OPT2022: 14th Annual Workshop on Optimization for Machine Learning

A Stochastic Prox-Linear Method for CVaR Minimization

Si Yi Meng SM2833@CORNELL.EDU
Department of Computer Science, Cornell University
Vasileios Charisopoulos VC333@CORNELL.EDU
Department of Operations Research & Information Engineering, Cornell University
Robert M. Gower RGOWER@FLATIRONINSTITUTE.ORG
Center for Computational Mathematics, Flatiron Institute, Simons Foundation

Abstract

We develop an instance of the stochastic prox-linear method for minimizing the Conditional Value-at-Risk (CVaR) objective. CVaR is a risk measure focused on minimizing worst-case performance, defined as the average of the top quantile of the losses. In machine learning, such a risk measure is useful to train more robust models. Although the stochastic subgradient method (SGM) is a natural choice for minimizing CVaR objective, we show that the prox-linear algorithm can be used to better exploit the structure of the objective, while still providing a convenient closed form update. We then specialize a general convergence theorem for the prox-linear method to our setting, and show that it allows for a wider selection of step sizes compared to SGM. We support this theoretical finding experimentally, by showing that the performance of stochastic prox-linear is more robust to the choice of step size compared to SGM.

1. Introduction

The most common approach to fit a model parametrized by $\theta \in \mathbb{R}^n$ to data, is to minimize the *expected* loss over the data distribution, that is

$$\min_{\theta \in \mathbb{R}^d} R_{\mathrm{ERM}}(\theta) = \mathbb{E}_{z \sim P}[\ell(\theta; z)]. \tag{1}$$

But in many cases, the expected loss may not be the suitable objective to minimize. When robustness or safety of the model are concerned, the emphasis should rather be on the extreme values of the distribution rather than the average value. For instance, in distributionally robust optimization, the goal is to optimize the model for the worst case distribution around some fixed distribution [14]. In extreme risk-averse settings, such as when safety is the top priority, one would minimize the maximum loss within a training set [30]. These applications can all be formulated as minimizing the expectation of the losses that are *above* some cutoff value,

$$\min_{\theta \in \mathbb{R}^d} R_{\mathrm{CVaR}}(\theta) = \mathbb{E}_{z \sim P}\left[\ell(\theta; z) \mid \ell(\theta; z) \geq \alpha_\beta(\theta)\right], \tag{2}$$

where $\alpha_\beta(\theta)$ is the upper β–quantile of the losses. For example, for $\beta = 0.9$, the problem in Equation 2 is to minimize the expectation of the worst 10% of the losses.

In this work, we investigate the use of the stochastic prox-linear (SPL) method introduced by Duchi and Ruan [15] for solving Equation 2. The possibility of applying SPL to CVaR minimization was mentioned in Davis and Drusvyatskiy [13], but not explored. We first derive a closed-form update for

SPL, and show why it is particularly well suited for minimizing CVaR. We give its convergence rates for convex and Lipschitz losses by specializing existing results from Davis and Drusvyatskiy [13]. Through several experiments comparing the stochastic prox-linear method to stochastic subgradient we show that the prox-linear algorithm is more robust to the choice of step size. We conclude with a discussion on several future applications for minimizing CVaR in machine learning.

1.1. Background

The CVaR objective was first introduced in finance as an alternative measure of risk, also known as the expected shortfall [2, 16]. Many applications in finance can be formulated as CVaR minimization problems, such as portfolio optimization [21, 25], insurance [17] and credit risk management [1]. The seminal work of Rockafellar and Uryasev [27] proposed a variational formulation of the CVaR objective that is amenable to standard optimization methods. This formulation has since inspired considerable research in applications spanning machine learning and adjacent fields, such as ν-SVM [18, 32], robust decision making and MDPs [7, 9–11, 29], influence maximization and submodular optimization [24, 26, 33], fairness [34], and federated learning [23].

Though it finds many applications, the CVaR objective is typically difficult to minimize. It is nonsmooth even when the individual losses $\ell(\cdot; z)$ are continuously differentiable. Indeed, if P does not admit a density — which is the case for all empirical distributions over training data — the variational objective is not everywhere differentiable. To address this, Laguel et al. [22] developed subdifferential calculus for a number of equivalent CVaR formulations and proposed minimizing a smoothed version of the dual objective. On the other hand, several works [20, 31] apply the stochastic subgradient method directly to the variational formulation proposed by Rockafellar and Uryasev [27], which is well-defined regardless of the distribution P. However, as we elaborate in Section 3, this approach is oblivious to the special structure of the variational form of the CVaR objective.

2. Problem setup

Let $\ell(\theta; z)$ be the loss associated with the model parameters $\theta \in \mathbb{R}^d$ and a measurable random variable $z(\omega)$ on some background probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

When z follows a distribution P with density $p(z)$, the cumulative distribution function on the loss for a fixed θ is given by $\mathbb{P}[\ell(\theta; z) \leq \alpha] = \int_{\ell(\theta;z)\leq\alpha} p(z)\, dz$, which we assume is everywhere continuous with respect to α. Let β be a confidence level, for instance $\beta = 0.9$. The Value-at-Risk (VaR) of the model is the lowest α such that with probability β, the loss will not exceed α. Formally,

$$\mathrm{VaR}_\beta(\theta) := \min\left\{\alpha \in \mathbb{R} \,:\, \mathbb{P}[\ell(\theta; z) \leq \alpha] \geq \beta\right\}, \quad (3)$$

The Conditional Value-at-Risk (CVaR) is the expectation of the upper tail starting at VaR_β, illustrated in Figure 1:

$$\mathrm{CVaR}_\beta(\theta) := \mathbb{E}_{z\sim P}[\ell(\theta; z) \mid \ell(\theta; z) \geq \mathrm{VaR}_\beta(\theta)]. \quad (4)$$

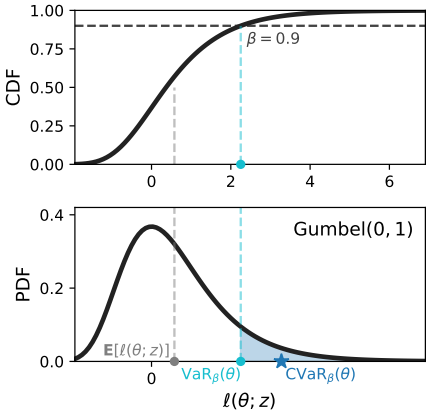

Figure 1: Expectation, VaR, and CVaR.

Clearly, the CVaR always upper bounds the VaR. Our goal is to minimize CVaR_β over $\theta \in \mathbb{R}^d$, but directly minimizing Equation 4 is not straightforward. Fortunately, Rockafellar and Uryasev [27] introduced a variational formulation where the solution to

$$\theta^*, \alpha^* \in \underset{\theta\in\mathbb{R}^d, \alpha\in\mathbb{R}}{\arg\min}\ F_\beta(\theta, \alpha) = \alpha + \frac{1}{1-\beta}\mathbb{E}_{z\sim P}\left[\max\left\{\ell(\theta; z) - \alpha,\ 0\right\}\right] \tag{5}$$

is such that θ^* is the solution to Equation 4, and we obtain $\alpha^* = \text{VaR}_\beta(\theta)$ as a byproduct.

