Efficient and proper GLM modelling with power link functions

Vali Asimit (Joint with Alexandru Badescu and Feng Zhou)

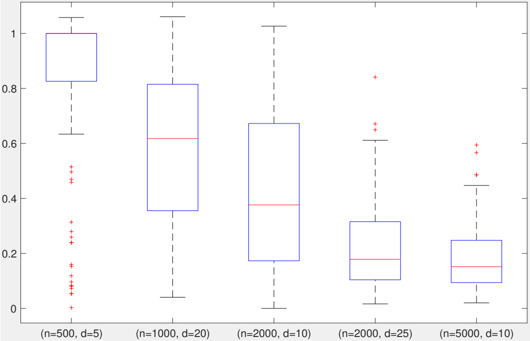
Bayes Business School, City, University of London

asimit@city.ac.uk

One World Actuarial Research Seminar, 06 December 2023

Agenda

Motivation behind a proper GLM model



Notes: Boxplots of the ratio between the L_1 distance (from the true value) of the IRLS GLM solution and L_1 distance (from the true value) of the MLE-based GLM solutions obtained in MATLAB with the use of *Fmincon*. Each box plot is built on $N = 500$ Inverse Gaussian samples of size n and d covariates, and all GLM models are fitted with log link functions.

Log Link $\quad P = \mathrm{E}\left[Y \mid \boldsymbol{X} = \boldsymbol{x}\right] = \exp\left\{\boldsymbol{x}^\top \boldsymbol{\beta}\right\}$ then $\dfrac{\partial P}{\partial x_k} = P \times \beta_k;$

Highlights of our work

Our Contributions:

- to formalise the concept of proper/ideal GLM modelling
- to provide a comprehensive characterisation of proper Tweedie GLM models
- to provide two novel, efficient and stable computational algorithms
- to highlight some insights (and possible pitfalls) about MATLAB, Python and R GLM packages

GLM: a quick overview

A univariate GLM: the response variable Y, $\mathcal{Y} \subseteq \Re$, is explained by covariates $\mathbf{X}$, $\mathcal{X} \subseteq \Re^d$. Let $\{P_{\theta,\phi} : \theta \in \Theta \subseteq \Re, \phi \in \Phi \subseteq \Re\}$ be the parametric set of distributions for Y (assume to be an *exponential dispersion model*) characterized by the probability density/mass function (pdf)

$$\log\left(f_Y(y;\theta,\phi)\right) = \frac{\theta y - b(\theta)}{a(\phi)} + c(y,\phi), \tag{1}$$

where $a(\cdot)$, $b(\cdot)$ and $c(\cdot,\cdot)$ are real-valued functions defined on Φ, Θ and $\mathcal{Y} \times \Phi$, respectively.

$$\mathrm{E}\left[Y_i \mid \boldsymbol{X}_i = \boldsymbol{x}_i\right] = h\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right). \tag{2}$$

The *link function* is denoted by $g = h^{-1}$. A common choice is the *Canonical link function*, $h(\eta) = b'(\eta)$. The MLE associated to (1) is obtained by solving the non-linear optimisation problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \Re^d}{\arg\max} \;\; \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{\theta_i y_i - b(\theta_i)}{a_i(\phi)} \;\text{ with }\; \theta_i = \left(b'^{-1} \circ h\right)\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right). \tag{3}$$

Without the loss of generality, we assume that $a_i(\phi) = a(\phi)$ for all $i = 1, \ldots, n$. This optimisation problem is well-defined and admits a (unique) solution for a *proper GLM model*.

GLM: a quick overview (cont'd)

A more general family is the *Exponential family* for which a subset – aka in the Canonical form is the same as the *exponential dispersion model* – while all other parametric families are in the Non-Canonical form.

$$\textit{Canonical form}: \qquad \log\left(f_Y(y;\theta,\phi)\right) = \frac{\theta y - b(\theta)}{a(\phi)} + c(y,\phi)$$

$$\textit{Non-Canonical form}: \qquad \log\left(f_Y(y;\theta,\phi)\right) = \frac{\xi(\theta)T(y) - b(\theta)}{a(\phi)} + c(y,\phi),$$

GLM implementations:

1) *Newton's method* – a second order method, so the Hessian needs to be computed
2) *Fisher Scoring* – like 1), but the Hessian is replaced by the Information matrix
3) *Iteratively Reweighted Least Squares (IRLS)* – a reweighted version of 2)

All GLM packages (MATLAB, Python and R) rely on IRLS implementations.

