2022	123	804.05	156,646.90	13,893.35	25,412.10	31,172.44
2023	109	681.94	476,322.80	19,225.69	35,527.84	61,953.50
All Years	58,243	4.80	692,480.00	47,332.86	55,795.29	51,312.22

Notes: This table summarises the annual statistics for the feature *amountPaidOnBuildingClaim* across three states (Florida, Texas, and Louisiana). The statistics are calculated for each year and state (Florida, Texas, and Louisiana) and include the number of observations, minimum, maximum, median, mean, and standard deviation. The "All Years" row aggregates the data across all years (2010-2023) to provide overall summary statistics for each state. The data provides insights into the variability and distribution of building claim payments over time and across states.

Table E.12
Summary Statistics for Flood Insurance Dataset (Building Coverage).

Year	Obs.	Min	Max	Median	Mean	Std.Dev.
Florida (FL)						
2010	241	18,400.00	90,967,200.00	200,000.00	1,092,283.00	7,211,990.00
2011	1,027	15,000.00	116,000,000.00	250,000.00	747,523.50	5,625,071.00
2012	2,741	4,900.00	93,250,000.00	198,400.00	419,302.80	3,164,952.00
2013	1,101	12,000.00	244,000,000.00	227,900.00	2,293,090.00	12,477,770.00
2014	2,352	5,500.00	142,000,000.00	240,000.00	633,667.90	4,383,651.00
2015	1,080	8,700.00	244,000,000.00	204,700.00	923,395.40	9,691,769.00
2016	5,044	5,000.00	61,000,000.00	250,000.00	370,075.90	1,967,997.00
2017	16,216	2,700.00	123,000,000.00	250,000.00	552,023.80	3,636,400.00
2018	2,864	10,000.00	64,000,000.00	250,000.00	307,067.80	1,583,612.00
2019	492	17,700.00	244,000,000.00	250,000.00	4,288,565.00	19,228,059.00
2020	6,322	5,500.00	58,250,000.00	250,000.00	369,857.60	1,723,522.00
2021	384	17,700.00	114,000,000.00	240,000.00	801,724.20	6,515,546.00
2022	31,061	3,600.00	63,750,000.00	250,000.00	493,944.60	1,844,845.00
2023	7,754	4,000.00	244,000,000.00	250,000.00	346,802.90	3,204,194.00
All Years	78,679	2,700.00	244,000,000.00	250,000.00	529,721.10	3,643,900.00
Texas (TX)						
