The Perils of Learning Before Optimizing

Chris Cameron[1], Jason Hartford[2], Taylor Lundy[1], Kevin Leyton-Brown[1]

[1]Department of Computer Science, University of British Columbia
[2]Mila, Université de Montréal
{cchris13, tlundy, kevinlb}@cs.ubc.ca, jason.hartford@mila.quebec

Abstract

Formulating real-world optimization problems often begins with making predictions from historical data (e.g., an optimizer that aims to recommend fast routes relies upon travel-time predictions). Typically, learning the prediction model used to generate the optimization problem and solving that problem are performed in two separate stages. Recent work has showed how such prediction models can be learned end-to-end by differentiating through the optimization task. Such methods often yield empirical improvements, which are typically attributed to end-to-end making better error tradeoffs than the standard loss function used in a two-stage solution. We refine this explanation and more precisely characterize when end-to-end can improve performance. When prediction targets are stochastic, a two-stage solution must make an a priori choice about which statistics of the target distribution to model—we consider expectations over prediction targets—while an end-to-end solution can make this choice adaptively. We show that the performance gap between a two-stage and end-to-end approach is closely related to the *price of correlation* concept in stochastic optimization and show the implications of some existing POC results for the predict-then-optimize problem. We then consider a novel and particularly practical setting, where multiple prediction targets are combined to obtain each of the objective function's coefficients. We give explicit constructions where (1) two-stage performs unboundedly worse than end-to-end; and (2) two-stage is optimal. We use simulations to experimentally quantify performance gaps and identify a wide range of real-world applications from the literature whose objective functions rely on multiple prediction targets, suggesting that end-to-end learning could yield significant improvements.

1 Introduction

It is increasingly common to face optimization problems whose inputs must be learned from data rather than being given directly. For example, consider a facility location application in which loss depends on predictions about both traffic congestion and demand. The most natural way to address such problems is to separate prediction from optimization into two separate stages (Yan and Gregory 2012; Centola 2018; Bahulkar et al. 2018). In the first stage, a model is trained to optimize some standard loss function (e.g., predict travel times to minimize mean squared error). In the second stage, the model predictions are used to parameterize an optimization problem, which is then solved (e.g., recommend facility locations given demand and travel time predictions).

Recently there has been a lot of interest in the "end-to-end" approach of training a predictive model to minimize loss on the downstream optimization task (Donti, Amos, and Kolter 2017; Demirović et al. 2019; Wilder, Dilkina, and Tambe 2019; Wilder et al. 2020). Advances in deep learning have enabled any set of differentiable functions to be chained together, allowing for the end-to-end training of a differentiable prediction task coupled with a differentiable downstream task using back propagation. Amos and Kolter (2017) introduced the concept of optimization as a layer, showing how to analytically compute gradients through a QP solver. This sparked a number of follow up papers showing how to build differentiable layers for different classes of optimization problems (e.g., submodular optimization (Djolonga and Krause 2017), linear programming (Wilder, Dilkina, and Tambe 2019), general cone programs (Agrawal et al. 2019), and disciplined convex programs (Agrawal et al. 2020)).

Such "end-to-end" approaches have been shown to improve performance on a diverse set of applications (e.g., inventory stocking (Amos and Kolter 2017), bipartite matching (Wilder, Dilkina, and Tambe 2019), and facility location (Wilder et al. 2020)). The improvements are often attributed to the end-to-end approach having made better error tradeoffs than the two-stage approach. For example, Donti, Amos, and Kolter (2017) argue that since all models inevitably make errors, it is important to look at a final task-based objective; Wilder, Dilkina, and Tambe (2019) suggest that end-to-end optimization is likely to be especially beneficial for difficult problems where the best model is imperfect (e.g., when either model capacity or data is limited); and Elmachtoub and Grigas (2020) argue for the effectiveness of end-to-end when there is model misspecification and perfect prediction is unattainable. One might therefore assume that end-to-end learning offers no benefit in an "error-free" setting where we have access to the Bayes optimal predictor for a given loss function.

This paper shows otherwise: that even when there is no limit on model capacity or training set size, the two-stage approach can fail catastrophically in the common case where the prediction stage models expectations over prediction targets. In contrast, end-to-end approaches can adaptively learn statistics of the joint distribution that matter most for the downstream task. This gives them a clear advantage when the joint distribution is not well approximated by the product distribution of its marginal probabilities. We formalize this advantage by drawing a connection to the *price of correlation* concept in stochastic optimization (Agrawal et al. 2012). The price of correlation compares the performance of an optimal solution of a stochastic program to the approximate solution that treats every input as independent. We show that in stochastic optimization settings where we can prove that end-to-end is optimal, we can leverage price-of-correlation bounds to obtain worst-case lower bounds on the gap between a two-stage and end-to-end approach. Unfortunately, we also show that end-to-end is not optimal for all stochastic optimization settings.

We then consider a novel setting in which predict-then-optimize is a common paradigm and end-to-end learning is particularly relevant: where multiple prediction targets are combined to obtain each of the objective function's coefficients. We show that this setting can give rise to potentially unbounded performance gaps between the end-to-end and the two-stage approaches. The key properties we leverage to show a gap are nonlinearity in the objective function's dependence on the prediction targets and correlation structure across targets. All of our results consider a class of problems where end-to-end performs just as well as a stochastic optimization oracle. We make three main contributions: (1) we establish a base case by showing that when the parameterization is linear, the two-stage approach is optimal; (2) we construct a distribution where the gap is unbounded for a pairwise product parameterization function that occurs commonly in practice; and (3) we show how to construct a distribution for any nonlinear parameterization such that there is a gap.