Note that the three implementations [1) – 3)] are identical for *exponential dispersion models* with a canonical link function, and thus, for Exponential family in Canonical form.

The main drawback 1)-3): we may find i) only local maxima/minima estimates if the optimisation function is **not concave**, or ii) the (global) *minimum Likelihood Estimate* if the optimisation function is **convex**.

GLM: a quick overview (cont'd)

Definition

The GLM model defined in (1) and (2) is said to be *proper* if the following two conditions are satisfied:

C1. The conditional mean relationship from (2) is properly mapped, i.e. $h : \Re \to b'(\Theta) \subseteq Conv(\mathcal{Y})$ with $b' : \Theta \to b'(\Theta)$ an injective function, where $Conv(\cdot)$ is the convex-hull of a set.[a]

C2. Assume that the likelihood function is well-defined in (3). The individual likelihood contribution is a (strictly) concave function, i.e.

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

where $\operatorname{sgn}(\cdot)$ is the signum function.

[a] Note that $Conv(\mathcal{Y})$ 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).

Condition **C1** ensures that the GLM estimation is well-defined.
Condition **C2** implies that the likelihood function ℓ defined in (3) is a well-defined and concave function in $\eta \in \Re$.

Tweedie family

Assume that $Y \sim \textit{Tweedie}(\theta, \phi)$ with pdf

$$\log\left(f_Y(y;\theta,\phi)\right) = \frac{\theta y - K_p(\theta)}{\phi} + \log\left(\mu'_\phi\left((-\infty, y]\right)\right), \quad (y,\theta,\phi) \in \mathcal{Y} \times \Theta \times \Re^*_+, \tag{4}$$

where $\Theta \subseteq \Re$, μ_ϕ is a Radon measure on $\mathcal{Y} \subseteq \Re$ and

$$K_p(\theta) := \begin{cases} \dfrac{\alpha - 1}{\alpha}\left(\dfrac{\theta}{\alpha - 1}\right)^{\alpha}, & p \in (-\infty, 0] \cup (1, \infty) \setminus \{2\}, \\ e^{\theta}, & p = 1, \\ -\log(-\theta), & p = 2, \end{cases} \tag{5}$$

with $\alpha = \dfrac{p-2}{p-1}$.
All Tweedie parametrisations are in Canonical form. Moreover, the Gauss, Poisson, Gamma and Inverse Gaussian families are obtained as special cases by taking $p = 0$, $p = 1$, $p = 2$, and $p = 3$, respectively.

GLM: Poisson

Proposition

The MLE-based Poisson GLM model is proper if and only if $h : \Re \rightarrow \Re_+^*$, *and*

$$-\,y \log\left(h\left(\eta\right)\right) + h\left(\eta\right) \quad \text{is convex in } \eta \text{ on } \Re \text{ for any given } y \in \mathbb{N}.$$

The *half-power* link function is one choice of proper link functions when $\gamma = 2k$, $k \in \mathbb{N}^*$, where the *power* LF is defined via the following expression

$$h\left(\eta\right) = \eta^{\gamma}, \qquad \eta \in \Re \quad \text{and} \quad \gamma \in \Re^*. \tag{6}$$

The *half-power* link function is one choice of proper link functions

$$h\left(\eta\right) = \begin{cases} \eta^{\gamma}, & \eta > 0, \\ +\infty, & \eta \leq 0, \end{cases} \tag{7}$$

with $\gamma \in \Re^*$. By taking $\gamma \geq 1$ in (7), it leads to solving a proper GLM model as follows

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \Re^d}{\arg\max}\;\; \ell\left(\boldsymbol{\beta}\right) = \sum_{i=1}^{n} \left(2ky_i \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right) - \left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^{2k} \right). \tag{8}$$

Note: While *half-power* link functions with positive even powers lead to proper GLM models that could be solved via a general convex programming algorithm, the case of *half-power* (with $\gamma = 2$ or $k = 1$), could be solved via our computationally efficient Algorithm 1 as explained later.