3. The Stochastic Subgradient Method

A natural choice for minimizing Equation 5 is the stochastic subgradient method (SGM). Letting ∂f denote the convex subdifferential of f, at each step t we sample $z \sim P$ uniformly and compute a subgradient g_t from the subdifferential

$$\partial F_\beta(\theta_t, \alpha_t; z) = \begin{pmatrix}\mathbf{0}\\ 1\end{pmatrix} + \frac{1}{1-\beta}\,\partial \max\{u, 0\}|_{u=\ell(\theta_t;z)-\alpha_t}\begin{pmatrix}\partial\ell(\theta_t; z)\\ -1\end{pmatrix}. \tag{6}$$

Given some step size sequence $\{\lambda_t\} > 0$, the SGM then takes the step

$$x_{t+1} = x_t - \lambda_t g_t, \quad \text{where } g_t \in \partial F_\beta(\theta_t, \alpha_t; z), \text{ and } x = (\theta, \alpha)^\intercal.$$

For reference, the complete SGM algorithm is given in Algorithm 2. SGM is very sensitive to the step size choice and may diverge if not carefully tuned. This issue can be explained from a modeling perspective [13]. Indeed, SGM can be written as a model-based method where at each iteration t, it uses the following linearization of the sampled $F_\beta(x; z)$ at the current point x_t:

$$m_t(x; z) := F_\beta(x_t; z) + \langle g_t,\ x - x_t\rangle. \tag{7}$$

This provides an approximate, stochastic model of the objective $F_\beta(x)$. The SGM update is then a proximal step on this model, that is

$$x_{t+1} = \underset{x\in\mathbb{R}^{d+1}}{\arg\min}\ m_t(x; z) + \frac{1}{2\lambda_t}\left\|x - x_t\right\|^2. \tag{8}$$

The issue with this model $m_t(x; z)$ is that it uses a linearization to approximate the $\max\{\cdot, 0\}$ function. The linearization, which can take negative values, is a poor approximation of the non-negative $\max\{\cdot, 0\}$ operation. The main insight to the SPL method is to leverage the structure of $F_\beta(x)$ as a truncated function. This structure allows for a more accurate model that still has an easily computable proximal operator.

4. Stochastic prox-linear method for CVaR minimization

Here we introduce an alternative model for our objective that only linearizes *inside* the $\max\{\cdot, 0\}$, which is a strictly more accurate model when the objective is convex [3]. In particular, we use

$$m_t(x; z) = \alpha + \frac{1}{1-\beta}\max\left\{\ell(\theta_t; z) + \langle v_t,\ \theta - \theta_t\rangle - \alpha,\ 0\right\} \quad \text{for some } v_t \in \partial\ell(\theta_t; z). \tag{9}$$

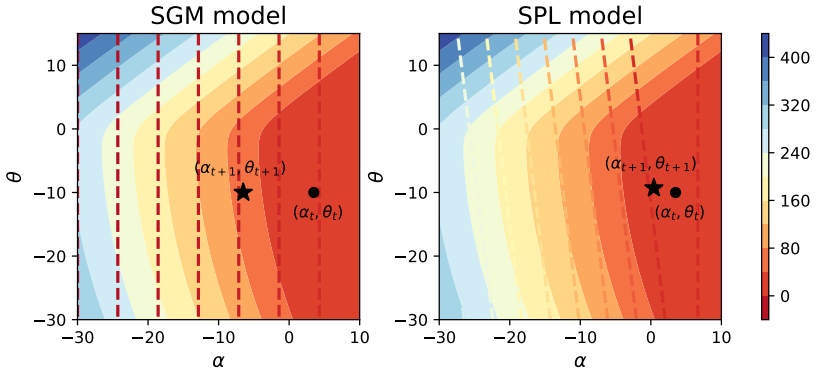

Figure 2: Comparison of SGM and SPL models on the CVaR objective with a single $\ell(\theta) = \log(1+\exp(\theta)) + \frac{0.01}{2}\theta^2$. Filled contours are the level sets of the objective, while the dashed contour lines are the level sets of the respective model m_t constructed at (θ_t, α_t). With the same step size, the SGM model results in an update that increases the objective, whereas the SPL model does not. Note that because the subgradient of the objective is 0 in θ, the SGM model is constant in θ.

The algorithm resulting from Equation 8 using this model is known as the stochastic prox-linear (SPL) method [15]. Figure 2 illustrates the difference between the two models. We derive the closed-form updates of SPL, which can be found in Algorithm 1. Furthermore, the cost of computing each iteration of SPL is of the same order of computing an iteration of SGM. In addition, we instantiate the convergence analyses from Davis and Drusvyatskiy [13] in the case of CVaR minimization, and compare the rates for SGM and SPL for losses satisfying the following Assumption.

Assumption 1 (Convex, subdifferentiable, and Lipschitz) *There exist square integrable random variables $M : \Omega \to \mathbb{R}$ such that for a.e. $z \in \Omega$ and all $\theta \in \mathbb{R}^d$, the sample losses $\ell(\theta; z)$ are convex, subdifferentiable*[1]*, and $M(z)$-Lipschitz.*

Theorem 1 (Convergence rates of SGM and SPL under convexity) *Suppose Assumption 1 holds. Let $x^* = (\theta^*, \alpha^*)^\intercal$ be a minimizer of $F_\beta(\theta, \alpha)$, and $\Delta = \|x_0 - x^*\|$ for an arbitrary initialization x_0. Consider the iterates $(x_t)_{t=1}^{T+1}$ given by SGM in Algorithm 2 or by SPL in Algorithm 1. There exists $\mathsf{L} > 0$ such that by using a constant step size $\lambda_t = \frac{\lambda}{\sqrt{T+1}}$ with $\lambda = \frac{\Delta}{\mathsf{L}\sqrt{2}}$, we have*

$$\mathbb{E}\left[F_\beta(\bar{x}_T) - F_\beta(x^*)\right] \leq \frac{\sqrt{2}\Delta\mathsf{L}}{\sqrt{T+1}}. \tag{10}$$

where $\bar{x}_T = \frac{1}{T}\sum_{t=1}^{T+1} x_t$ is the averaged iterate. In particular, the Lipschitz constants are given by

$$\mathsf{L}^2 = \mathbb{E}_z\left[\left(1 + \frac{1}{1-\beta}\sqrt{M(z)^2+1}\right)^2\right] \qquad (\textit{for } \textbf{\textit{SGM}}), \tag{11}$$

$$\mathsf{L}^2 = \mathbb{E}_z\left[\left(\frac{1}{1-\beta}\sqrt{M(z)^2+1}\right)^2\right] \qquad (\textit{for } \textbf{\textit{SPL}}). \tag{12}$$

1. Historically, the prox-linear method was proposed for composite optimization problems where the inner function is C^1 [6]. Here we slightly abuse the terminology and allow for general subdifferentiable losses $\ell(\cdot; z)$.

This result follows from Theorem 4.4 in Davis and Drusvyatskiy [13], and we verify the assumptions necessary in Appendix B. Note that $\lambda > 0$ can be chosen independent of L and Δ at the cost of a worse bound in Equation 10. Since L^2 is smaller for SPL, it allows for larger step sizes and improved constants in the convergence rate of Equation 10, though both methods enjoy the same asymptotic rate of $\mathcal{O}(1/\sqrt{T})$. In practice, the exact value of L is typically unavailable and λ is tuned heuristically via a grid search. Since SPL has theoretical guarantees under a wider range of valid step sizes, it should be easier to tune in practice. Our numerical experiments corroborate the theory, at least when the loss functions $\ell(\cdot; z)$ are smooth. When ℓ is itself nonsmooth we find the benefits to be negligible, though the performance of SPL remains at least as good as that of SGM.

