You Shall Pass: Dealing with the Zero-Gradient Problem in Predict and Optimize for Convex Optimization

Grigorii Veviurko [1] **Wendelin Böhmer** [1] **Mathijs de Weerdt** [1]

Abstract

In predict and optimize, machine learning models are trained to predict parameters of optimization problems using task performance as the objective. A key challenge is computing the Jacobian of the solution with respect to its parameters. While linear problems typically use approximations due to a zero or undefined gradient, non-linear convex problems often utilize the exact derivative. This paper demonstrates that the zero-gradient problem also occurs in the non-linear case and introduces a smoothing technique which, combined with quadratic approximation and projection distance regularization, solves the zero-gradient problem. Experiments on a portfolio optimization problem confirm the method's efficiency.

1. Introduction

Predict and optimize (P&O) [1] combines machine learning (ML) with mathematical programming, focusing on optimization problems with unknown parameters that need to be predicted before solving the problem. Instead of training ML models to match the distribution of unknown parameters, P&O uses task performance as the objective for training. To train P&O models with gradient descent [2], the Jacobian of the optimization problem solution with respect to the parameter should be computed. For linear problems, this Jacobian is either zero or undefined and is usually approximated [1], [3], [4]. In the non-linear case, differential optimization method introduced by Agrawal et al. [5] allows for exact computation of the Jacobian.

This paper demonstrates that the *zero-gradient* problem occurs not only in the linear case but also in general non-linear convex settings. Specifically, the null space of the true Jacobian of a convex optimization problem depends on the number of active constraints. Hence, the zero-gradient occurs when the solution approaches the boundary of the feasible set, activating the constraints. We propose a method that combines quadratic programming approximation similar to [6] with projection distance regularization from [7] and a novel idea of *local smoothing*. The resulting algorithm has a simple geometric interpretation and is theoretically justified. Using a portfolio optimization problem [8], we demonstrate that *a)* the zero-gradient problem indeed occurs in the non-linear setup, and *b)* the proposed solution resolves this problem and significantly outperforms the approach using the exact Jacobian.

2. Related work

The P&O framework was first introduced by Elmachtoub et al. [1], who consider combinatorial optimization problems and derive a convex sub-differentiable approximation of the task performance function to enable training. Vlastelica et al. [3] obtain a piecewise-linear approximation of the task performance, Berthet et al. [9] employ stochastic perturbations to approximate the Jacobian, and Sahoo et al. [4] show that using projections on top of the predictor enables using the identity matrix as an approximation of the Jacobian.

In convex optimization, exact differentiation is possible [5] for disciplined convex programs [10]. This result gave rise to new applications of P&O in convex optimization, such as the portfolio optimization problem [11] and surrogate model learning [8]. Moreover, several studies use this technique to approximate the Jacobian of combinatorial problems. For example, Wilder et al. [6] construct a quadratic approximation of the problem and use its Jacobian. This approach is then extended by using logarithmic approximations [12].

3. Problem formulation

In this section, we define the P&O problem. We refer readers to Elmachtoub et al. [1] for further details. In predict and optimize, the *original optimization problem* is of the form

$$\arg\max_{x} f(x, w) \text{ s. t. } x \in \mathcal{C}, \tag{1}$$

where $x \in \mathbb{R}^n$ is the decision variable, $w \in \mathbb{R}^u$ is a vector of unknown parameters, $f : \mathbb{R}^n \times \mathbb{R}^u \to \mathbb{R}$ is the objective

[1] Delft University of Technology. Correspondence to: Grigorii Veviurko <g.veviurko@tudelft.nl>.

Published at the 2^{nd} Differentiable Almost Everything Workshop at the 41^{st} International Conference on Machine Learning, Vienna, Austria. July 2024. Copyright 2024 by the author(s).

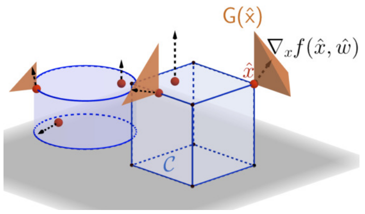


Figure 1. Gradient cones $\hat{x} + G(\hat{x})$ (orange cones) and internal gradients $\nabla_x f(\hat{x}, \hat{w})$ (black arrows) at different points $\hat{x}$ (red dots) in different feasible sets $\mathcal{C}$ (blue cube and cylinder).

function, $\mathcal{C}$ is the feasible set, and w are unknown parameters. P&O employs predictions $\hat{w}$ and the decision can be computed by solving the *internal problem*:

$$x^*(\hat{w}) = \arg\max_x f(x, \hat{w}) \text{ s. t. } x \in \mathcal{C}. \tag{2}$$

As the true parameters w are unknown, we assume that we observe a feature vector o that in some way relates to w. We employ a prediction model ϕ_θ to compute predictions $\hat{w}$, i.e., $\hat{w} = \phi_\theta(o)$. To train ϕ_θ, we have a dataset $\mathcal{D} = \{(o_k, w_k)\}$.

In P&O, the predictive model ϕ_θ is trained to maximize the task performance $f(x, w)$ and *not* to accurately predict the unknown parameters. Therefore, the loss function minimized during training is

$$L(\theta) = -\frac{1}{|\mathcal{D}|} \sum_{(o,w)\in\mathcal{D}} f\Big(x^*\big(\phi_\theta(o)\big), w\Big). \tag{3}$$

To train ϕ_θ with a gradient-based algorithm, we need to differentiate L over θ, and hence we need to compute the gradient $\nabla_\theta f\Big(x^*(\hat{w}), w\Big)$, where $\hat{w} = \phi_\theta(o)$. Applying the chain rule, it can be decomposed into three terms:

$$\nabla_\theta f\big(x^*(\hat{w}), w\big) = \nabla_x f\big(x^*(\hat{w}), w\big)\ \nabla_{\hat{w}} x^*(\hat{w})\ \nabla_\theta \hat{w} \tag{4}$$

The term $\nabla_{\hat{w}} x^*(\hat{w})$ is the Jacobian of the solution of the optimization problem with respect to the prediction $\hat{w}$. In non-linear convex problems, it can be computed exactly [5]. However, properties of this Jacobian are not well studied. In the next section, we show that $\nabla_{\hat{w}} x^*(\hat{w})$ can have a large null space thereby causing suboptimal performance.