GLM: Gamma

Proposition

The MLE-based Gamma GLM model is proper if and only if $h : \Re \to \Re_+^*$, *and*

$$\frac{y}{h(\eta)} + \log\left(h(\eta)\right) \quad \textit{is convex in } \eta \textit{ on } \Re \textit{ for any given } y \in \Re_+^*.$$

In fact, $h(\eta) = \eta^{-2k}$ with $k \in \mathbb{N}^*$ are the only proper *power* link functions, where the *power* LF is defined in (6), so we have

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \Re^d}{\arg\max} \quad \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(2k \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right) - y_i \left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^{2k} \right). \tag{9}$$

Half-power link functions in (7) with $\gamma \leq -1$ lead to solving a proper GLM model.

Note: Similar to the Poisson GLM, the case of *half-power* (with $\gamma = -2$ or $k = 1$) link function, (9) can be solved by our computationally efficient Algorithm 1.

GLM: Inverse Gaussian (IG)

Proposition

The MLE-based Inverse Gaussian GLM model is proper if and only if $h : \Re \to \Re_+^*$, *and*

$$\frac{y}{2h^2(\eta)} - \frac{1}{h(\eta)} \quad \text{is convex in } \eta \text{ on } \Re \text{ for any given } y \in \Re_+^*.$$

Note 1: The *canonical* link function for *IG* GLM is the *inverse-square* function, $h(\eta) = \eta^{-1/2}$ with $\eta \in \Re$. It does not satisfy the conditions stated in the Proposition above, therefore it does not lead to a proper GLM.

Note 2: None of the *power* link functions are proper (see next slides). A compromise for running an *IG* regression would be to identify a link function for which the MLE could be efficiently solved, as we explained in our Algorithm 2.

Note 3: *half-power* link functions lead to proper IG GLM if and only if $\gamma \in [-1, -1/2]$.

Violations of Conditions **C1** and/or **C2** for Tweedie

Theorem

Let $Y \sim$ Tweedie(θ, ϕ) parameterised as in (4) with $p \in (-\infty, 0) \cup (1, 2) \cup (2, \infty)$ (or equivalently, $\alpha \in (-\infty, 2) \setminus \{0, 1\}$) such that $\mathcal{Y}, \Theta \in \{\Re, \Re^*, \Re^*_+, \Re^*_-\}$. Then, the following statements for the Tweedie GLM hold:

(i) *Condition **C1** in Definition 1 is only satisfied for the following settings:*
  a) $\Theta = b'(\Theta) = \Re^*_+$ (or $\Re_+$), $\mathcal{Y} \in \{\Re^*_+, \Re\}$ (or $\mathcal{Y} \in \{\Re_+, \Re\}$) and $1 < \alpha < 2$ (which is equivalent to $p < 0$) with $h : \Re \to \Re^*_+$ (or $h : \Re \to \Re_+$);
  b) $\Theta = \Re^*_-$, $b'(\Theta) = \Re^*_+$, $\mathcal{Y} \in \{\Re^*_+, \Re_+, \Re\}$ and $\alpha \in (-\infty, 1) \setminus \{0\}$ (which is equivalent to $p \in (1, \infty) \setminus \{2\}$) with $h : \Re \to \Re^*_+$;
  c) $\Theta = \Re$, $b'(\Theta) = \Re^*_+$, $\mathcal{Y} \in \{\Re^*_+, \Re_+, \Re^*\}$, $\alpha \in \{-2l + 1 : l \in \mathbb{N}^*\}$ with $h : \Re \to \Re^*_+$.
  d) $\Theta = \Re$, $b'(\Theta) = \Re^*$, $\mathcal{Y} \in \{\Re^*, \Re\}$, $\alpha \in \{-2l : l \in \mathbb{N}^*\}$ with $h : \Re \to \Re^*$.