5. Experiments

We now compare the empirical performance of SGM and SPL for minimizing the CVaR objective (Equation 5) using synthetic data. Similar to the setup of Holland and Haress [20], described in detail in Appendix C, we fix $\beta = 0.95$ and experiment with various combinations of loss functions $\ell(\cdot; z)$ and data distributions controlled by noise ζ. We employ a decreasing step size $\lambda_t = \lambda/\sqrt{t+1}$ and study the sensitivity of both methods to the initial step $\lambda_0 = \lambda$, varied over a logarithmically-spaced grid. Since the expectation in the objective is difficult to compute in closed form, we evaluate the suboptimality gaps using an empirical average over $N = 10^6$ data points sampled i.i.d. from the corresponding distribution under a single fixed seed. We denote this approximation by $\tilde{F}_\beta(\theta, \alpha)$.

Figure 3 shows the final suboptimality achieved by SGM and SPL for different λ. For smooth losses (squared and logistic) we see that SPL is significantly more robust and admits a much larger range of λ for which it does not diverge. Interestingly, for the absolute loss there is barely a difference. The same stability to larger step size can be observed by instead looking at the minimum number of iterations required to achieve ϵ final suboptimality $\tilde{F}(\theta, \alpha) - \tilde{F}^* \leq \epsilon$ (see Figure 5). Finally, we also include the same set of experiments on two real datasets. Details and results can be found in Appendix C.1.

6. Conclusion and future work

Our numerical evidence suggests that for the CVaR minimization problem, while both SGM and SPL can be tuned to achieve similar performance, SPL is strictly more tolerant to misspecified step sizes. To further speed up SPL and make it more competitive over SGM, some natural heuristics include the use of non-uniform sampling to bias towards training examples with higher losses (as in Curi et al. [12], Sagawa et al. [28]), as well as updating the parameters θ_t and α_t using different step sizes.

Efficient CVaR minimization with a stochastic algorithm opens up the possibility for new applications in machine learning. For instance, we could now consider models that trade-off between low average risk and heavy tails by adding the CVaR objective as a regularizer:

$$\min_{\theta \in \mathbb{R}^d} R_{\mathrm{ERM}}(\theta) + \rho R_{\mathrm{CVaR}_\beta}(\theta)$$

where $\rho > 0$ is a parameter that captures this trade-off. Controlling this trade-off is important as machine learning models are increasingly deployed in safety-critical applications that call for

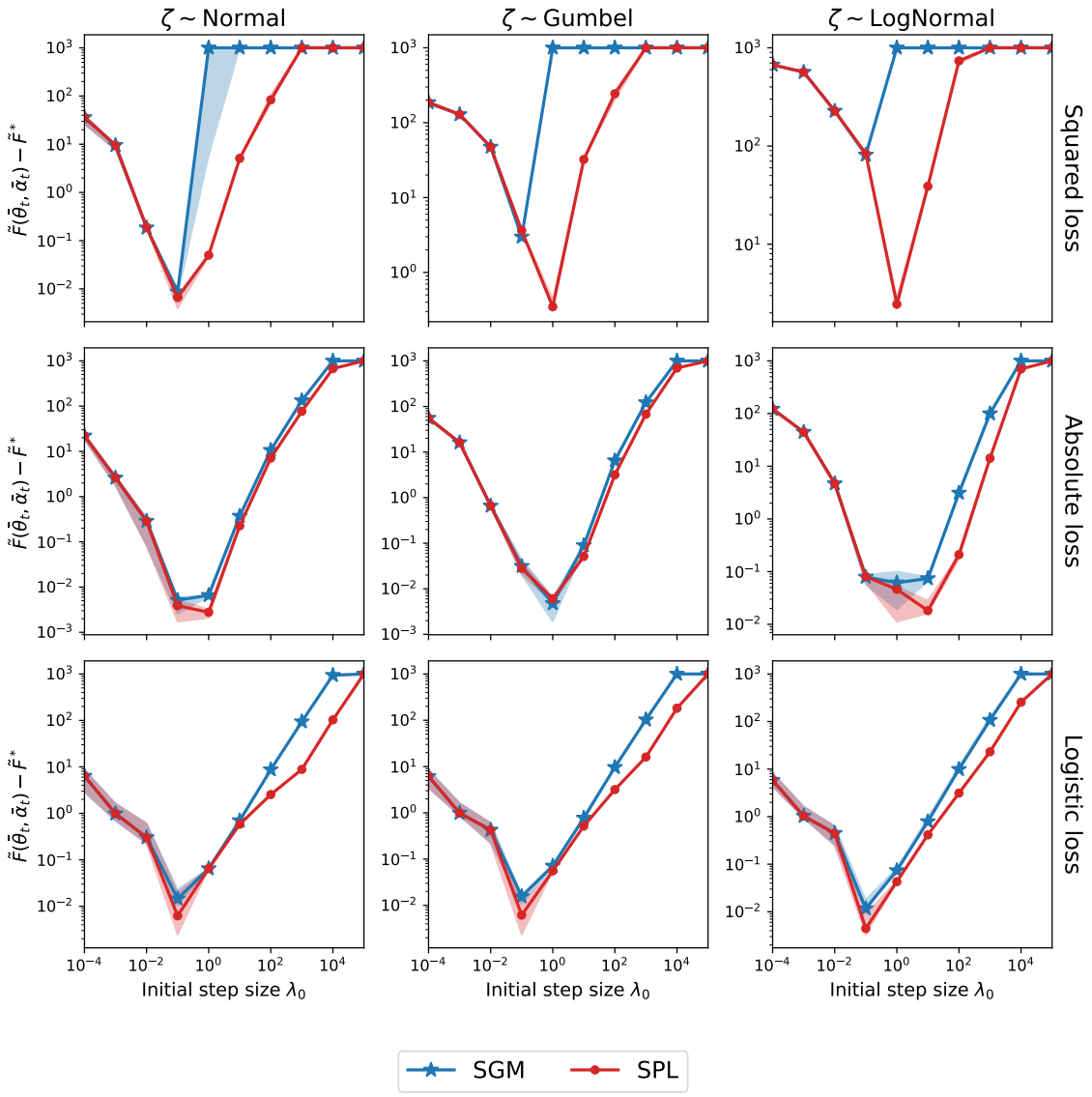

Figure 3: Sensitivity of final suboptimality to step size choices under a fixed $T = 10^5$ budget. The first two rows are regression tasks under the ℓ_1 and ℓ_2 losses, while the third row correspond to a binary classification task under the logistic loss. The columns correspond to different noise distributions in the data generation that controls the difficulty of the problem. More details can be found in Appendix C.

control over the likelihood of failure. As future work, we also see applications in training neural networks, where CVaR can be used to disincentivize the activations from being saturated too often, and thus help in speeding up training. This would offer an alternative to normalization layers, such as batchnorm or layernorm, which are used to bring the activations within a suitable range of the activations and thus avoid saturation.

References

[1] Fredrik Andersson, Helmut Mausser, Dan Rosen, and Stanislav Uryasev. Credit risk optimization with conditional value-at-risk criterion. *Mathematical programming*, 89(2):273–291, 2001.