4. Differentiable optimization

Without loss of generality, we consider a single instance of the problem, i.e., one sample $(o, w) \in \mathcal{D}$. We denote the prediction by $\hat{w} = \phi_\theta(o)$ and the corresponding decision $\hat{x}$ is then computed as a solution to the internal problem (2). We make the following assumptions:

Assumption 4.1. The objective function $f(x, w)$ is concave and twice continuously differentiable in x for any w.

Assumption 4.2. The feasible set $\mathcal{C} = \{x | g_i(x) \leq 0,\ i = 1, \ldots, l\}$ is convex, i.e., $g_i(x)$ are convex and differentiable. Also, the gradients $\{\nabla_x g_i(x) | g_i(x) = 0\}$ of the active constraints are linearly independent[1] $\forall\ x \in \mathcal{C}$.

Assumption 4.3. The objective function $f(x, w)$ is twice continuously differentiable in w.

Next, we establish some crucial properties of $\nabla_{\hat{w}} x^*(\hat{w})$.

4.1. The zero-gradient theorem

Let $n_i := \nabla_x g_i(\hat{x})$, $i = 1, \ldots, l$ be the normals of the constraints at $\hat{x}$, and let $\alpha_1, \ldots, \alpha_l$ be the KKT multipliers [13]. According to Theorem 2.1 by Fiacco et al. [14], strict complementary slackness (SCS) conditions (i.e., $\alpha_i > 0\ \forall i \in I(x)$) are sufficient for differentiability of $x^*(\hat{w})$. We concentrate on these, since the points violating SCS are a measure-zero set, and can thus be neglected in practice.

First, we obtain a geometrical perspective of the KKT conditions by introducing the following definition:

Definition 4.4. For $x \in \mathcal{C}$ let $I(x) = \{i | g_i(x) = 0\}$ be the set of indices of the constraints active at x. Let $n_i = \nabla_x g_i(x)$ be the normal vectors of these constraints for $i \in I(x)$. The gradient cone, $G(x) := \Big\{\sum_{i\in I} \alpha_i n_i | \alpha_i \geq 0\Big\}$, is the positive linear span of normal vectors n_i.

Combining the KKT conditions with Definition 4.4, we immediately arrive at the following property:

Property 4.5. *Let $x \in \mathcal{C}$ and let $\nabla_x f(x, \hat{w})$ be the internal gradient at x. Then, x is a solution to the problem in Eq.* (2) *if and only if $\forall i \in I(x), \exists \alpha_i \geq 0$, such that $\nabla_x f(x, \hat{w}) = \sum_{i\in I(x)} \alpha_i n_i \in G(x)$, where $I(x)$ is the set of indices of active constraints, $I(x) = \{i | g_i(x) = 0\}$.*

While trivial, this property provides a geometrical interpretation of the problem: a point x is a solution to the problem in Eq. (2) if and only if the internal gradient at this point lies inside its gradient cone. Figure 1 illustrates this property.

Now, we have all the necessary tools to describe the structure of the Jacobian $\nabla_{\hat{w}} x^*(\hat{w})$. Assume that we perturb $\hat{w}$ and obtain $\hat{w}'$. Let $\hat{x}' = x^*(\hat{w}')$. Strict complementary slackness implies that the constraints active at $\hat{x}$ will remain active at $\hat{x}'$ if the difference $\|\hat{w}' - \hat{w}\|_2^2$ is small enough. Therefore, the decision $\hat{x}'$ can only move within the tangent space of $\mathcal{C}$ at $\hat{x}$, i.e., orthogonally to all n_i, $i \in I(\hat{x})$. Hence, when more constraints are active, $\hat{x}'$ can move in less directions. Formally, we obtain the following lemma:

Lemma 4.6. *Suppose that the SCS conditions hold at $\hat{x}$ and let $\nabla_x f(\hat{x}, \hat{w}) = \sum_{i\in I(\hat{x})} \alpha_i n_i$, $\alpha_i > 0$, $\forall i \in I(\hat{x})$ be the internal gradient. Let $\mathcal{N}(\hat{x}) = span(\{n_i \,|\, i \in I(\hat{x})\})$ be the linear span of the gradient cone. Then $\mathcal{N}(\hat{x})$ is contained in the left null space of $\nabla_{\hat{w}} x^*(\hat{w})$.*

The proof can be found in Appendix A. Lemma 4.6 is very important, as it specifies in what directions $x^*(\hat{w})$ *can move* as a consequence of changing $\hat{w}$. The first term in the chain rule in Eq. (4), $\nabla_x f(\hat{x}, w)$, specifies in what directions

[1] For clarity, Assumption 2 does not include equality constraints. In the appendix, we show that all our results hold with those, too.

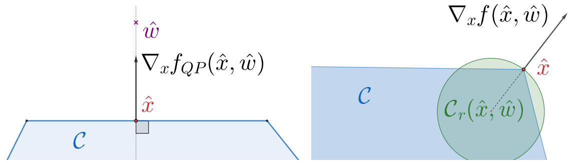

$\hat{w}$
$\nabla_x f_{QP}(\hat{x}, \hat{w})$
$\hat{x}$
$\mathcal{C}$
$\nabla_x f(\hat{x}, \hat{w})$
$\hat{x}$
$\mathcal{C}$
$\mathcal{C}_r(\hat{x}, \hat{w})$

Figure 2. Left: Illustration of the QP approximation. *Right*: Illustration of local smoothing.