2010	1,642	7,400.00	6,442,000.00	150,000.00	172,664.50	244,207.70
2011	82	20,000.00	500,000.00	164,800.00	179,298.80	98,285.68
2012	1,336	10,000.00	7,247,400.00	177,900.00	185,815.90	249,984.40
2013	1,241	16,500.00	20,500,000.00	150,000.00	208,194.20	835,324.90
2014	654	12,100.00	32,524,800.00	150,000.00	261,836.70	1,532,190.00
2015	8,972	2,800.00	29,250,000.00	191,900.00	205,652.40	654,214.20
2016	10,509	3,500.00	26,000,000.00	193,600.00	192,483.20	336,868.60
2017	61,229	2,300.00	99,809,800.00	227,600.00	216,316.20	584,933.90
2018	2,091	5,500.00	6,797,800.00	200,000.00	199,071.40	279,317.40
2019	8,998	6,000.00	7,477,600.00	200,000.00	187,718.70	127,481.70
2020	1,191	5,100.00	9,949,300.00	150,000.00	194,172.20	451,069.50
2021	1,237	4,500.00	4,126,100.00	200,000.00	200,803.20	200,332.00
2022	346	19,000.00	1,750,000.00	196,500.00	195,624.60	133,564.20
2023	285	12,000.00	500,000.00	250,000.00	211,410.50	82,326.96
All Years	99,813	2,300.00	99,809,800.00	200,000.00	208,407.40	540,750.00
Louisiana (LA)						
2010	177	6,500.00	500,000.00	118,100.00	144,411.90	104,203.10
2011	1,789	5,500.00	21,072,500.00	118,600.00	143,834.30	503,543.70
2012	9,195	2,000.00	13,310,000.00	175,000.00	185,403.80	273,969.90
2013	702	1,000.00	500,000.00	125,000.00	129,675.50	79,398.39
2014	486	10,000.00	500,000.00	150,000.00	163,741.60	81,913.65
2015	488	6,500.00	500,000.00	125,000.00	148,917.00	98,669.01
2016	26,704	1,000.00	4,341,700.00	170,000.00	176,025.70	85,021.68
2017	1,741	6,500.00	1,625,000.00	153,100.00	178,626.40	115,088.40
2018	346	8,800.00	500,000.00	143,000.00	159,257.20	108,916.30
2019	2,031	2,900.00	21,072,500.00	200,000.00	230,164.90	668,780.60
2020	2,737	1,800.00	3,250,000.00	170,000.00	179,188.80	119,142.70
2021	11,615	2,200.00	3,250,000.00	203,500.00	203,445.70	98,007.40
2022	123	20,400.00	500,000.00	200,000.00	191,066.70	84,045.23
2023	109	33,000.00	500,000.00	250,000.00	227,714.70	108,870.80
All Years	58,243	1,000.00	21,072,500.00	177,100.00	183,144.40	205,325.90