These results highlight two main features of an optimization problem that together can lead to suboptimal two-stage solutions, and that practitioners should look out for in practice. The first is that multiple targets are learned independently, without accounting for correlations. This is likely to happen when the targets are conceptually different and tend to be learned separately, but may be correlated via latent factors that affect them both. The second is that targets are combined nonlinearly to compute coefficients of the objective function for downstream optimization. This is quite common in practice, notably where the cost of some decision is the product of two inputs (e.g., total cost = cost/unit · #units).

Through a running example of the demand-weighted facility location problem, we (1) provide intuition for how such correlations might arise in practice and (2) experimentally explore how correlation impacts the performance gap between a two-stage and end-to-end approach. We design a synthetic benchmark that controls the level of correlation between travel times and demand. We assume that a two-stage approach is able to optimally predict both demand and travel time, but show that it increasingly underperforms relative to the end-to-end approach as the correlation between demand and travel time increases.

Although—particularly given our way of framing the problem—the reader might feel that two-stage learning is obviously problematic in such domains, we illustrate that two-stage learning has been widely used in real-world optimization problems likely to satisfy these conditions. Unfortunately, raw data was unavailable for any of the published real-world application work we reviewed. We thus support our argument by describing this work where the parameterization involves pairwise products of the targets (Mauttone and Urquhart 2009; Zhen, Zhuge, and Lei 2016; Alonso-Mora, Wallar, and Rus 2017; Sert et al. 2020), and identifying plausible correlation structures between targets that would induce practical gaps between two-stage and end-to-end performance.

The remainder of this paper is organized as follows. In Section 2, we survey the literature. In Section 3, we give our results characterizing a gap between two-stage and end-to-end in an error-free setting. In Section 4, we perform a simulation study analyzing how correlation impacts the performance gap. In Section 5, we discuss examples from the literature where end-to-end learning could yield improvements. Finally, we summarize our contributions in Section 6.

2 Related Work

To fully exploit the power of gradient-based learning methods, the end-to-end learning philosophy advocates for jointly learning all the parameters between the raw input and the final outputs of the application. End-to-end models are typically learned with generic architectures that are extremely flexible and have been shown to be effective for a wide variety of applications.

However, leveraging knowledge about the underlying task can also provide a useful inductive bias so the appropriate structure does not need to be discovered from scratch. In the predict-then-optimize setting, we know that the downstream task involves solving a specific optimization problem. Instead of using a generic method to learn to predict the solution to the optimization problem, the end-to-end learning setup learns the target parameters and relies on existing solvers to find associated solutions. This approach can be optimized end to end with gradient-based methods if we can compute gradients through the underlying optimization problem.

There is a significant body of recent work that builds differentiable optimization solvers for various classes of optimization problems. Amos and Kolter (2017) introduced OptNet, an optimization layer that computes gradients for a quadratic program (QP) solver. Djolonga and Krause (2017) and Tschiatschek, Sahin, and Krause (2018) showed how to differentiate through submodular optimization. Barratt (2018) and Agrawal et al. (2019) later developed a differentiable layer for solving cone programs and Agrawal et al. (2020) developed a differentiable solver for disciplined convex programs along with a rich implementation leveraging the `cvxpy` python package. Wilder, Dilkina, and Tambe (2019) showed how to differentiate through linear programs. They added a quadratic regularization term to the objective function to ensure it is everywhere differentiable. Ferber et al. (2020) expanded on this work to build an approximate differentiable layer for solving linear programs with integer constraints (mixed-integer programs). They approximately differentiated through a mixed-integer program by relaxing to a linear program and adding additional constraints via cutting plane methods to push the optimum of the linear program close to the integer-optimal solution. Motivated by the fact that running these optimization problems can be computationally expensive, Elmachtoub and Grigas (2020) and Wilder et al. (2020) developed less computationally-demanding approaches. Elmachtoub and Grigas (2020) developed a cheaper, convex surrogate optimization solver to approximate polyhedral and convex problems with a linear objective. Wilder et al. (2020) found a cheaper proxy optimization problem that is structurally similar to the true optimization problem. They created a differentiable layer for k-means clustering as a proxy for solving structurally-related-combinatorial-optimization problems like facility location that are typically solved via linear or integer programs.

These methods developed the machinery for differentiating through optimization routines and demonstrated their benefits experimentally. Our main contribution is to provide explicit worst-case lower bounds on the performance gap between end-to-end and two-stage solutions. In doing so, we refine the standard explanation for why end-to-end approaches can outperform two-stage methods: most literature talks about end-to-end learning making the right error trade-offs, but here we show that even when one uses the optimal model implied by the first-stage loss, a mismatch between the first-stage loss and the optimization task of interest can lead to unbounded gaps in performance.

With respect to nonlinearity, Elmachtoub and Grigas (2020) showed empirical evidence that end-to-end learning performs better relative to a two-stage approach with increasing non-linearity between the features and prediction targets. Our results leverage a different form of nonlinearity where the objective function is a nonlinear function of prediction targets.

Our results build on an insight from the stochastic optimization literature that the gap between the optimal solution to a given stochastic program over correlated variables, and an approximate solution that treats each variable independently, can be bounded in terms of the *price of correlation* (Bertsimas, Natarajan, and Teo 2004; Agrawal et al. 2012). Agrawal et al. (2012) recognized that in practice, estimating correlations is usually much harder than estimating the mean. They investigated the possible loss incurred by solving a stochastic optimization problem when the correlations are not known. We show for many settings studied in the optimization literature that end-to-end solutions can achieve the stochastic optimum, and as a result, they represent a useful alternative to approximate solutions even in settings with small price of correlation.