$x^*(\hat{w})$ *should* move in order for the true objective to increase. Naturally, if these directions are contained in the null space of $\nabla_{\hat{w}} x^*(\hat{w})$, then the total gradient in Eq. (4) is zero. This constitutes the main result of this paper:

Theorem 4.7 (Zero-gradient theorem)**.** *Let $\hat{w}$ be a prediction, and $\hat{x}$ be the internal problem in Eq. (2). Suppose the SCS conditions hold at $\hat{x}$ and let $\mathcal{N}(\hat{x}) = span(\{n_i \,|\, i \in I(\hat{x})\})$ be the linear span of the gradient cone at $\hat{x}$. Then, $\nabla_x f(\hat{x}, w) \in \mathcal{N}(\hat{x}) \implies \nabla_\theta f(\hat{x}, w) = 0$.*

This theorem claims that the gradient of the P&O loss in Eq. (3) can be zero in points outside the optimal solution. Hence, any gradient-following method "shall not pass" these points. In particular, this phenomenon happens in points $\hat{x}$ where the true gradient $\nabla_x f(\hat{x}, w)$ is contained in the space $\mathcal{N}(\hat{x})$ spanned by the gradient cone $G(\hat{x})$. As the dimensionality of this space grows with the number of active constraints, the zero-gradient issue is particularly important for problems with a large number of constraints. In the worst case, $\mathcal{N}(\hat{x})$ can be as big as the whole decision space $\mathbb{R}^n$, thereby making the total gradient $\nabla_\theta f(\hat{x}, w)$ from Eq. (4) zero for any value of the true gradient $\nabla_x f(\hat{x}, w)$.

4.2. Quadratic programming approximation

The models trained with predict and optimize can output $\hat{w}$ that is significantly different from the true w, yet result in good decisions. Hence, we claim that the objective function $f(x, \hat{w})$ in the internal problem also does not need to be the same as the true objective $f(x, w)$. In particular, we suggest computing decisions using a quadratic program (QP) resembling the method proposed in Wilder et al. [6]:

$$x^*_{QP}(\hat{w}) = \underbrace{\arg\max_x -\|x - \hat{w}\|_2^2}_{f_{QP}(x, \hat{w})} \text{ s.t. } x \in \mathcal{C}. \qquad (5)$$

The main advantage of using QP is that it has the smallest reasonable prediction vector $\hat{w}$ (one scalar per decision variable). Besides, QP approximation can represent any solution, and its Jacobian has a simple analytic form, which allows computing it cheaply and allows for studying its theoretical properties. Specifically, the Jacobian $\nabla_{\hat{w}}\, x^*_{QP}$ is described by the following lemma:

Lemma 4.8. *Let $\{e_j | j = 1, \ldots, n - |I(\hat{x}|)\}$ be an orthogonal complement of vectors $\{n_i | i \in I(\hat{x})\}$ to a basis of $\mathbb{R}^n$. Then, the Jacobian $\nabla_{\hat{w}} x_{QP}(\hat{w})$ in the basis $\{n_i\} \cup \{e_j\}$ is a diagonal matrix. Its first $|I(\hat{x})|$ entries are zero, and the others are one.*

While providing computational benefits, QP approximation does not address the zero-gradient problem. Below, we introduce a smoothing technique that reduces the size of the null space to one.

4.3. Local smoothing

Above, we concluded that the null space of the Jacobian $\nabla_{\hat{w}} x(\hat{w})$ depends on the number of constraints active at $\hat{x}$. Therefore, to solve the zero-gradient problem, we propose a *local smoothing* method, that reduces the number of active constraints to one during the Jacobian computation.

Let $\nabla_x f_{QP}(\hat{x}, \hat{w}) = \sum_{i \in I(\hat{x})} \alpha_i n_i$ be the internal gradient at $\hat{x}$ for some $\alpha_i \geq 0,\ \forall i \in I(\hat{x})$. Then, we introduce the following definition:

Definition 4.9. Let $c = \hat{x} - r\frac{\nabla_x f_{QP}(\hat{x}, \hat{w})}{\|\nabla_x f_{QP}(\hat{x}, \hat{w})\|_2}$, with real $r > 0$, and let $\mathcal{C}_r(\hat{x}, \hat{w}) := \{y | y \in \mathbb{R}^n, \|y - c\|_2 \leq r\}$. The local smoothed problem $P_r(\hat{x}, \hat{w})$ with parameters $\hat{x}, \hat{w}$ is defined as $x^*_r(\hat{w}) := \arg\max_{x \in \mathcal{C}_r(\hat{x}, \hat{w})} f_{QP}(x, \hat{w})$.