Notes: This table summarises the annual statistics for the feature *totalBuildingInsuranceCoverage* across three states (Florida, Texas, and Louisiana). The statistics are calculated for each year and state (Florida, Texas, and Louisiana) and include the number of observations, minimum, maximum, median, mean, and standard deviation. The "All Years" row aggregates the data across all years (2010-2023) to provide overall summary statistics for each state. These values reflect the distribution and variation in total insurance coverage amounts for building over time and across states.

Table E.13
Summary Statistics for Flood Insurance Dataset (Building Coverage Ratio).

Year	Obs.	Min	Max	Median	Mean	Std.Dev.
Florida (FL)						
2010	241	0.0005	0.9267	0.0327	0.0795	0.1195
2011	1,027	0.0000	0.9661	0.0393	0.0828	0.1051
2012	2,741	0.0000	0.9963	0.0509	0.1193	0.1599
2013	1,101	0.0001	0.9696	0.0654	0.1079	0.1304
2014	2,352	0.0001	0.9951	0.1284	0.1853	0.1890
2015	1,080	0.0000	0.8852	0.0616	0.1125	0.1386
2016	5,044	0.0002	0.9929	0.1485	0.2007	0.1895
2017	16,216	0.0000	0.9990	0.1075	0.1909	0.2088
2018	2,864	0.0001	0.9991	0.1397	0.2345	0.2457
2019	492	0.0001	0.9831	0.0686	0.1317	0.1622
2020	6,322	0.0001	0.9986	0.0959	0.1733	0.1868
2021	384	0.0000	0.9950	0.1060	0.2217	0.2668
2022	31,061	0.0000	1.0000	0.2990	0.3272	0.2654
2023	7,754	0.0001	0.9998	0.2890	0.2975	0.2068
All Years	78,679	0.0000	1.0000	0.1748	0.2491	0.2392
Texas (TX)						
Year	Observations	Min	Max	Median	Mean	Std_Dev
2010	1,642	0.0000	0.9900	0.1100	0.1900	0.2100
2011	82	0.0000	0.5300	0.0400	0.1000	0.1300
2012	1,336	0.0000	1.0000	0.0700	0.1300	0.1600
2013	1,241	0.0000	1.0000	0.1700	0.2500	0.2300
2014	654	0.0000	0.9700	0.0600	0.1300	0.1600
2015	8,972	0.0000	1.0000	0.1800	0.2400	0.2200
2016	10,509	0.0000	1.0000	0.2400	0.2700	0.2200
2017	61,229	0.0000	1.0000	0.3900	0.4000	0.2600
2018	2,091	0.0000	1.0000	0.1700	0.2600	0.2500
2019	8,998	0.0000	1.0000	0.3000	0.3200	0.2300
2020	1,191	0.0000	0.9900	0.1100	0.1900	0.2100
2021	1,237	0.0000	0.9700	0.1000	0.1700	0.1800
2022	346	0.0000	0.9600	0.1300	0.2200	0.2200
2023	285	0.0000	0.8900	0.1200	0.1900	0.2100
All Years	99,813	0.0000	1.0000	0.3200	0.3400	0.2600
Louisiana (LA)						
2010	177	0.0025	0.9125	0.0634	0.1153	0.1444
2011	1,789	0.0000	0.9982	0.1023	0.1971	0.2248
2012	9,195	0.0000	0.9984	0.1778	0.2195	0.2088
2013	702	0.0002	0.9934	0.1285	0.2132	0.2217
2014	486	0.0009	0.9543	0.1306	0.1990	0.1960
2015	488	0.0003	0.9467	0.0977	0.1830	0.2047
2016	26,704	0.0001	1.0000	0.4226	0.4247	0.2450
2017	1,741	0.0002	0.9939	0.1395	0.2134	0.2167
2018	346	0.0006	0.9952	0.1447	0.2312	0.2460
2019	2,031	0.0009	0.9961	0.1448	0.2045	0.1955
2020	2,737	0.0000	0.9996	0.1917	0.2482	0.2261
2021	11,615	0.0001	0.9999	0.2943	0.3077	0.2388
2022	123	0.0049	0.6765	0.0802	0.1506	0.1676
2023	109	0.0061	0.9761	0.0862	0.1514	0.1696
All Years	58,243	0.0000	1.0000	0.3123	0.3301	0.2510