3 Characterizing Gap Between End-to-End and Two-Stage

We begin by formally defining the *decision-focused learning* setting (cf. Donti, Amos, and Kolter 2017). Let $x \in \mathbb{R}^d$ be observable inputs (e.g., time of day, weather, etc.) and $y \in \mathbb{R}^d$ be outputs drawn from some unknown distribution $P(Y|X)$ (e.g., the distribution of travel times given those observables). For any observable input, x, we want to optimize decision variables, z, to minimize our downstream task loss $\mathcal{L}(z;x) = \mathbb{E}_{y\sim P(Y|x)}[f(y,z)]$ subject to a set of constraints, $\mathcal{G}$: $z^*(x) = \arg\min_{z\in\mathcal{G}} \mathcal{L}(z;x)$. In our example, if y represents likely travel times given the current road conditions, x, then $z^*(x)$ would be the optimal route recommendation, taking into account uncertainty over y.

Identifying z^* is a stochastic programming problem. Since such problems are intractable in general, it is common practice instead to optimize some proxy deterministic optimization problem $\tilde{\mathcal{L}}(z;x) = f(\phi(x), z)$ that replaces the expectation over $P(Y|x)$ with some deterministic prediction $\hat{y} = \phi(x)$. The hope is that the corresponding optimum $\tilde{z}^*(x) = \arg\min_{z\in\mathcal{G}} f(\phi(x), z)$ will be similar to the stochastic optimum $z^*(x)$. For example, in the predict-then-optimize setting, one must first precommit to learning some functional of the distribution—usually, $\phi(x) = E[y|x]$—and then proceed with the deterministic optimization on the basis of these predictions. We refer to this as the two-stage approach. It incurs a loss of,

$$\mathcal{L}(z^*_{\text{two-stage}}; x) = \mathbb{E}_{y\sim P(Y|x)}[f(y, z^*_{\text{two-stage}})] \tag{1}$$

on the true stochastic problem, where $z^*_{\text{two-stage}}(x) = \arg\min_{z\in\mathcal{G}} f(\phi(x), z)$.[1] Two-stage methods are more computationally efficient because they do not require resolving the optimization problem at each iteration of the learning algorithm, and if they do not result in suboptimal performance in the stochastic optimization, they should thus be preferred.

Of course, one might expect that proxy optima from task-agnostic choices for $\phi(x)$ of this form are unlikely to correspond to the stochastic optimum; indeed, a key contribution of this paper is formally investigating this intuition. In cases where indeed $\mathcal{L}(z^*, x) < \mathcal{L}(z^*_{\text{two-stage}}, x)$, a better approach can be selecting $\phi(x)$ with the task in mind. The end-to-end approach learns a parameterized model $\phi(x;\theta)$ such that the solutions to the deterministic proxy optimization function $\tilde{\mathcal{L}}(z;x)$ minimize the loss of the associated stochastic optimization problem.

Given its parameters θ and the observed input x, the end-to-end approach (1) outputs some prediction $\phi(x;\theta)$; (2) given that prediction, determines the optimal decisions $z_{\text{end-to-end}}(x,\theta)$; and (3) evaluates $z_{\text{end-to-end}}(x,\theta)$ on the true distribution, updating θ to minimize the resulting loss (typically by back-propagating through the optimization routine). The resulting solution takes the form

$$\mathcal{L}(z_{\text{end-to-end}}(x;\theta); x) = \mathbb{E}_{y\sim P(Y|x)}[f(y, z_{\text{end-to-end}})] \tag{2}$$

where

$$z_{\text{end-to-end}}(x;\theta) = \arg\min_{z\in\mathcal{G}} f(\phi(x;\theta), z) \tag{3}$$

$$\theta^* = \arg\min_{\theta} \mathcal{L}(z^*_{\text{end-to-end}}(x,\theta); x). \tag{4}$$

[1] Note that there can be multiple optimal solutions $z^*_{\text{two-stage}}$ and $z'^*_{\text{two-stage}}$ that incur different loss $\mathcal{L}(z^*_{\text{two-stage}}; x) \neq \mathcal{L}(z'^*_{\text{two-stage}}; x)$. We assume the two-stage approach breaks ties arbitrarily since it has no knowledge of the downstream task. All results are unaffected by this assumption.

End-to-end approaches represent an attractive middle ground between solving the full stochastic problem and achieving computational tractability via strong independence assumptions.

In what follows, we focus on bounding the performance gap between the two-stage and end-to-end approaches $\mathcal{L}(z^*_{\text{two-stage}};x)/\mathcal{L}(z^*_{\text{end-to-end}};x)$. For all of our results, we assume an "error-free" setting, where both the end-to-end and two-stage approaches have access to the Bayes optimal prediction for their respective loss functions, such that our results are driven by the choice of target, not by estimation error. The Bayes optimal predictor, $f^* = \arg\min_f E[\mathcal{L}(y, f(x))]$ minimizes generalization error for a given loss, $\mathcal{L}$, with features x and targets y. The two-stage approach models the conditional expectation, $f^*(x) = E[Y|x]$, which is Bayes optimal when the loss is mean squared error, $\mathcal{L}(y, f(x)) = (y - f(x))^2$.