(ii) *If $b'(\Theta) = \mathcal{Y}$, Condition **C2** in Definition 1 are not satisfied by any of the settings a)–d) with any power or negative power LFs, except of the following cases:*
  - setting b) with $0 < \alpha < 1$ and a power LF as in (6) such that its parameter $\gamma = -2k$ for any $k \in \mathbb{N}^*$ with $(1 - \gamma)\alpha \leq 1$,
  - setting b) with $\alpha < 0$ and a power LF as in (6) such that its parameter $\gamma = 2k$ for any $k \in \mathbb{Z}^*$,
  - setting c) and a power LF as in (6) such that its parameter $\gamma = 2k$ for any $k \in \mathbb{Z}^*$.

(iii) *If $b'(\Theta) = \mathcal{Y}$, Condition **C2** in Definition 1 are not satisfied by any of the settings a)–d) with any canonical LFs.*

Violations of Conditions **C1** and/or **C2** for Tweedie

Log LF setting

Proposition

Let $Y \sim$ Tweedie(θ, ϕ) parameterised as in (4) for which condition **C1** *from Definition 1 is satisfied. If $b'(\Theta) = \mathcal{Y}$, then a Tweedie GLM with a log LF is proper if and only if we either are in setting b) with $\alpha < 0$ or in setting c), where these setting are defined as in Theorem 5 i).*

Theorem

Let $Y \sim$ Tweedie(θ, ϕ) parameterised as in (4) for which condition **C1** *from Definition 1 is satisfied. Assume $b'(\Theta) = \mathcal{Y}$. Then, Condition* **C2** *in Definition 1 is not satisfied by any of the settings a)–d) with a negative half-power LF. Further, a Tweedie GLM with a half-power LF is proper if and only if*

- *setting a) with $1 < \alpha < 2$ and (7) such that $\frac{\alpha-1}{\alpha} \leq \gamma \leq \alpha - 1$,*
- *setting b) with $0 < \alpha < 1$ and (7) such that $\frac{\alpha-1}{\alpha} \leq \gamma \leq \alpha - 1$,*
- *setting b) with $\alpha < 0$ and (7) such that $\gamma \leq \alpha - 1$ or $\frac{\alpha-1}{\alpha} \leq \gamma$,*
- *setting c) with $\alpha \in \{-2l + 1 : l \in \mathbb{N}^*\}$ and (7) such that $\gamma \leq \alpha - 1$ or $\frac{\alpha-1}{\alpha} \leq \gamma$,*

where setting a)–c) are defined as in Theorem 5 i).

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 $\gamma \in [-1, -1/2]$

From the practical perspective, a proper GLM model is achieved only in the following instances:

i) *Compund Poisson-Gamma GLM* with $\alpha < 0$ (or equivalently $1 < p < 2$) is proper only for any *log LF* and any *power LF* as in (6) such that its parameter $\gamma = 2k$ with $k \in \mathbb{Z}^*$,

ii) *Gamma GLM* with $\alpha = 0$ (or equivalently $p = 2$) is proper only for any *log LF* and any *power LF* as in (6) such that its parameter $\gamma = -2k$ with $k \in \mathbb{N}^*$,

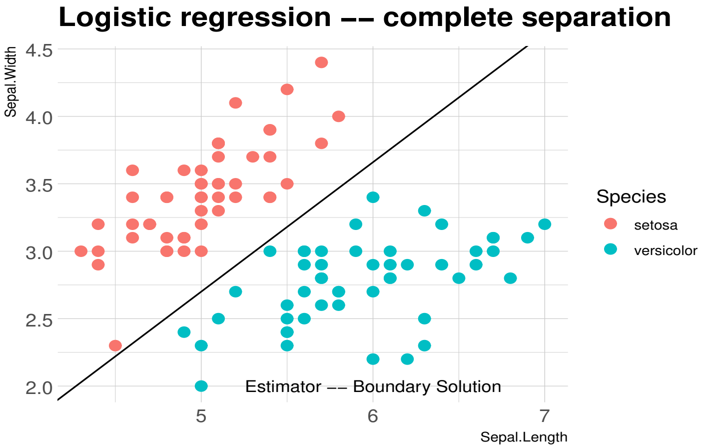
iii) *Positive stable distributions* with $0 < \alpha < 1$ (or equivalently $p > 2$) is proper only for any *power LF* as in (6) such that its parameter $\gamma = -2k$ with $k \in \mathbb{N}^*$,

iv) *Poisson GLM* with $\alpha = -\infty$ (or equivalently $p = 1$) is proper only for any *log LF* and any *power LF* as in (6) such that its parameter $\gamma = 2k$ with $k \in \mathbb{N}^*$.