Notes: This table provides annual summary statistics for the dependent variable *ratioCoverage*, which represents the ratio of the amount paid on building claims to the total building insurance coverage. The statistics are calculated for each year and state (Florida, Texas, and Louisiana) and include the number of observations, minimum, maximum, median, mean, and standard deviation. The "All Years" row aggregates the data across all years (2010-2023) to provide overall summary statistics for each state. These ratios reflect the proportion of insurance coverage utilised in claims, helping to evaluate the efficiency and adequacy of flood insurance coverage over time.

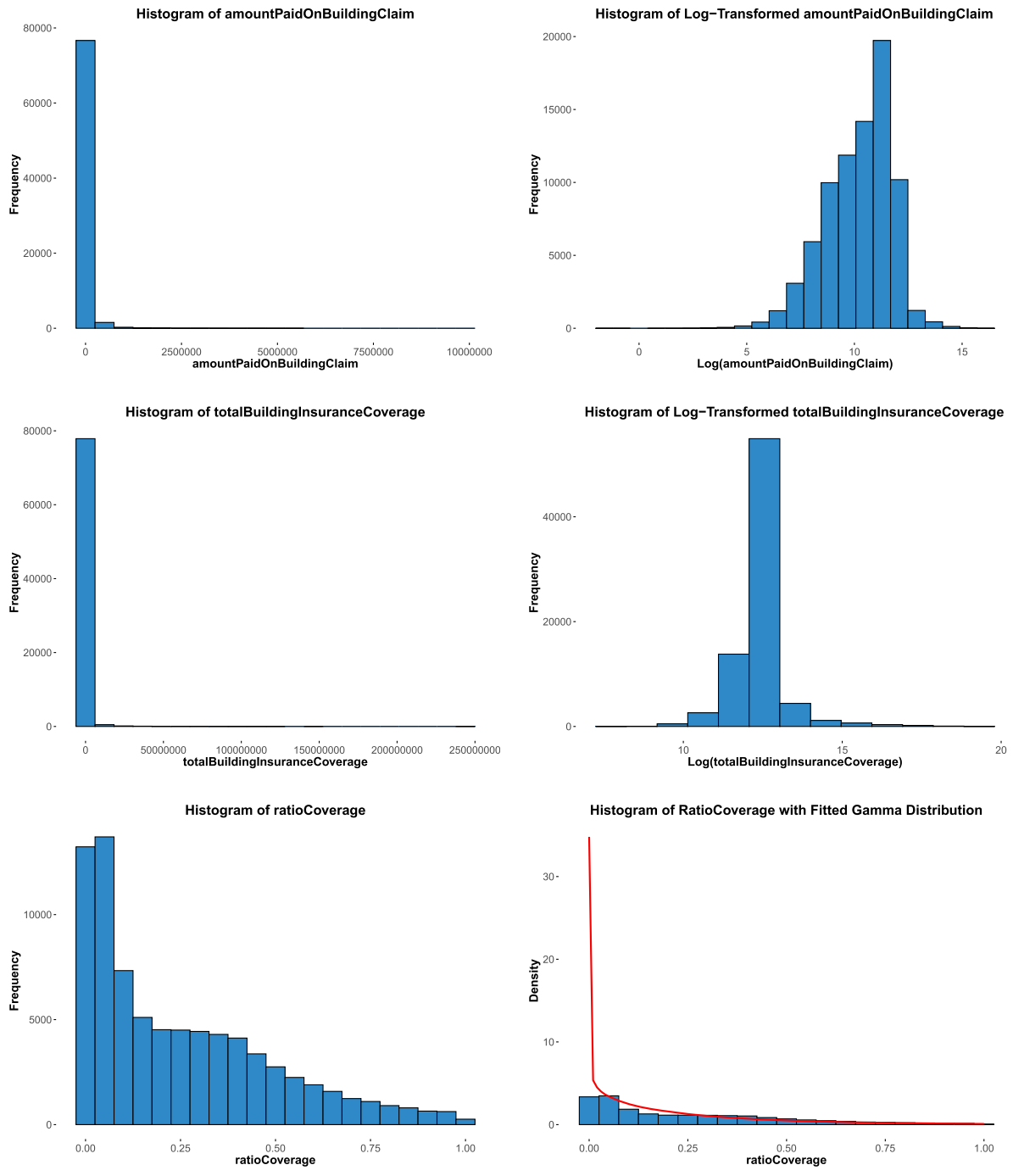

Fig. E.4. Distribution of Building Claim Data for Florida (FL).
Notes: The first row shows the histogram of the amount paid on building claims (left) and its logarithmic transformation (right). The second row depicts the histogram of the total building insurance coverage (left) and its logarithmic transformation (right). The third row illustrates the histogram of the ratio of the amount paid on building claims to the total building insurance coverage (left) and the histogram of the same ratio with an overlaid Gamma distribution fit (right). These plots represent all data aggregated across years (2010-2023) for Florida. Logarithmic transformations are applied to reduce skewness and provide better visual interpretation for highly dispersed data. The Gamma fit in the last plot highlights the suitability of a Gamma distribution for modelling the ratio of payments to coverage.