[2] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[3] Hilal Asi and John C. Duchi. The importance of better models in stochastic optimization. *Proceedings of the National Academy of Sciences*, 116(46):22924–22930, 2019.

[4] Hilal Asi and John C. Duchi. Stochastic (approximate) proximal point methods: Convergence, optimality, and adaptivity. *SIAM Journal on Optimization*, 29(3):2257–2290, 2019.

[5] Amir Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, 2017.

[6] J. V. Burke and M. C. Ferris. A Gauss-Newton method for convex composite optimization. *Mathematical Programming*, 71(2):179–194, 1995. doi: 10.1007/BF01585997.

[7] Adrian Rivera Cardoso and Huan Xu. Risk-Averse Stochastic Convex Bandit. In *The 22nd International Conference on Artificial Intelligence and Statistics*, volume 89, pages 39–47, 2019.

[8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[9] Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for CVaR Optimization in MDPs. In *Advances in Neural Information Processing Systems 27*, pages 3509–3517, 2014.

[10] Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-Sensitive and Robust Decision-Making: a CVaR Optimization Approach. In *Advances in Neural Information Processing Systems 28*, pages 1522–1530, 2015.

[11] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-Constrained Reinforcement Learning with Percentile Risk Criteria. *Journal of Machine Learning Research*, 18:167:1–167:51, 2017.

[12] Sebastian Curi, Kfir Y. Levy, Stefanie Jegelka, and Andreas Krause. Adaptive sampling for stochastic risk-averse learning. In *Advances in Neural Information Processing Systems 33*, 2020.

[13] Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimimization*, 29(1):207–239, 2019.

[14] John C. Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv:1810.08750*, 2018.

[15] John C. Duchi and Feng Ruan. Stochastic methods for composite and weakly convex optimization problems. *SIAM Journal on Optimimization*, 28(4):3229–3259, 2018.

[16] Paul Embrechts, Sidney I. Resnick, and Gennady Samorodnitsky. Extreme value theory as a risk management tool. *North American Actuarial Journal*, 3(2):30–41, 1999.

[17] Paul Embrechts, Claudia Klüppelberg, and Thomas Mikosch. *Modelling extremal events: for insurance and finance*, volume 33. Springer Science & Business Media, 2013.

[18] Jun-ya Gotoh and Akiko Takeda. CVaR minimizations in support vector machines. *Financial Signal Processing and Machine Learning*, pages 233–265, 2016.

[19] Elad Hazan and Sham Kakade. Revisiting the Polyak step size. *arXiv:1905.00313*, 2019.

[20] Matthew Holland and El Mehdi Haress. Learning with risk-averse feedback under potentially heavy tails. In *International Conference on Artificial Intelligence and Statistics*, pages 892–900. PMLR, 2021.

[21] Pavlo Krokhmal, Jonas Palmquist, and Stanislav Uryasev. Portfolio optimization with conditional value-at-risk objective and constraints. *Journal of Risk*, 4:43–68, 2002.

[22] Yassine Laguel, Krishna Pillutla, Jérôme Malick, and Zaid Harchaoui. Superquantiles at work: Machine learning applications and efficient subgradient computation. *Set-Valued and Variational Analysis*, 29(4):967–996, 2021.

[23] Yassine Laguel, Krishna Pillutla, Jérôme Malick, and Zaid Harchaoui. A superquantile approach to federated learning with heterogeneous devices. In *55th Annual Conference on Information Sciences and Systems, CISS*, pages 1–6. IEEE, 2021.

[24] Takanori Maehara. Risk averse submodular utility maximization. *Operations Research Letters*, 43(5):526–529, 2015.

[25] Renata Mansini, Wlodzimierz Ogryczak, and Maria Grazia Speranza. Conditional value at risk and related linear programming models for portfolio optimization. *Annals of Operations Research*, 152(1):227–256, 2007.

[26] Naoto Ohsaka and Yuichi Yoshida. Portfolio optimization for influence spread. In *Proceedings of the 26th International Conference on World Wide Web*, pages 977–985, 2017.

[27] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of Conditional Value-at-Risk. *Journal of Risk*, 2:21–42, 2000.

[28] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020.

[29] Amir Sani, Alessandro Lazaric, and Rémi Munos. Risk-Aversion in Multi-armed Bandits. In *Advances in Neural Information Processing Systems 25*, pages 3284–3292, 2012.

[30] Shai Shalev-Shwartz and Yonatan Wexler. Minimizing the Maximal Loss: How and Why. In *Proceedings of the 33nd International Conference on Machine Learning*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 793–801. JMLR.org, 2016.

[31] Tasuku Soma and Yuichi Yoshida. Statistical learning with conditional value at risk. *arXiv:2002.05826*, 2020.

[32] Akiko Takeda and Masashi Sugiyama. ν-support vector machine as conditional value-at-risk minimization. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, volume 307, pages 1056–1063, 2008.

[33] Bryan Wilder. Risk-sensitive submodular optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[34] Robert C. Williamson and Aditya Krishna Menon. Fairness risk measures. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6786–6797, 2019.

Appendix A. SPL derivation for CVaR minimization

Before deriving the SPL updates, we first introduce the following lemma based on the truncated model from Asi and Duchi [4].

Lemma 1 (Truncated model) *Consider the problem*

$$x^{t+1} = \arg\min_{x \in \mathbb{R}^n} \ \max\left\{c + \langle a,\, x - x_t\rangle,\, 0\right\} + \frac{1}{2\lambda}\left\|x - x_t\right\|^2.$$

for some scalar c and vector $a \in \mathbb{R}^n$. The solution can be written in closed-form as

$$x^{t+1} = x_t - \min\left\{\lambda,\, \frac{\max\{c,\, 0\}}{\|a\|^2}\right\} a$$

Proof Note that x_{t+1} is the proximal point of the function

$$f(x) = h(\langle a, x\rangle + b), \quad \text{with } \ h(z) = \max\{z,\, 0\},\ b = c - \langle a, x_t\rangle.$$

centered at $x \equiv x_t$. Using Beck [5, Theorem 6.15], we have

$$\begin{aligned}\operatorname{prox}_{\lambda f}(x) &= x + \frac{a}{\|a\|^2}\left(\operatorname{prox}_{\lambda\|a\|^2 h}(\langle a, x\rangle + b) - (\langle a, x\rangle + b)\right) \\ &= x_t + \frac{a}{\|a\|^2}\left(\operatorname{prox}_{\lambda\|a\|^2 \max\{\cdot,\, 0\}}(c) - c\right) \end{aligned} \tag{13}$$

In turn, the max function is the support function of the interval $[0, 1]$. By Beck [5, Theorem 6.46], it follows that

$$\operatorname{prox}_{\lambda\|a\|^2 \max\{\cdot,\, 0\}}(c) = c - \lambda\|a\|^2 \operatorname{proj}_{[0,1]}\left(\frac{c}{\lambda\|a\|^2}\right). \tag{14}$$

Plugging Equation 14 into Equation 13, we obtain

$$\begin{aligned}\operatorname{prox}_{\lambda f}(x_t) &= x_t - \frac{a}{\|a\|^2} \cdot \lambda\|a\|^2 \operatorname{proj}_{[0,1]}\left(\frac{c}{\lambda\|a\|^2}\right) \\ &= x_t - \lambda a \cdot \operatorname{proj}_{[0,1]}\left(\frac{c}{\lambda\|a\|^2}\right).\end{aligned}$$