This simplified summary does not include the results in the last theorem that cannot be solved by using well-known GLM packages. These proper GLMs requires constrained convex programming tools.

Boundary Solutions in GLM modelling – IRIS dataset

Optimal objective function may be $\pm\infty$, and thus, iterative methods are not reliable



Alternative algorithms

to Iteratively Reweighted Least Squares (IRLS) for GLM with power link functions

- *Newton's method for Self-Concordant problems (**NSC**)*:
  - Poisson regressions with *half-power (e.g. with $\gamma = 2$)* link function
  - Gamma regressions with *half-power (e.g. with $\gamma = -2$)* link function
- *Alternating Linearisation Method (**ALM**)*:
  - Inverse Gaussian regressions with *inverse-square-root* link function

The structure of ***self-concordant*** functions allows defining a refined Newton's method which is generally more efficient due to a reduced number of iterations.

Definition

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

[a] 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.

SC examples – Poisson and Gamma

Theorem

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* $\Omega := \bigcup_{i=1}^{n} \left\{\boldsymbol{\beta} \in \Re^d : \boldsymbol{x}_i^\top \boldsymbol{\beta} > 0\right\}$. *The following statements hold:*

a) *The MLE-based Poisson GLM equipped with the half-power LF from* (7) *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*

$$\min_{\boldsymbol{\beta} \in \Omega} \quad f_P(\boldsymbol{\beta}) := \sum_{i=1}^{n} \left(\frac{1}{2} \left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2 - y_i \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)\right), \tag{10}$$

$$\min_{\boldsymbol{\beta} \in \Omega} \quad \check{f}_P(\boldsymbol{\beta}) := \sum_{i=1}^{n} \left(\boldsymbol{x}_i^\top \boldsymbol{\beta} - y_i \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)\right). \tag{11}$$

b) *The MLE-based Gamma GLM equipped with the half-power LF from* (7) *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*

$$\min_{\boldsymbol{\beta} \in \Omega} \quad f_G(\boldsymbol{\beta}) := \sum_{i=1}^{n} \left(\frac{y_i}{2} \left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2 - \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)\right), \tag{12}$$

$$\min_{\boldsymbol{\beta} \in \Omega} \quad \check{f}_G(\boldsymbol{\beta}) := \sum_{i=1}^{n} \left(y_i \cdot \boldsymbol{x}_i^\top \boldsymbol{\beta} - \log\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}\right)\right). \tag{13}$$

Algorithm 1

Standard SC algorithm for half-power link function

This algorithm consists of two phases, the Step 1 guarantees that $f\left(\boldsymbol{z}^{(k)}\right) - f\left(\boldsymbol{z}^{(k+1)}\right) \geq \omega\left(\lambda^*\right)$, so the number of iterations in *Damped phase* N_{DP} is bounded with: $N_{DP} \leq \frac{f\left(\boldsymbol{z}^{(0)}\right) - f\left(\boldsymbol{z}^*\right)}{\omega(\lambda^*)}$, where $\omega(\lambda) := \lambda - \log\left(1+\lambda\right)$ on $\Re_+$.

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

Choose $\boldsymbol{z}^{(0)} \in dom\left(f\right)$, $\epsilon > 0$, and $\lambda^* \in \left(0, \tilde{\lambda}\right)$ where $\tilde{\lambda} = \frac{3-\sqrt{5}}{2}$;

Let $\nabla f\left(\cdot\right)$ and $\nabla^2 f\left(\cdot\right)$ be the gradient and Hessian, respectively, of f on Ω;

Define the *step/search direction* function $\Delta\left(\cdot\right) := \left[\nabla^2 f\left(\cdot\right)\right]^{-1} \nabla f\left(\cdot\right)$ on Ω;

Define $\lambda_f\left(\cdot\right) := \left(\nabla f\left(\cdot\right)^\top \left[\nabla^2 f\left(\cdot\right)\right]^{-1} \nabla f\left(\cdot\right)\right)^{1/2}$ 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.

