Supplementary Material

A. Distributions for Synthetic Data

The first two distributions, which we call 'sine' distributions, can be described as follows (we used two different settings for the 'rarity' parameter r):

1. Fix $\boldsymbol{\mu} \in \mathbb{R}^k$ and a symmetric positive semidefinite matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{k \times k}$. (In our experiments, $\boldsymbol{\mu}$ was sampled from a k-variate standard normal distribution, and $\boldsymbol{\Sigma}$ was sampled from a Wishart distribution with parameters $(\mathbf{I}, k)$, where $\mathbf{I}$ is the $k \times k$ identity matrix.)
2. Fix $\boldsymbol{\beta}^* \in \mathbb{R}^{k+1}$. (In our experiments, $\boldsymbol{\beta}^*$ was sampled from a $(k+1)$-variate standard normal distribution.)
3. The distribution D on $\mathcal{X} \times \{\pm 1\}$ is defined as follows: To generate an example $(\mathbf{x}, y)$, sample $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and let $\widetilde{\mathbf{x}} = [\mathbf{x}\,; 1]$; then sample $y \in \{\pm 1\}$ according to the conditional probability
$$\eta(\mathbf{x}) = \frac{1}{r}\left(\sin\left(\frac{\pi}{4}(\boldsymbol{\beta}^*)^\top \widetilde{\mathbf{x}}\right) + 1\right),$$
where $r > 0$ is a 'rarity' parameter that controls p: the higher the value of r, the more rare the positive class.

In our experiments, we used two 'sine' distributions with rarity parameters $r = 64$ (which for the specific distribution generated yielded $p = 0.0158$) and $r = 32$ (which for the specific distribution generated yielded $p = 0.0312$).

The third distribution, which we call the 'step' distribution, was defined similarly; the only difference was in the form of the class probability function:

1. Same as for 'sine' distribution above.
2. Same as for 'sine' distribution above.
3. The distribution D on $\mathcal{X} \times \{\pm 1\}$ is defined as follows: To generate an example $(\mathbf{x}, y)$, sample $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and let $\widetilde{\mathbf{x}} = [\mathbf{x}\,; 1]$; then sample $y \in \{\pm 1\}$ according to the conditional probability
$$\eta(\mathbf{x}) = \begin{cases} 0.10 & \text{if } (\boldsymbol{\beta}^*)^\top \widetilde{\mathbf{x}} < -30 \\ 0.03 & \text{if } -30 \le (\boldsymbol{\beta}^*)^\top \widetilde{\mathbf{x}} \le 30 \\ 0.13 & \text{if } (\boldsymbol{\beta}^*)^\top \widetilde{\mathbf{x}} > 30. \end{cases}$$

The 'step' distribution in our experiments had $p = 0.095$.

B. Run-Time Comparisons for Real Data

Table 5 shows the time (in seconds; rounded off to the nearest integer) it takes for the GEV-log and the GEV-canonical method to run for the data sets listed in Table 2. This includes the time for training as well as for validation of parameters. The results were averaged over 10 runs.

Table 5. Average training time (in seconds, including validation time) for GEV-log & GEV-canonical regression on UCI data sets.

DATASET	GEV-LOG	GEV-CANONICAL
NURSERY	21947	7242
LETTER-A	84037	39138
CAR	197	40
GLASS	7	1
ECOLI	7	2
LETTER-VOWEL	72628	6503
CMC	179	19
VEHICLE	39	5
HABERMAN	6	1
YEAST	200	27
GERMAN	114	13
PIMA	20	4

We observed that the optimization for GEV-canonical loss converges faster than for GEV-log on all the data sets. This is likely due to the Hessian of the GEV-canonical objective being better conditioned (the eigenvalues of the Hessian in this case are easily bounded, which is not the case for the Hessian of the GEV-log objective); the Newton method converges faster when the Hessian of the objective is well conditioned (Boyd & Vandenberghe, 2004).