Writing $\operatorname{proj}_{[0,1]}(v) = \min\{\max\{v,\, 0\},\, 1\}$ yields the result. ■

We now derive the the SPL updates. Recall that for the CVaR objective, using the model in Equation 9, the stochastic model-based approach solves the following problem at each iteration:

$$\begin{aligned}\arg\min_{\theta,\alpha}\ f_t(\theta, \alpha; z) \coloneqq \alpha + \frac{1}{1-\beta}\max\left\{\ell(\theta_t; z) + \langle v_t,\, \theta - \theta_t\rangle - \alpha, 0\right\} \\ + \frac{1}{2\lambda}\left(\|\theta - \theta_t\|^2 + (\alpha - \alpha_t)^2\right)\end{aligned}$$

for some $v_t \in \partial \ell(\theta_t; z)$. We can apply Lemma 1 by transforming f_t into the truncated model form. First, we combine the α with its regularization term,

$$\begin{aligned}
\alpha + \frac{1}{2\lambda}(\alpha - \alpha_t)^2 &= \frac{1}{2\lambda}((\alpha - \alpha_t)^2 + 2\lambda\alpha) \\
&= \frac{1}{2\lambda}((\alpha - \alpha_t)^2 + 2\lambda\alpha - 2\lambda\alpha_t + \lambda^2) + \frac{1}{2\lambda}(2\lambda\alpha_t - \lambda^2) \\
&= \frac{1}{2\lambda}((\alpha - \alpha_t)^2 + 2\lambda(\alpha - \alpha_t) + \lambda^2) + \text{Const.} \\
&= \frac{1}{2\lambda}(\alpha + \lambda - \alpha_t)^2 + \text{Const.}
\end{aligned}$$

Using a change of variable

$$\hat{\alpha} = \alpha \qquad \text{and} \qquad \hat{\alpha}_t = \alpha_t - \lambda, \tag{15}$$

the quadratic regularization plus α is simplified to

$$\alpha + \frac{1}{2\lambda}(\alpha - \alpha_t)^2 = \frac{1}{2\lambda}\left(\|\theta - \theta_t\|^2 + (\hat{\alpha} - \hat{\alpha}_t)^2\right) + \text{Const.}.$$

Adding and subtracting $\hat{\alpha}_t$, the linearization inside the $\max\{\cdot, 0\}$ then becomes

$$\ell(\theta_t; z) - \hat{\alpha}_t + \left\langle \begin{pmatrix} v_t \\ -1 \end{pmatrix}, \begin{pmatrix} \theta - \theta_t \\ \hat{\alpha} - \hat{\alpha}_t \end{pmatrix} \right\rangle.$$

Letting $x = (\theta, \hat{\alpha})^\intercal$ and $x_t = (\theta_t, \hat{\alpha}_t)^\intercal$, we arrive at the form in Lemma 1 (up to constants) with

$$c = \frac{1}{1-\beta}(\ell(\theta_t; z) - \hat{\alpha}_t) \quad \text{and} \quad a = \frac{1}{1-\beta}\begin{pmatrix} v_t \\ -\lambda \end{pmatrix}.$$

The step size from Lemma 1 is

$$\eta := \min\left\{\lambda, \frac{\max\{c, 0\}}{\|a\|^2}\right\} \tag{16}$$

which, combined with the definition of $\hat{\alpha}$ in Equation 15, translates to the following updates:

1. If $c < 0 \implies \ell(\theta_t; z) < \alpha_t - \lambda$, then $\eta = 0$

$$\hat{\alpha}^* = \hat{\alpha}_t \implies \alpha^* = \alpha_t - \lambda, \quad \text{and} \quad \theta^* = \theta_t.$$

2. If $c > \lambda \|a\|^2 \quad (> 0)$, which implies checking for the case

$$\begin{aligned}
\frac{1}{1-\beta}\left(\ell(\theta_t; z) - \hat{\alpha}_t\right) &> \lambda\frac{1}{(1-\beta)^2}\left(\|v_t\|^2 + 1\right) \\
\ell(\theta_t; z) + \lambda - \alpha_t &> \frac{\lambda}{1-\beta}\|v_t\|^2 + \frac{\lambda}{1-\beta} \\
\alpha_t &< \ell(\theta_t; z) - \frac{\lambda}{1-\beta}\|v_t\|^2 - \lambda\frac{\beta}{1-\beta}.
\end{aligned}$$

Then the step size taken is $\eta = \lambda$, and the updates are

$$\hat{\alpha}^* = \hat{\alpha}_t + \frac{\lambda}{1-\beta} \implies \alpha^* = \alpha_t - \lambda + \frac{\lambda}{1-\beta} = \alpha_t + \lambda\frac{\beta}{1-\beta}$$
$$\theta^* = \theta_t - \frac{\lambda}{1-\beta} v_t.$$

3. Otherwise, $0 < \frac{c}{\|a\|^2} < \lambda$, then $\eta = \frac{c}{\|a\|^2}$, and the updates are

$$\hat{\alpha}^* = \hat{\alpha}_t + \frac{c}{\|a\|^2}\frac{1}{1-\beta}$$
$$\implies \alpha^* = \alpha_t - \lambda + \frac{\ell(\theta_t; z) - \hat{\alpha}_t}{\|v_t\|^2 + 1} = \alpha_t - \lambda + \frac{\ell(\theta_t; z) + \lambda - \alpha_t}{\|v_t\|^2 + 1}$$
$$\theta^* = \theta_t - \frac{c}{\|a\|^2}\frac{1}{1-\beta} v_t = \theta_t - \frac{\ell(\theta_t; z) + \lambda - \alpha_t}{\|v_t\|^2 + 1} v_t.$$

Together, these give us the closed-form updates of the SPL method, collected in Algorithm 1.

Algorithm 1 `SPL`: Stochastic prox-linear method for CVaR minimization

initialize: $\theta_0 \in \mathbb{R}^d$, $\alpha_0 \in \mathbb{R}$, **hyperparameter:** $\lambda > 0$
for $t = 0, 1, 2, \ldots, T-1$ **do**
  Sample data point $z \sim P$, compute $\ell(\theta_t; z)$ and $v_t \in \partial\ell(\theta_t; z)$
  $\lambda_t \leftarrow \lambda/\sqrt{t+1}$
  if $\alpha_t > \ell(\theta_t; z) + \lambda_t$ **then** ▷ α_t too big
    $\theta_{t+1} \leftarrow \theta_t$
    $\alpha_{t+1} \leftarrow \alpha_t - \lambda_t$
  else if $\alpha_t < \ell(\theta_t; z) - \frac{\lambda_t}{1-\beta}\left(\|v_t\|^2 + \beta\right)$ **then** ▷ α_t too small
    $\theta_{t+1} \leftarrow \theta_t - \frac{\lambda_t}{1-\beta} v_t$
    $\alpha_{t+1} \leftarrow \alpha_t + \frac{\lambda_t}{1-\beta}\beta$
  else ▷ α_t in middle range
    $\nu \leftarrow \frac{\ell(\theta_t) + \lambda_t - \alpha_t}{\lambda_t\left(\|v_t\|^2 + 1\right)}$
    $\theta_{t+1} \leftarrow \theta_t - \lambda_t \nu \nabla\ell(\theta_t; z)$
    $\alpha_{t+1} \leftarrow \alpha_t - \lambda_t + \lambda_t \nu$
  end if
end for
return $\bar{x}_T = \frac{1}{T+1}\sum_{t=1}^{T+1}(\theta_t, \alpha_t)^\intercal$

We also include the closed-form updates for the stochastic subgradient method applied to CVaR minimization in Algorithm 2.