Algorithm 2

Standard ALM algorithm

As mentioned earlier, the IG regression model is not proper for any power link function (including the canonical). If we assume an *inverse-square-root* link function ($h(\eta) = \eta^{-2}$), then only Condition **C1** is satisfied, which is not a proper GLM case.

$$\min_{\boldsymbol{\beta} \in \Omega} \quad f_{IG}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(\frac{y_i}{2} \left(\boldsymbol{x}_i^\top \boldsymbol{\beta} \right)^4 - \left(\boldsymbol{x}_i^\top \boldsymbol{\beta} \right)^2 \right). \tag{14}$$

The advantage of using the *inverse-square-root* link function is that (14) has a tractable solution via the *Alternating Linearisation Method (ALM)*.

$$\min_{(\boldsymbol{z},\boldsymbol{t}) \in \Re^d \times \Re^d} \quad G(\boldsymbol{z}, \boldsymbol{t}) = \sum_{i=1}^{n} \left(\frac{y_i}{2} \left(\boldsymbol{x}_i^\top \boldsymbol{z} \right)^2 \left(\boldsymbol{x}_i^\top \boldsymbol{t} \right)^2 - \left(\boldsymbol{x}_i^\top \boldsymbol{z} \right) \left(\boldsymbol{x}_i^\top \boldsymbol{t} \right) \right) \quad \text{so that } \; \boldsymbol{z} = \boldsymbol{t}. \tag{15}$$

This algorithm provides an approximation for $\boldsymbol{\beta}^*$, which denotes a local optimum of (14), 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 $\boldsymbol{t}_s \to \boldsymbol{\beta}^*$.

Algorithm 2

Standard ALM algorithm

As shown in the Algorithm 2, we define:

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

where $\langle .,. \rangle$ is the inner product, $\|\cdot\|_2$ 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 as:

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

Algorithm 2 (ALM-based)

Result: $(\boldsymbol{z}_{s^*}, \boldsymbol{t}_{s^*})$ that approximates $\boldsymbol{\beta}^*$, a local optimum of (14), 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**

$\quad \boldsymbol{z}_{s+1} := \arg\min_{\boldsymbol{z} \in \Re^d} H_1(\boldsymbol{z}, \boldsymbol{t}_s; \mu_{1,s})$;

$\quad$ **if** $f_{IG}(\boldsymbol{z}_{s+1}) \leq H_1(\boldsymbol{z}_{s+1}, \boldsymbol{t}_s; \mu_{1,s})$ **then**

$\quad\quad$ choose $\mu_{1,s+1} \geq \mu_{1,s}$;

$\quad$ **else**

$\quad\quad$ 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 $\boldsymbol{u}_{1,s} := \arg\min_{\boldsymbol{z} \in \Re^d} H_1\left(\boldsymbol{z}, \boldsymbol{t}_s; \mu^*_{1,s}\right)$;

$\quad\quad$ $\mu_{1,s+1} := \mu^*_{1,s}/b$ and $\boldsymbol{z}_{s+1} := \boldsymbol{u}_{1,s}$;

$\quad$ **end**

$\quad \boldsymbol{t}_{s+1} := \arg\min_{\boldsymbol{t} \in \Re^d} H_2(\boldsymbol{z}_{s+1}, \boldsymbol{t}; \mu_{2,s})$;

$\quad$ **if** $f_{IG}(\boldsymbol{t}_{s+1}) \leq H_2(\boldsymbol{z}_{s+1}, \boldsymbol{t}_{s+1}; \mu_{2,s})$ **then**

$\quad\quad$ choose $\mu_{2,s+1} \geq \mu_{2,s}$;

$\quad$ **else**

$\quad\quad$ 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 $\boldsymbol{u}_{2,s} := \arg\min_{\boldsymbol{t} \in \Re^d} H_2\left(\boldsymbol{z}_{s+1}, \boldsymbol{t}; \mu^*_{2,s}\right)$;