Now, let $\hat{x}_r = x^*_r(\hat{w})$ denote the solution of $P_r(\hat{x}, \hat{w})$. By construction, $\hat{x}_r = \hat{x}$. The main purpose of smoothing is to approximate the gradient in Eq. (4) by substituting $\nabla_{\hat{w}} x^*_{QP}(\hat{w})$ with $\nabla_{\hat{w}} x^*_r(\hat{w})$. We emphasize that the decisions are still computed using the non-smoothed problem. Thus, we use the following approximation for Eq. (4):

$$\nabla_\theta f(x^*(\hat{w}), w) \approx \nabla_x f\big(\hat{x}, w\big)\ \nabla_{\hat{w}} x^*_r(\hat{w})\ \nabla_\theta \hat{w} \qquad (6)$$

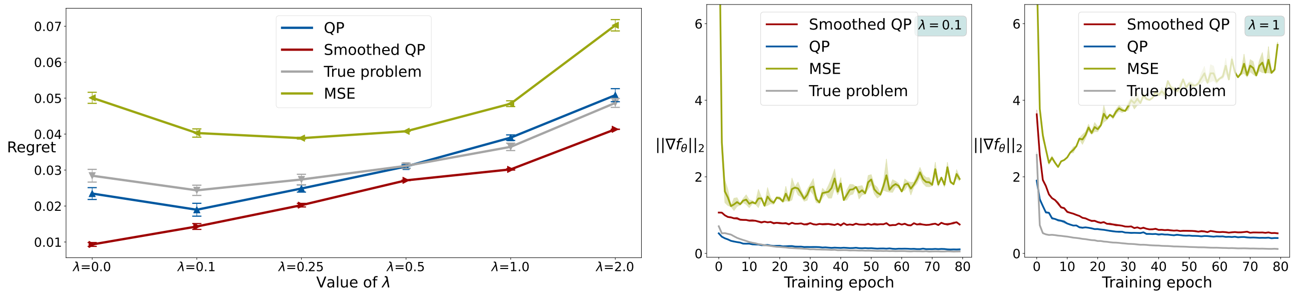
Applying Lemma 4.8, we obtain the following:

Property 4.10. *Let $\hat{x} = x^*_{QP}(\hat{w})$ be a decision derived via QP. Suppose that the SCS conditions hold for $P_r(\hat{x}, \hat{w})$ and let $e_1 = \nabla_x f_{QP}(\hat{x}, \hat{w})$ be the internal gradient. Let $\{e_2, \ldots, e_n\}$ be a complement of e_1 to an orthogonal basis of $\mathbb{R}^n$. Then, the Jacobian $\nabla_{\hat{w}} x^*_r(\hat{w})$ of the locally smoothed problem in the basis $\{e_1, e_2, \ldots, e_n\}$ is a diagonal matrix. Its first entry is zero, others are ones.*

Using this property, we can show that the smoothed Jacobian is consistent with the loss function:

Theorem 4.11. *Let $\hat{x} = x^*_{QP}(\hat{w})$ be the decision obtained via QP and let $\nabla_{\hat{w}} x^*_r(\hat{w})$ be the Jacobian of the local smoothed QP problem. Let $\Delta\hat{w} = \nabla_x f(\hat{x}, w)\ \nabla_{\hat{w}} x^*_r(\hat{w})$ be the prediction perturbation obtained by using this Jacobian and let $\hat{w}'(t) = \hat{w} + t\Delta\hat{w}$ be the updated prediction. Then, for $t \to 0^+$, using $\hat{w}'(t)$ results in a non-decrease in the task performance. In other words, $f\big(x^*_{QP}(\hat{w}'(t)), w\big) \geq f\big(x^*_{QP}(\hat{w}), w\big)$.*

As we see from Property 4.10, the value of r does not affect the Jacobian. We keep it only for clarity of the notation. As $C_r(\hat{x}, \hat{w})$ is defined by a single constraint, the null space of $\nabla_{\hat{w}} x^*_r(\hat{x}, \hat{w})$ is always one-dimensional. Hence,



(a)

(b)

Figure 3. Results on the portfolio optimization problem: (a) the final test regret for each of the algorithms for varying λ's. *(b)* Evolution of the l_2 norm of the gradient during training for $\lambda = 0.1$ and $\lambda = 1$.

the zero-gradient problem only occurs when the internal gradient $\nabla_x f_{QP}(\hat{x}, \hat{w})$ and the true gradient $\nabla_x f(\hat{x}, w)$ are collinear. To deal with this case, we use the projection distance regularization method first suggested in [7]. Specifically, we add a penalty $p(\hat{w}) = \alpha\|\hat{x} - \hat{w}\|_2^2$, where $\alpha \in \mathbb{R}^+$ is a hyperparameter. Minimizing this term, we push $\hat{w}$ along the null space of the Jacobian towards the feasible set and eventually move $\hat{x}$ inside $\mathcal{C}$.

5. Experiments

We consider the portfolio optimization problem [8], where we aim to maximize the immediate return but minimize the risk penalty under the budget constraint:

$$\arg\max_x \underbrace{p^\top x - \lambda\, x^\top Q x}_{f(x,p,Q)} \quad \text{s. t.} \quad \sum_{i=1}^{n} x_i = 1,\ x \geq 0. \quad (7)$$

The decision $x \in \mathbb{R}^n$ is the investment, $p \in \mathbb{R}^n$ is the immediate return, and $Q \in \mathbb{R}^{n \times n}$ is the covariance matrix. The unknown parameters are defined as $w := (p, Q)$ and $\lambda \geq 0$ represents the risk-aversion weight. We use historical data from QUANDL WIKI [15] and we refer the readers to the appendix for more details. We consider different values of λ from the set $\{0, 0.1, 0.25, 0.5, 1, 2\}$, in order to obtain a spectrum of problems, from the linear ($\lambda = 0$) to the "strongly quadratic" ($\lambda = 2$).

We compare the performance of four methods: minimization of the MSE of the prediction (labeled "MSE"); differentiation of the original problem in Eq (7) ("True problem"); differentiation of the QP approximation ("QP"); and combination of QP, smoothing, and projection distance regularization ("Smoothed QP"). For the performance metric, we use *regret* [1], defined as

$$\text{regret}(o, w) = f\Big(x^*\big(\phi_\theta(o)\big), w\Big) - \max_x f(x, w). \quad (8)$$

The results in Figure 3 demonstrate that the smoothed QP approach is dominant – it outperforms the competitors by a significant margin across all values of λ. Figure 3 (b) suggests the reason for this result is indeed the zero-gradient problem: for the methods using the exact Jacobian (QP and true problem), the gradient norm decreases rapidly with training. In accordance with theory, this effect is more significant for smaller values of λ. In the more quadratic cases, the relative difference in performance becomes smaller.