Appendix B. Proof of Theorem 1

Proof To apply Theorem 4.4 in Davis and Drusvyatskiy [13], we must first verify their assumptions (B1)-(B4) hold. We will enumerate these under their following general setup: writing the CVaR

Algorithm 2 SGM: Stochastic subgradient method for CVaR minimization

initialize: $\theta_0 \in \mathbb{R}^d$, $\alpha_0 \in \mathbb{R}$, **hyperparameter:** $\lambda > 0$
for $t = 0, 1, 2, \ldots, T-1$ **do**
  Sample data point $z \sim P$, compute $\ell(\theta_t; z)$ and $v_t \in \partial \ell(\theta_t; z)$
  $\lambda_t \leftarrow \lambda / \sqrt{t+1}$
  if $\alpha_t \geq \ell(\theta_t; z)$ **then** ▷ α_t too big
    $\theta_{t+1} \leftarrow \theta_t$
    $\alpha_{t+1} \leftarrow \alpha_t - \lambda_t$
  else ▷ α_t too small
    $\theta_{t+1} \leftarrow \theta_t - \frac{\lambda_t}{1-\beta} v_t$
    $\alpha_{t+1} \leftarrow \alpha_t + \frac{\lambda_t}{1-\beta} \beta$
  end if
end for
return $\bar{x}_T = \frac{1}{T+1} \sum_{t=1}^{T+1} (\theta_t, \alpha_t)^\intercal$

objective as

$$F_\beta(x) = f(x) + r(x), \tag{17}$$

and interpret $r(x) = 0$ for SGM while $r(x) = \alpha$ for SPL. In the SPL case, we further write $f(x) = \mathbb{E}_z[h(c(x; z))]$ where $h(\cdot) = \frac{1}{1-\beta} \max\{\cdot, 0\}$ and $c(x; z) = \ell(\theta; z) - \alpha$. Recall that the stochastic one-sided models used are

$$\text{SGM} \qquad f_t(x; z) = F_\beta(x_t; z) + \langle g_t,\, x - x_t \rangle \qquad \text{where } g_t \in \partial F_\beta(x_t; z) \tag{18}$$

$$\text{SPL} \qquad f_t(x; z) = h(c(x_t; z) + \langle u_t,\, x - x_t \rangle) \qquad \text{where } u_t \in \partial c(x_t; z) \tag{19}$$

and the update in Equation 8 is equivalent to

$$x_{t+1} = \operatorname*{arg\,min}_{x \in \mathbb{R}^{d+1}} r(x) + f_t(x; z) + \frac{1}{2\lambda_t} \|x - x_t\|^2 \tag{20}$$

The assumptions we need to verify are the following, adapted from Davis and Drusvyatskiy [13]:

(B1) (**Sampling**) *It is possible to generate i.i.d. realizations* $z_1, z_2, \cdots \sim P$.

(B2) (**One-sided accuracy**) *There is an open set* U *containing* $\operatorname{dom} r$ *and a measurable function* $(x, y; z) \mapsto g_x(y; z)$*, defined on* $U \times U \times \Omega$*, satisfying*

$$\mathbb{E}_z\left[f_t(x_t; z)\right] = f(x_t) \quad \forall x_t \in U,$$

and

$$\mathbb{E}_z\left[f_t(x; z) - f(x)\right] \leq \frac{\tau}{2} \|x_t - x\|^2 \quad \forall x_t, x \in U.$$

Assumptions (B1) follows trivially from i.i.d. sampling, while (B2) follows from convexity of $\ell(\cdot; z)$, which results in $\tau = 0$.

(B3) (**Weak-convexity**) *The function* $f_t(x; z) + r(x)$ *is* η*-weakly convex for all* $x \in U$*, a.e.* $z \in \Omega$. Since $r(x)$ is also convex in both methods and both models are convex, (B3) holds with $\eta = 0$.

(B4) (**Lipschitz property**) *There exists a measurable function* $L : \Omega \to \mathbb{R}_+$ *satisfying* $\sqrt{\mathbb{E}_z[L(z)^2]} \leq \mathsf{L}$ *and such that*

$$f_t(x_t; z) - f_t(x; z) \leq L(z) \|x_t - x\| \quad \forall x_t, x \in U \text{ and a.e. } z \sim P.$$

This can be easily proved for SGM using the Lipschitz assumption on $\ell(\cdot; z)$ and that $\max\{\cdot, 0\}$ is 1-Lipschitz:

$$\begin{aligned} f_t(x_t; z) - f_t(y; z) &\leq \|g_t\| \|x_t - y\| \\ &\leq \left(1 + \frac{1}{1-\beta}\sqrt{M(z)^2 + 1}\right) \|x_t - y\|, \end{aligned}$$

which gives us $\mathsf{L}^2 = \mathbb{E}_z\left[\left(1 + \frac{1}{1-\beta}\sqrt{M(z)^2+1}\right)^2\right]$. For SPL,

$$\begin{aligned} (1-\beta)(f_t(x_t; z) - f_t(y; z)) &= \max\{\ell(\theta_t; z) - \alpha_t, 0\} - \max\{\ell(\theta_t; z) - \langle v_t, \theta - \theta_t\rangle - \alpha, 0\} \\ &\leq \max\{\langle v_t, \theta - \theta_t\rangle + (\alpha - \alpha_t), 0\} \\ &= \max\left\{\left\langle \begin{pmatrix} v_t \\ 1 \end{pmatrix}, \begin{pmatrix} \theta - \theta_t \\ \alpha - \alpha_t \end{pmatrix} \right\rangle, 0\right\} \\ \implies f_t(x_t; z) - f_t(y; z) &\leq \frac{1}{1-\beta}\left(\sqrt{M(z)^2+1}\right)\|x_t - y\| \end{aligned}$$

This gives us the slightly improved constant for SPL:

$$\mathsf{L}^2 = \mathbb{E}_z\left[\left(\frac{1}{1-\beta}\sqrt{M(z)^2+1}\right)^2\right].$$

All assumptions for Theorem 4.4 in Davis and Drusvyatskiy [13] are satisfied, and so under the step size $\lambda_t = \frac{\lambda}{\sqrt{T+1}}$, with $\lambda = \frac{\Delta}{\mathsf{L}\sqrt{2}}$ and an averaged final iterate gives us the rate

$$\mathbb{E}\left[F_\beta(\bar{x}_T) - F_\beta(x^*)\right] \leq \frac{\sqrt{2}\Delta\mathsf{L}}{\sqrt{T+1}}.$$

■

Appendix C. Experiment details and additional results

For all problems we set the dimension to be $d = 10$. For regression problems, $\theta_{\text{gen}} \sim \mathcal{U}([0,1]^d)$, and for classification (logistic regression) we use $\theta_{\text{gen}} \sim \mathcal{U}([0,10]^d)$ to increase linear separability. The loss functions and target generation schemes are listed in Table 1. Each target of the corresponding problem contains an error ϵ from one of the distributions Table 2, which is intended to control the difficulty level of the problem. For each error distribution and loss function combination, we draw N independent samples and use the following discretization as an approximation to Equation 5

$$\tilde{F}_\beta(\theta, \alpha) = \alpha + \frac{1}{1-\beta}\frac{1}{N}\sum_{i=1}^{N}\max\{\ell(\theta; z_i) - \alpha, 0\}. \tag{21}$$

Table 1: Loss functions and data generation. The error distributions for ζ are described in Table 2. We use $\sigma(\cdot)$ to denote the sigmoid function, and all x's are sampled uniformly from the unit sphere.