We begin by making an important connection between the performance gap we have just defined and the *price of correlation* introduced by Agrawal et al. (2012). In words, the price of correlation is the worst-case loss in performance incurred by ignoring correlation in the random variables of a stochastic optimization problem.

Definition 1 (Price of Correlation)**.** *The price of correlation for a loss function $\mathcal{L}$ is POC $= \mathcal{L}(z')/\mathcal{L}(z^*)$, where z^* is the optimal solution to the stochastic program and z' is the solution minimizing a proxy stochastic program that makes the simplifying assumption that the random variables $\{y_i : i \in (1,\dots,d)\}$ are mutually independent.*

Making connections to the price of correlation will help us to establish performance gaps between two-stage and end-to-end. We first show that $\mathcal{L}(z^*_{\text{two-stage}};x)$ is lower bounded by the POC numerator, which implies the following result.

Lemma 3.1. *For any stochastic optimization problem over Boolean random variables $\{y_i : i \in (1,\dots,d)\}$, POC $= \mathcal{L}(z^*_{\textit{two-stage}};x)/\mathcal{L}(z^*;x)$.*

All proofs appear in the appendix[2]; roughly, this result follows from the fact that the two-stage approach produces an equivalent proxy stochastic optimization program to the program where all random variables are assumed to be mutually independent.

If we knew that $\mathcal{L}(z^*;x)$ was always equal to $\mathcal{L}(z^*_{\text{end-to-end}};x)$, we could leverage existing POC results to derive bounds on worst-case performance gaps between two-stage and end-to-end. Unfortunately, even the end-to-end approach can be suboptimal.

Proposition 3.2. *There exist stochastic optimization problems for which $\mathcal{L}(z^*;x) < \mathcal{L}(z^*_{\textit{end-to-end}};x)$.*

Nevertheless, for many stochastic optimization problems with POC results, the end-to-end approach does provably obtain an optimal solution.

Theorem 3.3. *End-to-end is optimal with respect to the stochastic objective for the problems described in Examples 1,2, and 3 from Agrawal et al. (2012): (i) Two-stage minimum cost flow, (ii) Two-stage stochastic set cover, (iii) Stochastic optimization minimization with monotone submodular cost function.*

[2]For the appendix, please see https://www.cs.ubc.ca/labs/beta/Projects/2Stage-E2E-Gap/

The key idea behind the proof for each part is proving that for any optimal solution z^*, end-to-end outputs a deterministic target prediction $y_{z^*} \in \mathbb{R}^d$ such that $\arg\min_z f(y_{z^*}, z) = z^*$.

Theorem 3.3 and Examples 1,2, and 3 from Agrawal et al. (2012) imply the following bounds on worst-case gaps for $\mathcal{L}(z^*_{\text{two-stage}};x)/\mathcal{L}(z^*_{\text{end-to-end}};x)$.

Corollary 3.3.1. *In the worst case, $\mathcal{L}(z^*_{\textit{two-stage}};x)/\mathcal{L}(z^*_{\textit{end-to-end}};x)$ is:*

(i) $\Omega\left(2^d\right)$ *for two-stage minimum cost flow;*

(ii) $\Omega\left(\sqrt{d}\frac{\log(\log(d))}{\log(d)}\right)$ *for two-stage stochastic set cover; and*

(iii) $\Omega\left(\frac{e}{1-e}\right)$ *for stochastic optimization minimization with monotone submodular cost function problems.*

While this corollary shows that we can leverage known results from the POC literature to identify large performance gaps between the end-to-end and two-stage approaches (e.g., setting *(i)*), the POC literature focuses mostly on finding settings such as *(ii)* or *(iii)* where the cost of ignoring correlations grows slowly as a function of the problem size. POC results also tend to arise in the context of stochastic optimization where learning a prediction model of inputs is not common practice and therefore end-to-end learning is not always applicable (Agrawal et al. 2012; Lu, Ran, and Shen 2015).

We now turn our attention to a setting not previously studied in the POC literature, in which we will show that correlations can lead to unbounded worst-case performance gaps between the two-stage and end-to-end approaches. This setting is particularly amenable to the "predict-then-optimize" approach. Specifically, we allow for two target vectors $y_1 \in \mathbb{R}^d$ and $y_2 \in \mathbb{R}^d$, aiming to capture scenarios where coefficients of the objective function depends on more than one unknown quantity (e.g., demand and travel time) and our objective function takes the form

$$f(y, z) = \sum_{i=1}^{d} \gamma(y_{1_i}, y_{2_i}) f_i(z) + C, \tag{5}$$

where $f_i(z)$ are arbitrary functions of the decision variables z (e.g., monomials of degree one or two for linear (LP) and quadratic programs (QP) respectively) and $\gamma(y_{1_i}, y_{2_i}) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ represent coefficients as a function of prediction targets.

When $\gamma(y_{i,1}, y_{i,2}) = y_{i,1} \cdot y_{i,2}$ (e.g., demand-weighted travel time if y_1 and y_2 represent demand and travel times, respectively), it is possible to construct settings where the gap between the end-to-end and two-stage approaches grows without bound in the dimensionality of the problem, d.

Theorem 3.4. *For $\gamma(a, b) = a \cdot b$, there exists an optimization problem with objective function of the form given by Equation 5 and distribution $P(X, Y)$ such that $\mathcal{L}(z^*_{\textit{two-stage}};x)/\mathcal{L}(z^*_{\textit{end-to-end}};x) \to \infty$ as $d \to \infty$.*

The key idea is to construct a distribution where two-stage wrongly estimates a coefficient to be nonzero when the true expected value is zero. We correlate some pairs a, b where γ is applied so that at least one is zero with probability 1, but the expectation of both a and b is above zero. We set up the optimization problem so that the two-stage approach systematically avoids selecting decision variables with zero cost, causing it to pay an unnecessary cost for every dimension of the problem. Note that Theorem 3.4 is an existence result and therefore must hold for more general γ (e.g., a function of any number of the prediction targets).