	Regret	Runtime (sec)
True problem	0.834 ± 0.120	7965 ± 52
QP	0.506 ± 0.009	762 ± 52
Smoothed QP	**0.438 ± 0.009**	801 ± 54

Table 1. Final (normalized) test regret and training time for the different methods on the LogSumExp portfolio problem.

Figure 3 suggests that the QP approximation is sufficient in portfolio optimization. However, it might be explained by the fact that the true problem in Eq. (7) is also quadratic. To gain more insight into the performance of our method, we also consider a *LogSumExp* objective function:

$$f_{lse}(x, p, Q) = -\log \sum_i e^{-p_i x_i} \quad (9)$$

The LogSumExp function acts as a soft maximum, and the corresponding problem can be interpreted as maximization of the most profitable investment. The results in Table 1 demonstrate that the QP approximation outperforms the true problem in terms of regret and significantly reduces the computation time. Moreover, smoothed QP again outperforms the other approaches, which suggests that the zero-gradient problem occurs in the LogSumExp case as well.

6. Conclusions

In this work, we theoretically demonstrate that the zero-gradient problem in the non-linear convex P&O setting extends beyond the linear case. To address this, we introduce a Jacobian approximation method by smoothing the feasible set, thereby reducing the null space's dimensionality to one. This approach, combined with ideas from prior work, enables effective gradient updates and escapes from zero-gradient cones. Our experiments with portfolio optimization problem confirm that the zero-gradient issue impedes standard differential optimization and show that our smoothed QP method solves it effectively.

References

[1] A. N. Elmachtoub and P. Grigas, “Smart” predict, then optimize”,” *arXiv preprint arXiv:1710.08005*, 2017.

[2] J. Kiefer and J. Wolfowitz, “Stochastic estimation of the maximum of a regression function,” *The Annals of Mathematical Statistics*, pp. 462–466, 1952.

[3] M. Vlastelica, A. Paulus, V. Musil, G. Martius, and M. Rolínek, “Differentiation of blackbox combinatorial solvers,” pp. 1–19, 2019.

[4] S. S. Sahoo, A. Paulus, M. Vlastelica, V. Musil, V. Kuleshov, and G. Martius, *Backpropagation through combinatorial algorithms: Identity with projection works*, 2023. arXiv: 2205.15213 [cs.LG].

[5] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, “Differentiable convex optimization layers,” *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[6] B. Wilder, B. Dilkina, and M. Tambe, “Melding the Data-Decisions pipeline: Decision-Focused learning for combinatorial optimization,” en, *AAAI*, vol. 33, no. 01, pp. 1658–1665, Jul. 2019.

[7] B. Chen, P. L. Donti, K. Baker, J. Z. Kolter, and M. Bergés, “Enforcing policy feasibility constraints through differentiable projection for energy optimization,” *e-Energy 2021 - Proceedings of the 2021 12th ACM International Conference on Future Energy Systems*, pp. 199–210, 2021.

[8] K. Wang, B. Wilder, A. Perrault, and M. Tambe, “Automatically learning compact quality-aware surrogates for optimization problems,” Jun. 2020. arXiv: 2006.10815 [cs.LG].

[9] Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J. P. Vert, and F. Bach, “Learning with differentiable perturbed optimizers,” *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, pp. 1–24, 2020.

[10] M. Grant, S. Boyd, and Y. Ye, “Disciplined Convex Programming,” in *Global Optimization: From Theory to Implementation*, L. Liberti and N. Maculan, Eds. Boston, MA: Springer US, 2006, pp. 155–210, ISBN: 978-0-387-30528-8.

[11] A. S. Uysal, X. Li, and J. M. Mulvey, “End-to-End risk budgeting portfolio optimization with neural networks,” Jul. 2021. arXiv: 2107.04636 [q-fin.PM].

[12] J. Mandi and T. Guns, “Interior point solving for LP-based prediction+optimisation,” Oct. 2020. arXiv: 2010.13943 [cs.NE].

[13] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” *In Berkeley Symposium on Mathematical Statistics and Probability*, vol. 2, pp. 481–492, 1951.

[14] A. V. Fiacco, “Sensitivity analysis for nonlinear programming using penalty methods,” *Mathematical programming*, vol. 10, no. 1, pp. 287–311, 1976.

[15] QUANDL, *Quandl wiki prices, 2020.* 2020.

A. Proofs

Proof of Lemma 4.6. Let $\Delta\hat{w}$ denote an arbitrary direction and let $d = \nabla_{\hat{w}}\, x^*(\hat{w})\, \Delta\hat{w}$ be the corresponding directional derivative of the decision. The existence of d is guaranteed by the strict complementary slackness conditions. Let $t \to 0^+$. Then, we have

$$\hat{x}'(t) := x^*(\hat{w} + t\Delta\hat{w}) = \hat{x} + td + o_x(t),$$

where $o_x(t)$ is the “little o” notation, i.e., $\lim_{t\to 0^+} \frac{\|o_x(t)\|_2}{t} = 0$. To prove the lemma, we first want to show that $d^\top n_i = 0,\ \forall i \in I(\hat{x})$. Then, we will show that it implies the lemma’s claim.

