Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Data Pooling in Stochastic Optimization

Vishal Gupta, Nathan Kallus

http://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Articles in Advance, pp. 1–21
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Data Pooling in Stochastic Optimization

Vishal Gupta,[a] Nathan Kallus[b]

[a] Data Science and Operations, USC Marshall School of Business, Los Angles, California 90089; [b] School of Operations Research and Information Engineering and Cornell Tech, Cornell University, New York, New York 10044

Contact: guptavis@usc.edu, https://orcid.org/0000-0003-4371-9114 (VG); kallus@cornell.edu, https://orcid.org/0000-0003-1672-0507 (NK)

Received: May 31, 2019
Revised: March 29, 2020; September 7, 2020; November 19, 2020
Accepted: November 23, 2020
Published Online in Articles in Advance: March 17, 2021

https://doi.org/10.1287/mnsc.2020.3933

Abstract. Managing large-scale systems often involves simultaneously solving thousands of unrelated stochastic optimization problems, each with limited data. Intuition suggests that one can decouple these unrelated problems and solve them separately without loss of generality. We propose a novel data-pooling algorithm called *Shrunken-SAA* that disproves this intuition. In particular, we prove that combining data across problems can outperform decoupling, even when there is no a priori structure linking the problems and data are drawn independently. Our approach does not require strong distributional assumptions and applies to constrained, possibly nonconvex, nonsmooth optimization problems such as vehicle-routing, economic lot-sizing, or facility location. We compare and contrast our results to a similar phenomenon in statistics (Stein's phenomenon), highlighting unique features that arise in the optimization setting that are not present in estimation. We further prove that, as the number of problems grows large, Shrunken-SAA learns *if* pooling can improve upon decoupling *and* the optimal amount to pool, even if the average amount of data per problem is fixed and bounded. Importantly, we highlight a simple intuition based on stability that highlights *when* and *why* data pooling offers a benefit, elucidating this perhaps surprising phenomenon. This intuition further suggests that data pooling offers the most benefits when there are many problems, each of which has a small amount of relevant data. Finally, we demonstrate the practical benefits of data pooling using real data from a chain of retail drug stores in the context of inventory management.

History: Accepted by Chung Piaw Teo, Special Issue on Data-Driven Prescriptive Analytics.
Funding: This work was supported by the National Science Foundation [Grant 1661732 to V.Gupta and Grant 1656996 to N. Kallus].
Supplemental Material: The online appendices are available at https://doi.org/10.1287/mnsc.2020.3933.

Keywords: data-driven optimization • small-data, large-scale regime • shrinkage • James-Stein estimation

1. Introduction

The stochastic optimization problem

$$\min_{x \in \mathcal{X}} \quad \mathbb{E}^{\mathbb{P}}[c(x, \xi)] \tag{1.1}$$

is a fundamental model, with applications ranging from inventory management to personalized medicine. In typical data-driven settings, the measure $\mathbb{P}$ governing the random variable ξ is unknown. Instead, we have access to a data set $\mathcal{S} = \{\hat{\xi}_1, \ldots, \hat{\xi}_N\}$ independent and identically distributed (i.i.d.) from $\mathbb{P}$ and seek a decision $x \in \mathcal{X}$ depending on these data. This model and its data-driven variant have been extensively studied in the literature (see Shapiro et al. 2009 for an overview).

Managing real-world, large-scale systems, however, frequently involves solving thousands of potentially unrelated stochastic optimization problems like Problem (1.1) simultaneously. For example, inventory management often requires optimizing stocking levels for many distinct products across categories, not just a single product. Firms typically determine staffing and capacity for many warehouses and fulfillment centers across the supply chain, not just at a single location. Logistics companies often divide large territories into many small regions and solve separate vehicle-routing problems, one for each region, rather than solve a single monolithic problem. In such applications, a more natural model than Problem (1.1) might be

$$\frac{1}{K}\sum_{k=1}^{K} \frac{\lambda_k}{\lambda_{\text{avg}}} \min_{x_k \in \mathcal{X}_k} \mathbb{E}^{\mathbb{P}_k}[c_k(x_k, \xi_k)], \tag{1.2}$$

where we solve a separate subproblem of the form (1.1) for each k, for example, setting a stocking level for each product. Here, $\lambda_k > 0$ represents the frequency with which the decision maker incurs costs from problems of type k, and $\lambda_{\text{avg}} = \frac{1}{K}\sum_{k=1}^{K} \lambda_k$. Thus, in Problem (1.2), our total costs are driven by the frequency-weighted average of the costs of many distinct optimization problems.