Furthermore, the end-to-end solution is optimal with respect to the stochastic objective.

Lemma 3.5. $\mathcal{L}(z^*; x) = \mathcal{L}(z^*_{\textit{end-to-end}}; x)$ *if the objective function is of the form given by Equation* (5)*.*

We now construct a gap between the two-stage and the end-to-end approaches for any nonlinear γ. First, we show an obvious implication of the linearity of expectation.

Lemma 3.6. *For all* $y, y' \in \mathbb{R}$, $\gamma(y, y')$ *is linear iff* $\forall P(Y), \mathbb{E}_{P(Y)}[\gamma(y, y')] = \gamma(\mathbb{E}_{P(Y)}[y], \mathbb{E}_{P(Y)}[y'])$.

Intuitively, only when the whole objective function f is linear are we guaranteed that f's expected value is equivalent to f applied to the expected values of each of its inputs. A two-stage approach moves the expectation inside the optimization parameterization since it only models the expected values of the target vector y. We show how wrongly estimating the expectation of the parameterization leads to suboptimal decisions.

Theorem 3.7. *For any nonlinear* γ*, we can construct a distribution* $P(Y)$*, functions* $f_i(z)$*, and constraints* $g(z)$ *such that* $\mathcal{L}(z^*; x) < \mathcal{L}(z^*_{\textit{two-stage}}; x)$ *for* $d \geq 2$.

Although we show that the two-stage predict-then-optimize approach can achieve poor performance in general, there do exist special cases in which it is optimal.

Theorem 3.8. $\mathcal{L}(z^*; x) = \mathcal{L}(z^*_{\textit{two-stage}}; x)$ *if the objective function is of the form given by Equation* (5) *and either (i)* γ *is a linear function of its inputs; or (ii)* $(Y_1 \perp Y_2)|X$ *and* $\gamma(y_{1_i}, y_{2,i}) = y_{1_i} \cdot y_{2_i}$ *for all i.*

This result follows directly from linearity of expectation and gives some explanation for the wide use of the two-stage approach in practice (In the appendix, we show more generally that two-stage is optimal for any optimization problem $f(y, z)$ that is linear in y.). The first condition of Theorem 3.8 is implicitly known in the literature but not stated in this form with multiple prediction targets (Elmachtoub and Grigas 2020). In Section 5, we argue that these conditions for two-stage optimality are unlikely to hold in various cases where two-stage approaches are nevertheless used in practice.

4 Simulation Study

We have seen that even in what one would imagine would be benign settings—such as a linear program where the coefficients of the decision variables are the product of two predictions, $y_1 y_2$—correlations between target variables can lead to potentially unbounded gaps between the performance of two-stage and end-to-end approaches. In this section, we contextualize these results by (1) providing intuition for how such correlations might arise in practice and (2) experimentally exploring how correlation impacts the performance gap through a simulation.

We will consider the example of a delivery company that faces a choice of where to locate its facilities, with the aim of minimizing average demand-weighted travel time between these facilities and its customers. This can be formulated as an integer program (and potentially relaxed to an LP) of the form $\arg\min_z \sum_{i,j} \mathbb{E}_{\mathcal{D}}[T_{ij} d_j] z_{ij}$ with constraints that ensure (1) at most k locations are chosen for placing facilities and (2) a customer i can only connect to a location j if j is chosen to be a facility. Every assignment $z_{i,j}$ between customer i and facility j is weighted by the expected demand-weighted travel time between i and j. This parameterization is broken down into the element-wise product of two conceptually different learning targets: (1) $T_{i,j}$, the travel times between every pair of endpoints (i.e., time cost between customer and facility); and (2) d_j, the demand of every customer j.

The magnitude of the performance gap between end-to-end optimization and two-stage optimization will depend on the degree of correlation between these targets T and d. If they were independent, then Theorem 3.8 would apply and it would be safe to use a two-stage approach, learning separate models to predict T and d. However, there are many reasons why these targets might be correlated. For example, the weather may have some effect on demand (e.g., when it is raining, customers' demand for deliveries increases) as well as travel times (e.g., in the rain, congestion tends to increase, both because people drive more cautiously and because accidents nevertheless become more common). If the models for demand and travel times do not condition on the weather, then a two-stage approach will wrongly estimate expected-demand-weighted travel times, potentially leading to suboptimal decisions. One solution is for practitioners to find the right conditioning set to ensure the targets are conditionally independent. However, sources of potential correlation are both abundant and challenging to discover.[3]

We wanted to experimentally investigate the dependence between correlation and the performance gap given plausible correlation structure, and so designed a synthetic benchmark for this demand-weighted facility location problem. Our input is a bipartite graph between n customers and m facilities and the output is an assignment of customers to facilities such that only k facilities are connected. Travel times $T_{i,j}$ and demands d_j are sampled from some ground truth distribution $\mathcal{D}$, which we parameterize with ρ that controls correlation structure. We want to learn to output $\hat{T_{ij}}$ and $\hat{d_j}$ from samples of the true distribution $\mathcal{D}$ in order to choose assignments $\hat{z}_{i,j}$

[3]There is a close connection to finding the appropriate adjustment set in the causality literature. If one can specify the causal graph that generates the data, it is possible to find minimal adjustment sets (see Pearl 2009, for details).

that minimize $\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}_{\mathcal{D}}[T_{ij}d_j]\hat{z_{ij}}$ subject to:

$\sum_{i=1}^{n} z_{ij} = 1$ for all $j = 1, \ldots, m$

$\sum_{j=1}^{m} z_{ij} \leq Mx_i$ for all $i = 1, \ldots, n$, $\quad \sum_{i=1}^{n} x_i \leq k$

$z_{ij} \in \{0, 1\}$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$

$x_{0,1} \in \{0, 1\}$ for all $i = 1, \ldots, n$.