By definition, $n_i = \nabla_x g_i(\hat{x})$. Then, since $g_i(\cdot)$ is differentiable and $g_i(\hat{x}) = 0,\ \forall i \in I(\hat{x})$, we have the following first-order approximation for $g_i\big(\hat{x}'(t)\big)$:

$$g_i\big(\hat{x}'(t)\big) = g_i\big(\hat{x} + td + o(t)\big) =$$
$$g_i(\hat{x}) + tn_i^\top d + o_g(t) = tn_i^\top d + o_g(t).$$

Since $\hat{x}'$ is the solution of the internal optimization problem, the inequality $g_i(\hat{x}'(t)) \le 0$ holds. Hence, the equation above implies that $n_i^\top d \le 0$. Now, we want to show that, in fact, $n_i^\top d = 0$. For a proof by contradiction, suppose that $n_i^\top d < 0$. Then, by definition of $o_g(t)$, there exists $\epsilon > 0$, such that

$$0 < t < \epsilon \implies g_i\big(\hat{x}'(t)\big) < 0.$$

Now, we will to show that $g_i\big(\hat{x}'(t)\big) < 0$ contradicts the complementary slackness condition at $\hat{x}$. From the work by Fiacco et al. [14], we know that the KKT multiplier, $\alpha_i'(t) := \alpha_i(\hat{w} + t\Delta\hat{w})$, is a continuous function of t. On the one hand, from the KKT conditions, we know that $g_i\big(\hat{x}'(t)\big) < 0 \implies \alpha_i'(t) = 0$. Therefore, $\alpha_i'(t) = 0$ for $t < \epsilon$. Hence, we have

$$\lim_{t\to 0^+} \alpha_i'(t) = 0.$$

On the other hand, the continuity implies that $\lim_{t\to 0^+} \alpha_i'(t) = \alpha_i'(0) = \alpha_i$ and, due to strict complementary slackness, $\alpha_i > 0$. Hence, we also have

$$\lim_{t\to 0^+} \alpha_i'(t) > 0.$$

We arrived at a contradiction and therefore can claim that $d^\top n_i = 0$ for all n_i. Since $\{n_i | i \in I(\hat{x})\}$ is a basis of $\mathcal{N}(\hat{x})$, this implies that for any direction $v \in \mathcal{N}(\hat{x})$ and for any $\Delta\hat{w}$, we have $v^\top\, \nabla_{\hat{w}}\, x^*(\hat{w})\, \Delta\hat{w} = 0$. In other words, vector $v^\top\, \nabla_{\hat{w}}\, x^*(\hat{w})$ is orthogonal to the whole space of $\hat{w}$ and hence it must be zero, $v^\top\, \nabla_{\hat{w}}\, x^*(\hat{w}) = 0,\ \forall v \in \mathcal{N}(\hat{x})$. Hence $\mathcal{N}(\hat{x})$ is contained in the left null space of $\nabla_{\hat{w}}\, x^*(\hat{w})$. □

Proof of Lemma 4.8. First, consider the case when the unconstrained maximum $\hat{w}$ is in the interior of $\mathcal{C}$. By definition of x^*_{QP}, it means that $\hat{x} = x^*_{QP}(\hat{w})$ is also in the interior of $\mathcal{C}$ and $\hat{x} = \hat{w}$. Then, x^*_{QP} is the identity function around $\hat{w}$, and hence $x^*_{QP}(\hat{w} + \Delta\hat{w}) = x(\hat{w}) + \Delta\hat{w}$ for small enough $\Delta\hat{w}$. Hence, $\nabla_{\hat{w}} x^*_{QP}(\hat{w}) = I$. Since no constraints are active in this case ($I(\hat{x}) = \emptyset$), the lemma’s claim holds.

Now, consider the case when some constraints are active, and thus $\hat{x}$ lies on the boundary of $\mathcal{C}$. To get the exact form of the Jacobian $\nabla_x\, x^*_{QP}(\hat{w})$, we will compute $\lim_{t\to 0} x^*_{QP}(\hat{w} + t\Delta\hat{w})$ for all possible $\Delta\hat{w}$. As in the QP case the predictions $\hat{w}$ lie in the same space as $\hat{x}$, we can do it first for $\Delta\hat{w} \in \mathcal{N}(\hat{x})$ and then for $\Delta\hat{w} \perp \mathcal{N}(\hat{x})$.

1. $\Delta\hat{w} \in \mathcal{N}(\hat{x})$. For $\Delta\hat{w} \in \mathcal{N}(\hat{x})$, we want to show that the corresponding directional derivative is zero. We begin by computing the internal gradient $\nabla_x f_{QP}(\hat{x}, \hat{w})$:

$$\nabla_x f_{QP}(\hat{x}, \hat{w}) = -\nabla_x \|x - w\|_2^2 = 2(\hat{w} - \hat{x}).$$

Using this formula, we can write the internal gradient for the perturbed prediction $\hat{w} + t\Delta\hat{w}$ at the same point $\hat{x}$:

$$\nabla_x f_{QP}(\hat{x}, \hat{w} + t\Delta\hat{w}) = \nabla_x f_{QP}(\hat{x}, \hat{w}) + 2t\Delta\hat{w}.$$

By definition, $\mathcal{N}(\hat{x})$ is a linear span of the vectors $\{n_i | i \in I(\hat{x})\}$. Hence, since $\Delta\hat{w} \in \mathcal{N}(\hat{x})$, it can be expressed as

$$\Delta\hat{w} = \sum_{i\in I(\hat{x})} \delta_i n_i, \quad \delta_i \in \mathbb{R}. \tag{$*$}$$

By Property 4.5, the internal gradient has the following representation:

$$\nabla_x f_{QP}(\hat{x}, \hat{w}) = \sum_{i\in I(\hat{x})} \alpha_i n_i, \quad \alpha_i > 0. \tag{$**$}$$