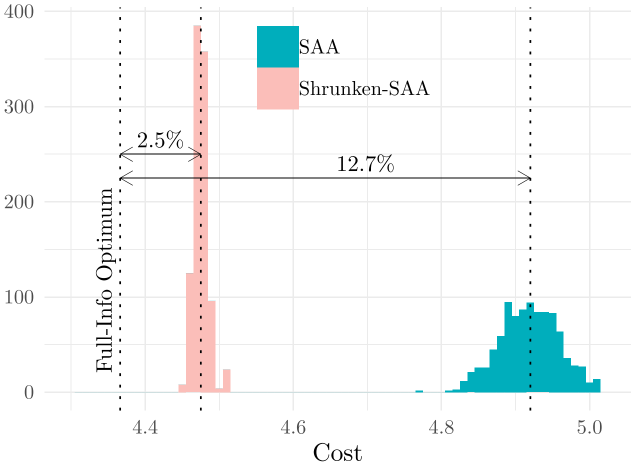
Of course, intuition strongly suggests that since there are no coupling constraints across the feasible regions $\mathcal{X}_k$ in Problem (1.2), one can and should decouple the problem into K unrelated subproblems and solve them separately. Indeed, when the measures $\mathbb{P}_k$ are known, this procedure is optimal. When the $\mathbb{P}_k$'s are unknown and unrelated, but one has access to a data set $\mathcal{S}_k = \{\hat{\xi}_{k,1}, \dots, \hat{\xi}_{k,\hat{N}_k}\}$ drawn i.i.d. from $\mathbb{P}_k$ independently across k, intuition *still* suggests that decoupling is without loss of generality and that data-driven procedures can be applied separately by subproblem.

A key message of this paper is that this intuition is *false*.

In the data-driven setting, when solving many stochastic optimization problems, we show that there exist algorithms that pool data across subproblems that outperform decoupling, *even* when the underlying problems are unrelated and the data are *independent*. This phenomenon holds, despite the fact that the kth data set $\mathcal{S}_k$ tells us nothing about $\mathbb{P}_l$ for $l \neq k$ and there is no a priori relationship between the $\mathbb{P}_k$. We term this phenomenon the *data-pooling phenomenon in stochastic optimization*.

Figure 1 illustrates the data-pooling phenomenon with a simulated example for emphasis. Here, $K = 10,000$, and the kth subproblem is a newsvendor problem with critical quantile 90%, that is, $c_k(x;\xi) = \max\{9(\xi - x), (x - \xi)\}$. The measures $\mathbb{P}_k$ are fixed, and in each run we simulate $\hat{N}_k = 20$ data points per subproblem. For the decoupled benchmark, we use a standard method, Sample Average Approximation (SAA; Definition 2.1), which is particularly well-suited to the data-driven newsvendor problem (Levi et al. 2015). For comparison, we use our novel Shrunken-SAA algorithm, which exploits the data-pooling phenomenon. We motivate and formally define Shrunken-SAA in Section 3, but, loosely speaking, Shrunken-SAA proceeds by replacing the kth data set $\mathcal{S}_k$ with a "pooled" data set that is a weighted average of the original kth data set and all of the remaining $l \neq k$ data sets. It then applies SAA to each of these new pooled data sets. Perhaps surprisingly, by pooling data across the unrelated subproblems, Shrunken-SAA reduces the loss to full-information optimum by over 80% compared with SAA in this example.

Figure 1. (Color online) The Data Pooling Phenomenon



Notes. Consider $K = 10,000$ data-driven newsvendor problems, each with critical fractile 90% and 20 data points drawn independently across problems. SAA decouples the problems and orders the 90th-sample quantile in each. Shrunken-SAA (see Algorithm 1 in Section 3), leverages data pooling. Indicated percentages are losses to the full-information optimum. Additional details in Section E.1 in Online Appendix E.

Our Contributions

We describe and study the data-pooling phenomenon in stochastic optimization in the context of Problem (1.2). Our analysis applies to constrained, potentially nonconvex, nonsmooth optimization problems under fairly mild assumptions on the data-generating process. Specifically, we assume that each $\mathbb{P}_k$ has finite support (potentially differing across k); in some cases, we can even relax this assumption. We contrast the data-pooling phenomenon to a similar phenomenon in statistics (Stein's phenomenon), highlighting unique features that arise in the optimization setting (see Theorem 2.2 and Example 2.3). Namely, unlike traditional statistical settings, the potential benefits of data pooling depend strongly on the structure of the underlying optimization problems, and, in some cases, data pooling may offer no benefit over decoupling.

This observation raises important questions: Given a particular data-driven instance of Problem (1.2), should we data-pool, and, if so, how? More generally, does data pooling *typically* offer a significant benefit over decoupling, or are instances like Figure 1 somehow the exception to the rule?

To help resolve these questions, we propose a simple, novel algorithm that we call *Shrunken Sample Average Approximation* (*Shrunken-SAA*). Shrunken-SAA generalizes the classical SAA algorithm and, consequently, inherits many of its excellent large-sample asymptotic properties (see Remark 4.1). Moreover, Shrunken-SAA is incredibly versatile and can be tractably applied to a wide variety of optimization problems with computational requirements similar to traditional SAA (see Remark 3.1). Unlike traditional SAA, however, Shrunken-SAA exploits the data-pooling phenomenon to improve performance over SAA, as seen in Figure 1. Moreover, Shrunken-SAA exploits the structure of the optimization problems and strictly improves upon an estimate-then-optimize approach using traditional statistical shrinkage estimators (see Example 2.3 and Section 6).