4.1 End-to-End Approach

For the end-to-end demand-weighted facility location solution, we used a one-layer feed-forward neural network which takes as input a matrix of features for every edge i,j between customer i and facility j and outputs a demand-weighted travel-time matrix. During training we used a differentiable optimization layer for solving the linear-programming relaxation of the facility location problem based on the work of Amos and Kolter (2017) and Wilder, Dilkina, and Tambe (2019). Amos and Kolter showed how to find the gradient of solutions for a class of convex programs with respect to the input parameters (i.e., demand and travel times) by differentiating through the Karush-Kuhn-Tucker (KKT) conditions, which are linear equations expressing the gradients of the objective and constraints around the optimum. The optimal solution to a linear program may not be differentiable with respect to its parameters, so Wilder, Dilkina, and Tambe add a quadratic smoothing term $||z||_2^2$ weighted by ζ to create a strongly concave quadratic program. This allows for the optimal solution to move continuously with changes in parameters. The objective function becomes $\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}_{\mathcal{D}}[T_{ij}d_j]z_{ij} + \zeta||z||_2^2$.

We implemented this smoothed differentiable version of a linear program with the encoding described earlier using the `cvxpylayers` package (Agrawal et al. 2020). To find an integer solution at test time, we solved this program optimally with a mixed-integer solver without the smoothing term. Since the feed-forward network and the quadratic program are all differentiable with respect to their inputs, we could train the system end-to-end with backpropagation. Following Wilder, Dilkina, and Tambe (2019), we defined the loss of the network to be the solution quality of the customer-facility assignments output from the linear program $\hat{z}$ given the ground truth demand-weighted distances c. By the objective function, the loss is then $c^T\hat{z}$.

We used the ADAM optimizer with a learning rate of 0.01 and performed 500 training iterations for each experiment. We set our quadratic penalty term ζ to be 10. We trained on an 8-core machine with Intel i7 3.60GHz processors and 32 GB of memory and an Nvidia Titian Xp GPU. To get integer solutions at test time, we used the mixed-integer solver GLPK in the `cvxpy` python package. Training times were very small. Each end-to-end model took a few minutes to train with each MIP requiring less than a second to find the optimal solution.

4.2 Experimental Setup

We now construct a simple graph distribution to simulate our working example. The principle behind the distribution is that there are some routes between customers and facilities whose travel times correlate with customer demand (e.g., inclement weather increases both demand and travel times within a city centre) and some that are independent (e.g., weather does not affect travel time on freeway routes). We model some latent variable l (e.g., weather) that correlates demand and travel times (e.g., demand and travel times will both be higher with inclement weather). Every customer has a correlation between demand and travel time controlled by some parameter ρ for half of the facilities F_1 and zero correlation for the other half of facilities F_2. The parameter ρ controls the correlations that the latent factor l induces between $T_{i,j}$ and D_i for the facilities in F_1. When ρ is 0, there is no correlation between the two; with $\rho = 1$, they are perfectly correlated; and when $\rho = -1$, they are perfectly anticorrelated. Our distribution $\mathcal{D}$ is as follows,

$$\mathcal{D} = \begin{cases} l_i \sim \mathcal{N}(0, \rho^2\sigma_{li}^2) \\ T_{i,j} \sim \mathcal{N}(\mu_{i,j} + |l_i|, (1-\rho^2)\sigma_t^2) \quad \forall j \in F_1 \\ T_{i,j} \sim \mathcal{N}(\mu_{i,j}, \sigma_t^2) \quad \forall j \in F_2 \\ D_i \sim \mathcal{N}(l_i + \mu_i, \sigma_d). \end{cases}$$

In our experiments, we set $m, n = 20, \sigma_{li} = 20.25$, $\sigma_d = \sigma_t = 3$, $\forall j \in F_1, \mu_{i,j} = 5.5$, $\forall j \in F_2, \mu_{i,j} = 6, K = 1$. We evaluated three approaches; our end-to-end approach described above, a two-stage approach, and an OPT approach. The two-stage approach is Bayes optimal for mean-squared error. It selects assignments based on sample means of training samples for every customer demand and customer-facility travel time. Knowing the expected value for every (demand, travel time) product is sufficient for optimal performance, therefore we defined OPT as selecting assignments based on sample means from the training set of every (demand, travel time) product. We found an integer-optimal solution for every approach at test time. We ran an experiment for 11 values of ρ between -1 and 1 at equal intervals, which we evaluated on 10 random seeds. For each evaluation, we generated 1000 samples from our distribution and left out 200 as test data. Please see https://www.cs.ubc.ca/labs/beta/Projects/2Stage-E2E-Gap/ for our code and data generation process.