Then, combining $(*)$ and $(**)$, we obtain

$$\nabla_x f_{QP}(\hat{x}, \hat{w} + t\Delta\hat{w}) = \nabla_x f_{QP}(\hat{x}, \hat{w}) + 2t\Delta\hat{w} = \sum_{i \in I(\hat{x})} (\alpha_i + 2t\delta_i) n_i$$

Since $\alpha_i > 0$, $\forall i \in I(\hat{x})$, there exists $\epsilon > 0$, such that $\alpha_i - 2t\delta_i > 0$ for $|t| < \epsilon$. Therefore, $\nabla_x f_{QP}(\hat{x}, \hat{w} + t\Delta\hat{w})$ lies in the gradient cone of $\hat{x}$, and hence, by Property 4.5, $x^*_{QP}(\hat{w} + t\Delta\hat{w}) = \hat{x}$ for $|t| < \epsilon$. Therefore, the directional derivative of $x^*_{QP}(\hat{w})$ along $\Delta\hat{w} \in \mathcal{N}(\hat{x})$ is zero.

2. $\Delta\hat{w} \perp \mathcal{N}(\hat{x})$. Next, let $\Delta\hat{w}$ be orthogonal to $\mathcal{N}(\hat{x})$. We begin with the first order approximation of $\hat{x}'(t)$:

$$\hat{x}'(t) = \hat{x} + td + o(t).$$

From the proof of Lemma 4.6, we know that $d \perp \mathcal{N}$. By definition of x^*_{QP}, we know that $\hat{x}$ is the point on $\mathcal{C}$ closest to $\hat{w}$. Likewise, $\hat{x}'(t)$ is the point on $\mathcal{C}$ closest to $\hat{w} + t\Delta\hat{w}$. Hence, $d = \Delta\hat{w}$. Therefore, for any $\Delta\hat{w} \perp \mathcal{N}$, the directional derivative of $x_{QP}(\hat{w})$ along $\Delta\hat{w}$ is one.

So, we have shown that

$$\nabla_{\hat{w}}\, x^*_{QP}(\hat{w})\, \Delta\hat{w} = \begin{cases} 0 & \text{for } \Delta\hat{w} \in \mathcal{N}(\hat{x}) \\ \Delta\hat{w} & \text{for } \Delta\hat{w} \perp \mathcal{N}(\hat{x}). \end{cases}$$

Therefore, the lemma is proven. □

Proof of Theorem 4.11. First, we want to construct an orthogonal basis $\{e_1, \dots e_n\}$ of $\mathbb{R}^n$ that will greatly simplify the calculations. We start by including the internal gradient in this basis, i.e., we define $e_1 = \nabla_x f_{QP}(\hat{x}, \hat{w})$. Then, let $I(\hat{x}) = \{i | g_i(\hat{x}) = 0\}$ be the set of indices of the active constraints of the original problem and let $\mathcal{N}(\hat{x}) = span(\{n_i | i \in I(\hat{x})\})$ be a linear span of their normals. By the liner independence condition from Assumption 2, $dim\big(\mathcal{N}(\hat{x})\big) = |I(\hat{x})|$. Moreover, by Property 4.5, we know that $e_1 \in \mathcal{N}(\hat{x})$. Then, we can choose vectors $e_2, \dots, e_{|I(\hat{x})|}$ that complement e_1 to an orthogonal basis of $\mathcal{N}(\hat{x})$. The remaining vectors $e_{|I(\hat{x})|+1}, \dots, e_n$, are chosen to complement $e_1, \dots, e_{|I(\hat{x})|}$ to an orthogonal basis of $\mathbb{R}^n$. The choice of this basis is motivated by Lemma 6: e_1 is a basis of the null-space of the $r-$smoothed Jacobian, $e_1, \dots, e_{|I(\hat{x})|}$ form a basis of the null space of the true QP Jacobian, and the remaining vectors form a basis of space in which we can move $x^*_{QP}(\hat{w})$.

For brevity, let $f_x = \nabla_x f(\hat{x}, w)$ denote the true gradient vector. By definition, $\Delta\hat{w} = f_x\, \nabla_{\hat{w}} x^*_r(\hat{x}, \hat{w})$ is obtained via the $r-$smoothed problem. From Property 4.10, we know that $\Delta\hat{w}$ is a projection of f_x on the vectors $e_2, \dots, e_n$. Then, since $e_1, \dots, e_n$ is an orthogonal basis, we have

$$\Delta\hat{w} = \sum_{i=2}^{n} \beta_i e_i, \quad \beta_i = f_x^\top e_i,\ i = 2, \dots, n.$$

Now, let's see how this $\Delta\hat{w}$ affects the true decision $x^*_{QP}(\hat{w} + t\Delta\hat{w})$ for $t \to 0^+$. First, we have a first-order approximation

$$x^*_{QP}(\hat{w} + t\Delta\hat{w}) = \hat{x} + td + o(t),$$

for some $d \in \mathbb{R}$. From Lemma 4.6, we know that d is actually a projection of $\Delta\hat{w}$ onto the vectors $e_{|I(\hat{x})|+1}, \dots, e_n$. Therefore, we have

$$x^*_{QP}(\hat{w} + t\Delta\hat{w}) = \hat{x} + \sum_{i=|I(\hat{x}|+1}^{n} \beta_i e_i + o(t).$$

Finally, the change in the true objective can be expressed as

$$f\Big(x^*_{QP}(\hat{w} + t\Delta\hat{w}), w\Big) - f\Big(x^*_{QP}(\hat{w}), w\Big) = t f_x^\top \Big(\sum_{i=|I(\hat{x}|+1}^{n} \beta_i e_i \Big) + o(t) =$$

$$= t \sum_{i=|I(\hat{x}|+1}^{n} \beta_i f_x^\top e_i + o(t) = t \sum_{i=|I(\hat{x}|+1}^{n} \beta_i^2 + o(t) \geq 0.$$