$\quad\quad$ $\mu_{2,s+1} := \mu^*_{2,s}/b$ and $\boldsymbol{t}_{s+1} := \boldsymbol{u}_{2,s}$;

$\quad$ **end**

end

Simulation study for Algorithm 1

Poisson GLM for *half-power* LF ($\gamma = 2$) and $N = 500$ samples

		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 *fitglm*	**MAER**	**0.9730**	**0.9620**	**0.9523**	**0.9685**	**0.9721**	**0.9713**	**0.9758**	**0.9782**	**0.9816**
	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 *sm.GLM*	**MAER**	**0.9393**	**0.8998**	**0.8431**	**0.9002**	**0.8463**	**0.6227**	**0.8838**	**0.8131**	**0.4723**
	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 *glm2*	**MAER**	**0.9999**	**0.9967**	**1.0014**	**1.0082**	**1.0085**	**1.0161**	**1.0087**	**1.0168**	**1.0376**
	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: The number of instances (out of $N = 500$) that the benchmarks cannot converge is shown as #NaN. All benchmarks use the same starting values with a maximum of $10,000$ iterations and 10^{-6} tolerance level.

Simulation study (cont'd)

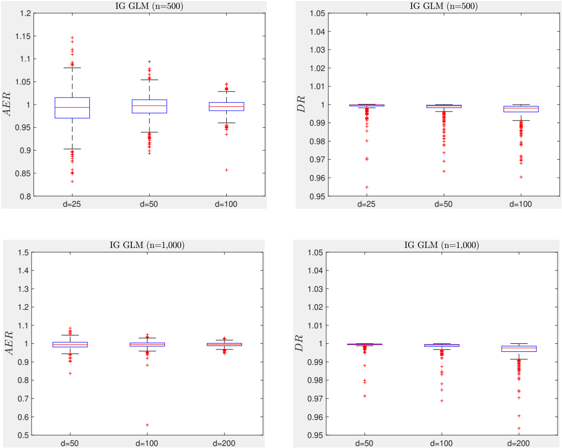
Gamma GLM for *half-power* LF ($\gamma = -2$) and $N = 500$ samples

		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 *fitglm*	**MAER**	**0.9216**	**0.9449**	**0.9722**	**0.6554**	**0.7141**	**0.8469**	**0.5547**	**0.5734**	**0.7167**
	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 *sm.GLM*	**MAER**	**0.9831**	**0.9930**	**0.9989**	**0.9962**	**0.9999**	**1.0000**	**0.9932**	**1.0000**	**1.0000**
	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 *glm2*	**MAER**	**0.9450**	**0.9608**	**0.9850**	**0.5843**	**0.7216**	**0.8928**	**0.4018**	**0.5434**	**0.7679**
	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: The number of instances (out of $N = 500$) that the benchmarks cannot converge is shown as #NaN. All benchmarks use the same starting values with a maximum of $10,000$ iterations and 10^{-6} tolerance level.

Simulation study for Algorithm 2 vs MATLAB only

Inverse Gaussian GLM with *inverse-square-root* LF and $N = 500$ samples



Notes: Box plots of Absolute Error Ratio (*AER*) are in left panel and Deviance Ratio (*DR*) are in right panel. MATLAB *fitglm* is implemented using the same starting value (as Algorithm 2) with a maximum of 10, 000 iterations and 10^{-6} tolerance level.

Conclusions and recommendations

- *Algorithm 1* is a more efficient and stable estimation tool
- In **Poisson GLM**, MATLAB *fitglm* does not converge in some cases
- In **Gamma** GLM, Python *sm.GLM* does not converge in some cases
- R *glm*2 improves the issues of convergence in the original R *glm*, but the accuracy of estimates are less than our *Algorithm 1* in **Gamma GLM**
- *Algorithm 2* is an efficient and stable estimation tool even for not proper GLM
- *Algorithms 1 and 2* address some issues of the **Poisson, Gamma and Inverse Gaussian GLMs**, but further novel and stable algorithms are needed (especially for **Tweedie GLMs**)
- General purpose GLM implementations are IRLS methods though some are "augmented" IRLS methods (MATLAB *fitglm* and R *glm*2) that are not always reliable


Thank You!