4.3 Results

Our results are shown in Figure 1. The end-to-end approach performed equally as good as OPT across all values of ρ. The performance gap grew substantially with increasing positive correlation towards $\rho = 1$. With positive correlation, the two-stage approach substantially underestimated the demand-weighted travel times to F_1 facilities and tended to incorrectly choose to place the facility to be in F_1. As we expect by Theorem 3.8, there was no gap when $\rho = 0$ as demands and travel times are independent of one another. With increasing negative correlation towards $\rho = -1$, we did not see much change in the gap. Negative correlation caused the two-stage approach to substantially overestimate the demand-weighted travel times but the two-stage approach tended to pick the correct facility. This illustrates an important point for understanding when gaps occur in practice. Incorrect estimations are not enough for bad decisions. The estimation must be wrong in a direction that causes a change in decision. The large confidence intervals of the two-stage approach towards the extremes at $\rho = -1$ and $\rho = 1$ suggest that two-stage is

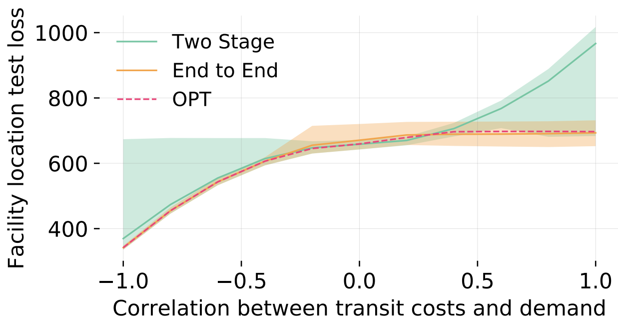

Figure 1: Facility location test performance vs correlation structure with samples from $\mathcal{D}$. End-to-end refers to the method described in Section 4.1; two-stage approach learns sample means for every demand and travel time; and OPT learns sample means for every (demand, travel time) product. The shaded areas represent 90th percentiles.

also more sensitive to noise in the distribution. An end-to-end approach will recognize that the value of F_1 facilities relative to F_2 facilities will grow as $\rho \to -1$, and as a result, it is more robust to estimation error. With infinite training samples, the two-stage approach will make the correct decision for $\rho \leq 0$, but as $\rho \to -1$, the margin of error stays constant while wrong decisions become more consequential.

5 Two-Stage Optimization with Potentially Correlated Targets in Real Applications

While we have seen that large performance gaps can arise in a synthetic setting, we would of course like to understand whether a similar phenomenon arises in real applications. Unfortunately, we are not practitioners ourselves, and we were not able to identify any publicly available dataset giving the raw inputs to a two-stage optimization approach. We observed that the relevant literature tends to concentrate on the optimization part of the two-stage optimization problem rather than publishing raw data underlying the prediction phase.

In the absence of such a dataset, we sought other evidence that two-stage optimization was being deployed in practical settings where our theoretical results suggest that end-to-end optimization could be expected to achieve better performance. We identified a wide range of applications from the literature in which a two-stage approach was used and where there exists potential for correlation between multiplied targets. The examples we found came from a range of different optimization problems, but all involved products of the demand and cost predictions on different travel routes (for buses, taxis, supply chains, and product delivery), and all failed to account for latent factors that might induce correlation in these targets, such as weather or seasonality. Of course, without access to the original data, we cannot be certain that accounting for these latent effects would have resulted in significant changes to solution quality, but they serve as useful examples to show how such correlations can potentially occur. Furthermore, we were struck by how many such examples we were able to discover. Overall, we hope that our work will be taken as a call to arms for the community to explore end-to-end learning on these applications, and that it will encourage practitioners to publish raw data underlying both prediction and optimization phases.

We now survey the specific work we identified. Line-planning problems involve taking the products of travel time and demands for bus routes (Mauttone and Urquhart 2009). Demand and travel time can be correlated through precipitation as increasing precipitation has been found to decrease bus ridership (Zhou et al. 2017) and increase travel time (Tsapakis, Cheng, and Bolbol 2013). In ride-sharing programs, the weights of the matching problem that assigns drivers to customers involves taking the product of the probabilities that a future request is real and the times to serve that request (Alonso-Mora, Wallar, and Rus 2017). Travel times are similarly affected by weather and taxi demand tends to decrease with rainfall during non-rush hours but increase with rainfall during rush hours (Chen et al. 2017). We could not find any papers on either ride sharing or line planning whose authors explicitly conditioned on weather to render these targets conditionally independent, so our analysis indicates that end-to-end approaches likely offer scope for improved solutions.

In supply chain planning, companies optimize routes across supplier stages to customers. For example, Zhen, Zhuge, and Lei (2016) investigate supply chain planning for automobile manufacturers. Route costs are proportional to the product of both customer demands and transportation costs between stages, which can be correlated through seasonality. Tayyeba Irum (2020) show that demand for automobiles peaks in the spring and fall and van der Tuin and Pel (2020) show it is more difficult to procure freight during peak season and transportation delays (and higher costs) are more common.

Finally, demand-weighted facility location problems place facilities to respond to customers and minimize transportation costs; as such, the cost of each route involves the product of a prediction for customer demands and the transportation costs from facilities to customers (Sert et al. 2020). Transportation delays are higher during peak season (van der Tuin and Pel 2020) and most product demands will have seasonal effects. Sert et al. use anonymized data so it is difficult to speculate on specific sources of correlation, but they do not explicitly discuss controlling for these effects.

6 Conclusions

This paper presented lower bounds on the worst-case difference between two-stage and end-to-end approaches. First, we derive bounds by drawing a connection to *price-of-correlation* results in stochastic optimization. Then, we prove results for a practical setting that apply to benign sources of non-linearity: taking the product of two predictions in an otherwise linear optimization problem is sufficient to construct gaps that grow with the dimensionality of the problem. We identify a number of applications with coefficients of this form where end-to-end solutions may offer large improvements in practice.