Therefore, perturbing prediction along $\Delta\hat{w}$ does not decrease the true objective $f(\hat{x}, w)$, and hence

$$f\big(x^*_{QP}(\hat{w} + t\Delta\hat{w}), w\big) \geq f\big(x^*_{QP}(\hat{w}), w\big)$$

for $t \to 0^+$. □

Parameter	Search space	Best value
Learning rate	$\{0.5, 1, 5, 10\} \times 10^{-5}$	5×10^{-5}
Training epochs	$\{40, 80, 160\}$	80
Batch size	$\{1, 4, 8, 32\}$	1
x_{shift}	$\{0, 0.1, 1\}$	0.1
x_{scale}	$\{0.1, 1\}$	1

Table 2. Hyperparameters for methods from Figure 3 for standard portfolio optimization problem with different λ's.

B. Equality constraints

Assumption 2 postulates that for any $x \in \mathcal{C}$, the gradients of active constraints, $\{\nabla_x g_i(x) | g_i(x) = 0\}$, are linearly independent. Now, suppose we include equality constraints in our problem. e.g., we have a constraint $g^{eq}(x) \leq 0$ and $-g^{eq}(x) \leq 0$ for some g. Clearly, the gradients of $g^{eq}(x)$ and $-g^{eq}(x)$ violate the independence assumption. However, we claim that it does not affect our results. Let $\hat{w}$ and $\hat{x}$ be a prediction and a corresponding decision and let $n^{eq} = \nabla_x \, g^{eq}(\hat{x})$. Suppose the equality constraint $g^{eq}(\hat{x}) = 0$ is active. Let $I(\hat{x})$ be the set of indices of the active constraints *not including* $g^{eq}(x)$. Then, we have a representation of the internal gradient,

$$\nabla_x f(\hat{x}, \hat{w}) = \alpha_1^{eq} n^{eq} - \alpha_2^{eq} n^{eq} + \sum_{i \in I(\hat{x})} \alpha_i n_i.$$

Suppose that $\alpha_1^{eq} \neq \alpha_2^{eq}$, e.g., without loss of generality, $\alpha_1^{eq} > \alpha_2^{eq}$. Then,

$$\nabla_x f(\hat{x}, \hat{w}) = (\alpha_1^{eq} - \alpha_2^{eq}) n^{eq} + \sum_{i \in I(\hat{x})} \alpha_i n_i$$

and hence removing the constraint $-g^{eq}(x) \leq 0$ would not change the optimality of $\hat{x}$. The remaining problem would satisfy complementary slackness and hence would have all the properties demonstrated in Section 3. Therefore, for the case with equality constraints, we need to extend the complementary slackness conditions by demanding $\alpha_1^{eq} \neq \alpha_2^{eq}$.

C. Experimental details

In this section, we provide the details of the experiments reported in the paper. All experiments were conducted on a machine with 32gb RAM and NVIDIA GeForce RTX 3070. The code is written in Python 3.8, and neural networks are implemented in *PyTorch* 1.11. For methods requiring differentiation of optimization problems, we use the implementation by Agrawal et. al [2019].

Experimental results reported in Figures 3 and 4 show the average and the standard deviation (shaded region) of the measured quantities across 4 random seeds. For each seed, we randomly split data into train, validation, and test sets by using 70%, 20%, and 10% of the whole dataset respectively. In Figure 3a, for each method and at each run, we take the model version corresponding to the best performance on the validation set and report its performance on the test set. In Figure 4, we do the same procedure at each training iteration.

In all experiments and methods, the predictor ϕ_θ is represented by a fully connected neural network with two hidden layers of 256 neurons each, and *LeakyReLU* activation functions. The output layer has no activation function. Instead, the output of the neural network is scaled by the factor x_{scale} and shifted by x_{shift}. For the methods using QP approximation, the output layer predicts vector $\hat{w}$ of the same dimensionality as the decision variable. For the method using the true model, the prediction size is defined by the number of unknown parameters in the true objective function. For training, we used the *Adam* optimizer from PyTorch, with custom learning rate and otherwise default parameters.

Hyperparameters of all methods were chosen based on the results of the grid search reported in Tables 2-3. The weight α of the projection distance regularization term is $\lambda-$dependent. Specifically, $\alpha = 0$ for $\lambda \in \{0, 0.1\}$, $\alpha = 0.01$ for $\lambda \in \{0.25, 0.5\}$, and $\alpha = 0.1$ for $\lambda \in \{1, 2\}$. Configuration files to reproduce the experiments and the code can be found at *placeholder for GitHub link.*

Portfolio optimization problem

Following Wang et. al [2020], we use historical data from QUANDL WIKI prices [15] for 505 largest companies on the American market for the period 2014-2017. The dataset is processed and for every day we obtain a feature vector

Parameter	Search space	Best value
Learning rate	$\{0.5, 1, 5, 10\}\times 10^{-5}$	5×10^{-5}
Batch size	$\{1, 4, 8, 32\}$	1
x_{shift}	$\{0, 0.1, 1\}$	0.1
x_{scale}	$\{0.1, 1\}$	0.1

Table 3. Hyperparameters for methods from Table 1 for LogSumExp portfolio optimization problem.

summarizing the recent price dynamic. For further details on the processing, we refer readers to the code and to Wang et. al [2020]. The processed dataset contained historical data describing the past price dynamics for each of the 505 securities. For every random seed, 50 securities (thus, 50 decision variables) were chosen randomly. The experiments on the LogSumExp variation of the portfolio optimization problem were conducted similarly. The hyperparameters for normal and LogSumExp portfolio problems are reported in Tables 2 and 3.