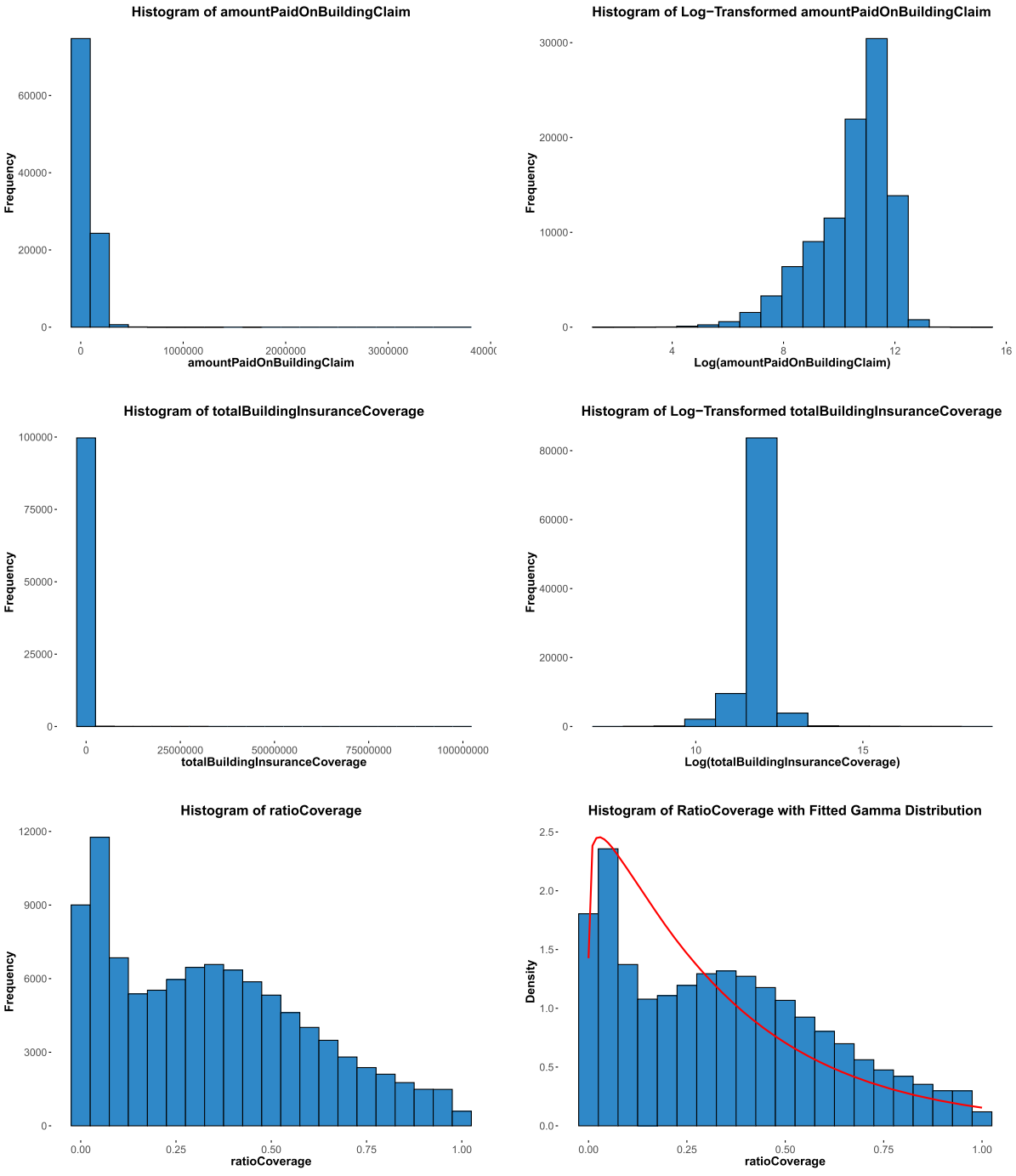

Fig. E.5. Distribution of Building Claim Data for Texas (TX).
Notes: The first row shows the histogram of the amount paid on building claims (left) and its logarithmic transformation (right). The second row depicts the histogram of the total building insurance coverage (left) and its logarithmic transformation (right). The third row illustrates the histogram of the ratio of the amount paid on building claims to the total building insurance coverage (left) and the histogram of the same ratio with an overlaid Gamma distribution fit (right). These plots represent all data aggregated across years (2010-2023) for Texas. Logarithmic transformations are applied to reduce skewness and provide better visual interpretation for highly dispersed data. The Gamma fit in the last plot highlights the suitability of a Gamma distribution for modelling the ratio of payments to coverage.