Task	**Loss** $\ell(\theta; x, y)$	**Target**
Regression	$\frac{1}{2}(x^\mathsf{T}\theta - y)^2$	$y = x^\mathsf{T}\theta_{\text{gen}} + \zeta$
Regression	$\lvert x^\mathsf{T}\theta - y\rvert$	$y = x^\mathsf{T}\theta_{\text{gen}} + \zeta$
Classification	$\log\left(1 + \exp\left(-yx^\mathsf{T}\theta\right)\right)$	$y = 1$ w.p. $\sigma(x^\mathsf{T}\theta_{\text{gen}} + \zeta)$ and -1 otherwise.

We set $\beta = 0.95$ for all experiments, and thus have omitted β from all plot descriptions. We run full-batch subgradient method with the adaptive Polyak step size from Hazan and Kakade [19] for a total of 2000 iterations divided into $K = 100$ iterations of the inner loop, using the lower bound 0 for the initial estimate of the minimum objective value. We record the final θ^*, α^*, and $F^* := \tilde{F}_\beta(\theta^*, \alpha^*)$.

For the SPL against SGM comparison, we set $\alpha_0 = 0$ and $\theta_0 \sim \mathcal{N}(0, I_d)$ at initialization. Both algorithms are run for $T = 100,000$ iterations using 5 different seeds that control the randomness of initialization and sampling during the course of optimization. In the sensitivity plots (Figures 3 and 5), solid lines show the median values, while the shaded regions indicate the range over the random seeds. All objective evaluations are on $\tilde{F}_\beta(\bar{\theta}_t, \bar{\alpha}_t)$ using the averaged iterates.

Table 2: Error distributions (all centered at 0).

Distribution of ζ	**Parameters**
Normal(μ, σ^2)	$\mu = 0, \sigma = 2$
Gumbel(μ, β)	$\mu = 0, \beta = 4$
LogNormal(μ, σ^2)	$\mu = 2, \sigma = 1$

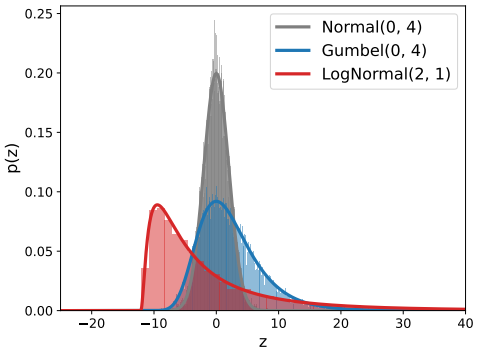

Figure 4: Error distributions in 1D.

C.1. Additional experiment results

Figure 5 shows a similar sensitivity analysis to Figure 3 in the main text. Instead of the sensitivity of final suboptimality, here we show the sensitivity of the minimum number of iterations to reach ϵ-suboptimality $\tilde{F}(\theta, \alpha) - \tilde{F}^* \leq \epsilon$.

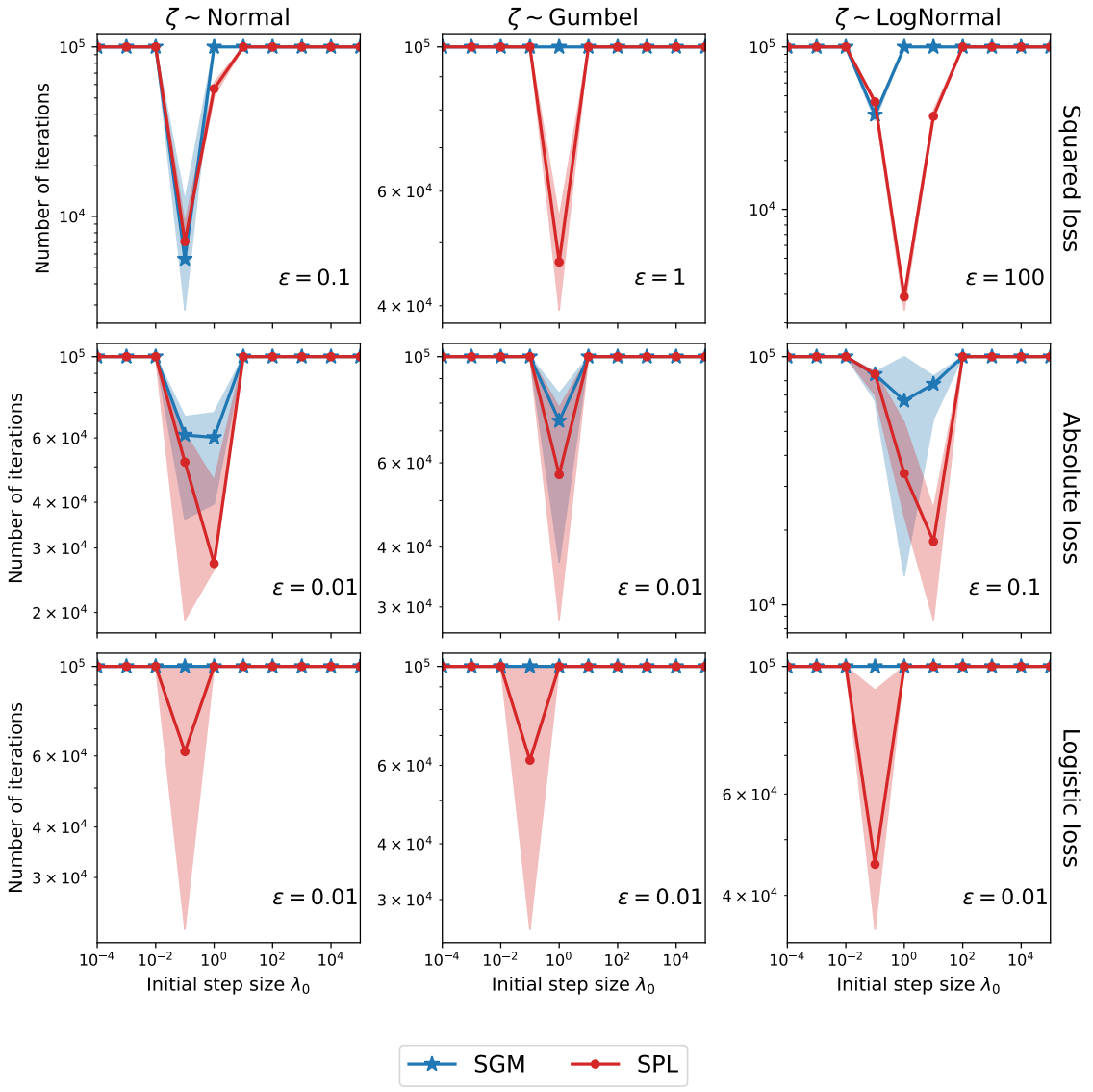

Figure 5: Sensitivity of number of iterations to achieve ϵ suboptimality to step size choices. The first two rows are regression tasks under the ℓ_1 and ℓ_2 losses, while the third row correspond to a binary classification task under the logistic loss. The columns correspond to different noise distributions in the data generation that controls the difficulty of the problem.