Acknowledgements

This work was funded by an NSERC Discovery Grant, a DND/NSERC Discovery Grant Supplement, a CIFAR Canada AI Research Chair (Alberta Machine Intelligence Institute), a Compute Canada RAC Allocation, a GPU grant from Nvidia, awards from Facebook Research and Amazon Research, and DARPA award FA8750-19-2-0222, CFDA #12.910 (Air Force Research Laboratory).

References

Agrawal, A.; Amos, B.; Barratt, S.; Boyd, S.; Diamond, S.; and Kolter, J. Z. 2020. Differentiable Convex Optimization Layers. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20, 9558–9570.

Agrawal, A.; Barratt, S.; Boyd, S.; Busseti, E.; and Moursi, W. M. 2019. Differentiating through a cone program. *Journal of Applied and Numerical Optimization*, 1: 107–115.

Agrawal, S.; Ding, Y.; Saberi, A.; and Ye, Y. 2012. Price of correlations in stochastic optimization. *Operations Research*, 60(1): 150–162.

Alonso-Mora, J.; Wallar, A.; and Rus, D. 2017. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3583–3590. IEEE.

Amos, B.; and Kolter, J. Z. 2017. OptNet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 136–145. PMLR.

Bahulkar, A.; Szymanski, B. K.; Baycik, N. O.; and Sharkey, T. C. 2018. Community detection with edge augmentation in criminal networks. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 1168–1175. IEEE.

Barratt, S. 2018. On the differentiability of the solution to convex optimization problems. *arXiv preprint*, 1804.05098.

Bertsimas, D.; Natarajan, K.; and Teo, C. 2004. Probabilistic Combinatorial Optimization: Moments, Semidefinite Programming, and Asymptotic Bounds. *SIAM J. Optim.*, 15(1): 185–209.

Centola, D. 2018. *How behavior spreads: The science of complex contagions*, volume 3. Princeton University Press Princeton, NJ.

Chen, D.; Zhang, Y.; Gao, L.; Geng, N.; and Li, X. 2017. The impact of rainfall on the temporal and spatial distribution of taxi passengers. *PloS one*, 12(9): e0183574.

Demirović, E.; Stuckey, P. J.; Bailey, J.; Chan, J.; Leckie, C.; Ramamohanarao, K.; and Guns, T. 2019. An investigation into prediction+ optimisation for the knapsack problem. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 241–257. Springer.

Djolonga, J.; and Krause, A. 2017. Differentiable Learning of Submodular Models. In *Advances in Neural Information Processing Systems 31*, NIPS '17, 1013–1023.

Donti, P. L.; Amos, B.; and Kolter, J. Z. 2017. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems 31*, NIPS '17, 5484–5494.

Elmachtoub, A. N.; and Grigas, P. 2020. Smart "Predict, then Optimize". arXiv:1710.08005.

Ferber, A.; Wilder, B.; Dilkina, B.; and Tambe, M. 2020. Mipaal: Mixed integer program as a layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1504–1511.

Lu, M.; Ran, L.; and Shen, Z.-J. M. 2015. Reliable facility location design under uncertain correlated disruptions. *Manufacturing & Service Operations Management*, 17(4): 445–455.

Mauttone, A.; and Urquhart, M. E. 2009. A route set construction algorithm for the transit network design problem. *Computers & Operations Research*, 36(8): 2440–2449.

Pearl, J. 2009. *Causality*. Cambridge University Press.

Sert, E.; Hedayatifar, L.; Rigg, R. A.; Akhavan, A.; Buchel, O.; Saadi, D. E.; Kar, A. A.; Morales, A. J.; and Bar-Yam, Y. 2020. Freight Time and Cost Optimization in Complex Logistics Networks. *Complexity*, 2020.

Tayyeba Irum, C. H. 2020. Sales drop in major auto markets during January. *S&P Global Market Intelligence*.

Tsapakis, I.; Cheng, T.; and Bolbol, A. 2013. Impact of weather conditions on macroscopic urban travel times. *Journal of Transport Geography*, 28: 204–211.

Tschiatschek, S.; Sahin, A.; and Krause, A. 2018. Differentiable submodular maximization. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI '18, 2731–2738. Morgan Kaufmann.

van der Tuin, M. S.; and Pel, A. J. 2020. The disruption transport model: computing user delays resulting from infrastructure failures for multi-modal passenger & freight traffic. *European Transport Research Review*, 12(1): 1–10.

Wilder, B.; Dilkina, B.; and Tambe, M. 2019. Melding the Data-Decisions Pipeline: Decision-Focused Learning for Combinatorial Optimization. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, AAAI '19, 1658–1665. AAAI Press.

Wilder, B.; Ewing, E.; Dilkina, B.; and Tambe, M. 2020. End to end learning and optimization on graphs. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20, 539–548.

Yan, B.; and Gregory, S. 2012. Detecting community structure in networks using edge prediction methods. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(09): P09008.

Zhen, L.; Zhuge, D.; and Lei, J. 2016. Supply chain optimization in context of production flow network. *Journal of Systems Science and Systems Engineering*, 25(3): 351–369.

Zhou, M.; Wang, D.; Li, Q.; Yue, Y.; Tu, W.; and Cao, R. 2017. Impacts of weather on public transport ridership: Results from mining data from different sources. *Transportation research part C: emerging technologies*, 75: 17–29.