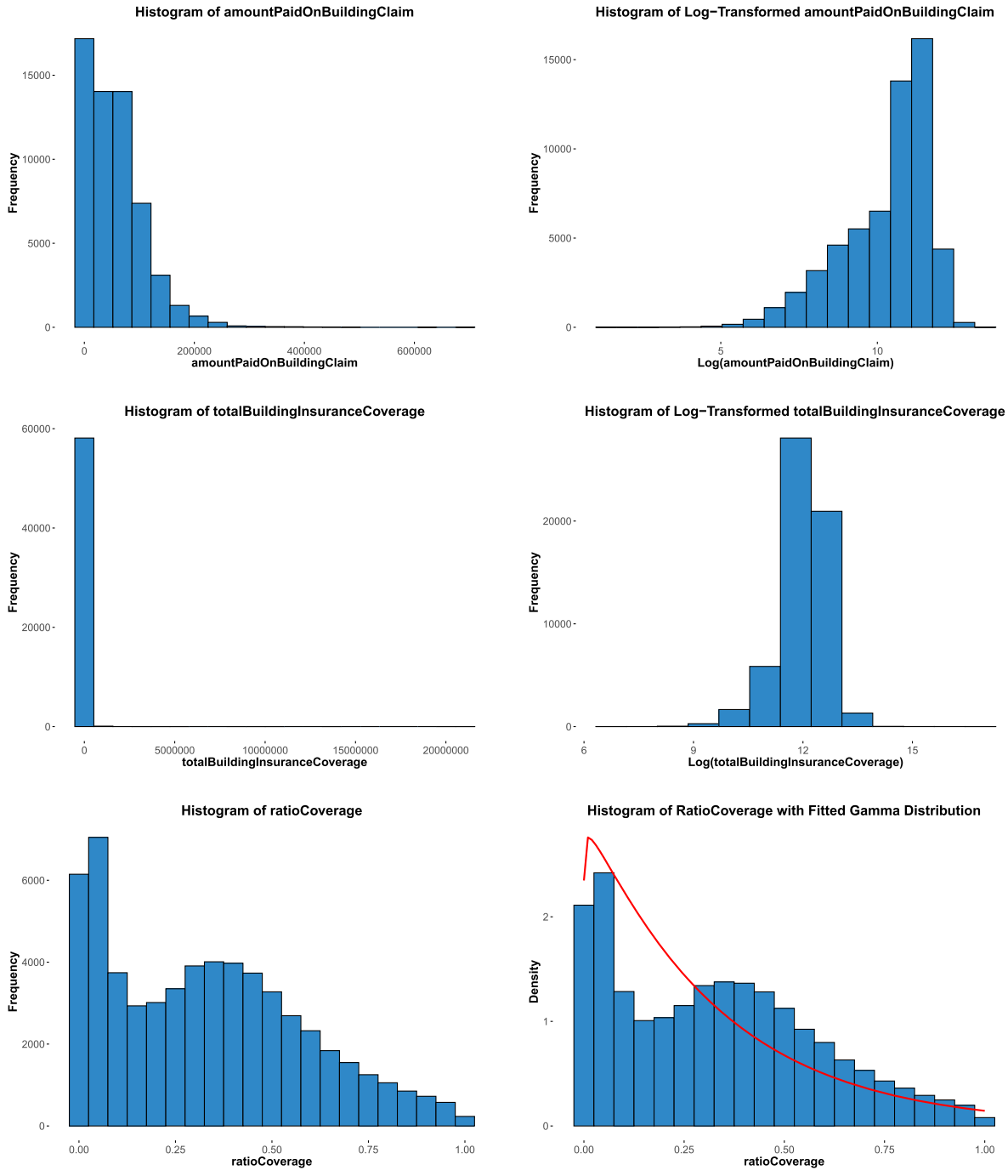

Fig. E.6. Distribution of Building Claim Data for Louisiana (LA).
Notes: The first row shows the histogram of the amount paid on building claims (left) and its logarithmic transformation (right). The second row depicts the histogram of the total building insurance coverage (left) and its logarithmic transformation (right). The third row illustrates the histogram of the ratio of the amount paid on building claims to the total building insurance coverage (left) and the histogram of the same ratio with an overlaid Gamma distribution fit (right). These plots represent all data aggregated across years (2010-2023) for Louisiana. Logarithmic transformations are applied to reduce skewness and provide better visual interpretation for highly dispersed data. The Gamma fit in the last plot highlights the suitability of a Gamma distribution for modelling the ratio of payments to coverage.