Finally, we present the same experiment on two real datasets, `YearPredictionMSD` and (binary) `Covertype` from the LIBSVM repository [8]. The objective now is the empirical CVaR

$$F_\beta(\theta, \alpha) = \alpha + \frac{1}{1-\beta}\frac{1}{n}\sum_{i=1}^{n} \max\{\ell(\theta; z_i) - \alpha,\, 0\}$$

where n is the number of examples in the training split. The loss function $\ell(\cdot; z_i)$ is the squared loss for `YearPredictionMSD`, and logistic loss for `Covertype`. Similar to the synthetic experiments, we set $\beta = 0.95$ and run full-batch subgradient method with the adaptive Polyak step size from Hazan and Kakade [19] to compute θ^* and α^*, but for a total of 10000 iterations divided into $K = 50$ iterations of the inner loop, as these datasets are large and more difficult than the simulated ones. For the comparison between SPL and SGM, we run both methods for $100n$ iterations. The results are in Figure 6 and Figure 7, from which we draw similar conclusions as in the synthetic experiments.

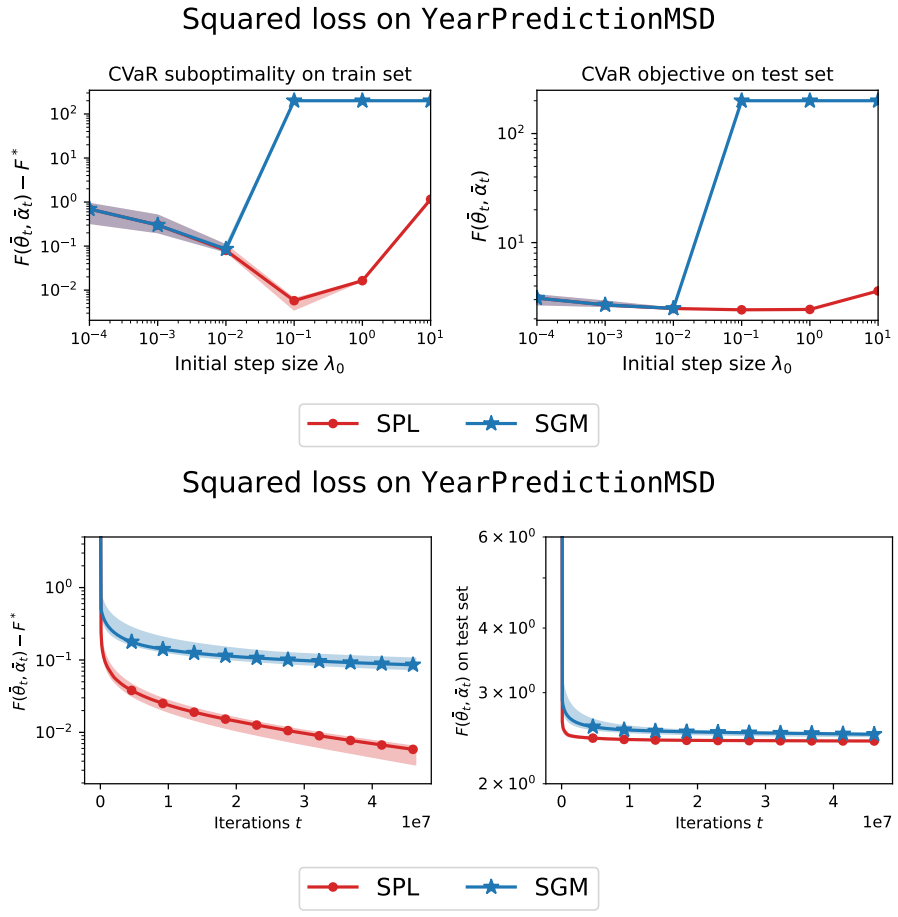

Figure 6: Sensitivity and convergence plots on the `YearPredictionMSD` linear regression task. The convergence plot is based on the best initial step size at the end of training for each method.

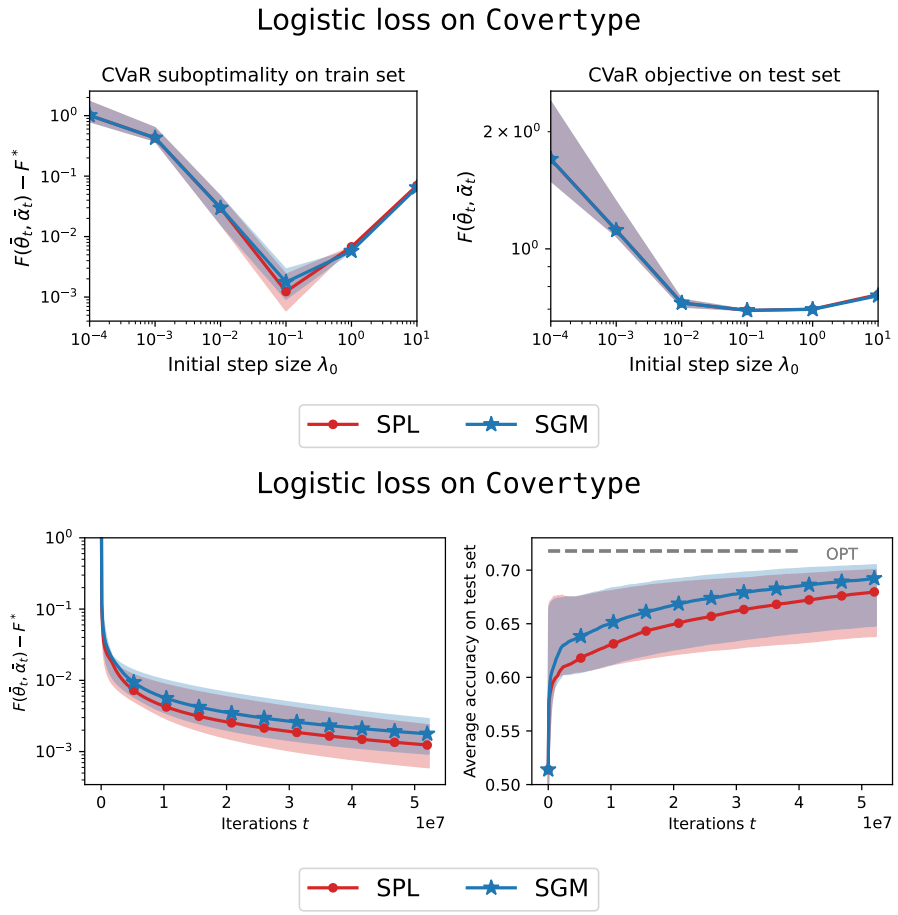

Figure 7: Sensitivity and convergence plots on the `Covertype` binary classification task. The convergence plot is based on the best initial step size at the end of training for each method. The grey dashed line is the average accuracy on the test set achieved by θ^*. Note that the reported accuracy is averaged across the entire training set, but since SPL reached a lower CVaR objective (rather than the average loss objective), it is reasonable that its average accuracy is lower.