Data availability

Data will be made available on request.

References

Antonio, K., Beirlant, J., 2007. Actuarial statistics with generalized linear mixed models. Insur. Math. Econ. 40 (1), 58–76.

Apley, D.W., Zhu, J., 2020. Visualizing the effects of predictor variables in black box supervised learning models. J. R. Stat. Soc., Ser. B, Stat. Methodol. 82 (4), 1059–1086.

Asimit, V., Peng, L., Tunaru, R., Zhou, F., 2024. Risk budgeting under general risk measures. City Research Online, https://openaccess.city.ac.uk/id/eprint/33733/. Working paper.

Asimit, V., Peng, L., Wang, R., Yu, A., 2019. An efficient approach to quantile capital allocation and sensitivity analysis. Math. Finance 29 (4), 1131–1156.

Bach, F., 2010. Self-concordant analysis for logistic regression. Electron. J. Stat. 4, 384–414.

Bickel, P.J., Doksum, K.A., 2015. Mathematical Statistics: Basic Ideas and Selected Topics, Volume I, second ed. Chapman and Hall/CRC.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al., 2011a. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 3 (1), 1–122.

Boyd, S.P., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011b. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 3 (1), 1–122.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

Cetingoz, A.R., Fermanian, J.-d., Guéant, O., 2024. Risk budgeting portfolios: existence and computation. Math. Finance 34 (3), 896–924.

Debón, A., Montes, F., Puig, F., 2008. Modelling and forecasting mortality in Spain. Eur. J. Oper. Res. 189 (3), 624–637.

Delong, L., Lindholm, M., Wüthrich, M.V., 2021. Making Tweedie's compound Poisson model more accessible. Eur. Actuar. J. 11 (1), 185–226.

Denault, M., 2001. Coherent allocation of risk capital. J. Risk 4, 1–34.

Eling, M., Wirfs, J., 2019. What are the actual costs of cyber risk events? Eur. J. Oper. Res. 272 (3), 1109–1119.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 29 (5), 1189–1232.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Jørgensen, B., 1987. Exponential dispersion models. J. R. Stat. Soc., Ser. B, Stat. Methodol. 49 (2), 127–145.

Kuo, L., Mallick, B., 1998. Variable selection for regression models. Sankhya, Ser. B 60 (1), 65–81.

Mäkeläinen, T., Schmidt, K., Styan, G., 1981. On the existence and uniqueness of the maximum likelihood estimate of a vector-valued parameter in fixed-size samples. Ann. Stat. 9 (4), 758–767.

Marschner, I.C., 2011. Glm2: fitting generalized linear models with convergence problems. R J. 3 (2), 12–15.

McCullagh, P., Nelder, J., Wedderburn, R., 1989. Generalized Linear Models, second ed. Chapman and Hall/CRC.

Nelder, J., Wedderburn, R., 1972. Generalized linear models. J. R. Stat. Soc. A 135 (3), 370–384.

Nesterov, Y.E., 2004. Introductory Lectures on Convex Optimization - A Basic Course. Applied Optimization, vol. 87. Springer.

Wedderburn, R., 1976. On the existence and uniqueness of the maximum likelihood estimates for certain generalized linear models. Biometrika 63 (1), 27–32.

Zhang, X., Huang, F., Hui, F.K., Haberman, S., 2023. Cause-of-death mortality forecasting using adaptive penalized tensor decompositions. Insur. Math. Econ. 111, 193–213.