Shrunken-SAA data-pools by combining data across subproblems in a particular fashion that is motivated by an empirical Bayesian argument. We prove that (under frequentist assumptions) for many classes of optimization problems, as the number of subproblems K grows large, Shrunken-SAA determines *if* pooling in this way can improve upon decoupling and, if so, also determines the optimal amount to pool (see Theorems 4.2, 4.3, 4.5, and 4.6). These theoretical results study Problem (1.2) when $\mathbb{P}_k$ has finite, discrete support and the amount of data available for the kth subproblem is, itself, random (see Assumption 3.1). Some of our results do extend to continuous distributions (see Section 4.6 and Theorems F.1–F.3 in Online Appendix F), and numerical experiments suggest that our results are generally robust to the assumption of a random amount of data.

More interestingly, our theoretical performance guarantees for Shrunken-SAA, hold even when the expected amount of data per subproblem is small and fixed and the number of problems K is large, as in Figure 1; that is, they hold in the so-called small-data, large-scale regime (Gupta and Rusmevichientong 2021). Indeed, since many traditional data-driven methods (including SAA) converge to the full-information optimum in the large-sample regime, the small-data, large-scale regime is arguably the more interesting regime in which to study the benefits of data pooling.

In light of the aforementioned results, Shrunken-SAA provides an algorithmic approach to deciding if, and by how much, to pool. To develop an intuitive understanding of *when* and *why* data pooling might improve upon decoupling, we also introduce the *Sub-Optimality-Instability Trade-Off*, a decomposition of the benefits of data pooling. We show that the performance of a data-driven solution to Problem (1.2) (usually called its *out-of-sample performance* in machine-learning settings) can be decomposed into a sum of two terms: a term that roughly depends on its in-sample suboptimality and a term that depends on its instability; that is, how much does in-sample performance change when training with one fewer data points? As we increase the amount of data pooling, we increase the in-sample suboptimality because we "pollute" the kth subproblem with data from other, unrelated subproblems. At the same time, however, we decrease the instability of the kth subproblem, because the solution no longer relies on its own data so strongly. Shrunken-SAA works by navigating this trade-off, seeking a "sweet spot" to improve performance. (See Section 5 for discussion.)

In many ways, the Sub-Optimality-Instability Trade-Off resembles the classical bias-variance trade-off from statistics. However, they differ in that the Sub-Optimality-Instability Trade-Off applies to general optimization problems, whereas the bias-variance trade-off applies specifically to the case of mean-squared error. Moreover, even in the special case when Problem (1.2) models mean-squared error, we prove that these two trade-offs are distinct (see Section D.2 in Online Appendix D). In this sense, the Sub-Optimality-Instability Trade-Off may be of independent interest outside data pooling.

Stepping back, this simple intuition suggests that Shrunken-SAA, and data pooling more generally, offer significant benefits whenever the decoupled solutions to the subproblems are sufficiently unstable, which typically happens when there is only a small amount of relevant data per subproblem. It is in this sense that the behavior in Figure 1 is typical and not pathological. Moreover, this intuition also naturally extends beyond Shrunken-SAA, paving the way to developing and analyzing new algorithms that also exploit the hitherto-underutilized data-pooling phenomenon.

Finally, we present numerical evidence in an inventory management context using real data from a chain of European drug stores showing that Shrunken-SAA can offer significant benefits over decoupling when the amount of data per subproblem is small to moderate. These experiments also suggest that Shrunken-SAA's ability to identify an optimal amount of pooling and improve upon decoupling are relatively robust to violations of our assumptions on the data-generating process.

Connections to Prior Work

Our proposal, Shrunken-SAA, generalizes SAA. In many ways, SAA is *the* most fundamental data-driven approach to Problem (1.1). SAA proxies $\mathbb{P}$ in (1.1) by the empirical distribution $\hat{\mathbb{P}}$ on the data and optimizes against $\hat{\mathbb{P}}$. It enjoys strong theoretical and practical performance in the large-sample limit, that is, when N is large (Kleywegt et al. 2002, Shapiro et al. 2009). For data-driven newsvendor problems, specifically—an example of which we use throughout our work—SAA is the maximum likelihood estimate of the optimal solution and also is the distributionally robust optimal solution for a Wasserstein ambiguity set (Esfahani and Kuhn 2018, p. 151). SAA is incredibly versatile and applicable to a wide variety of classes of optimization problems. This combination of strong performance and versatility has fueled SAA's use in practice. Applied to Problem (1.2), SAA decouples the optimization into its K subproblems. Thus, because of its strong theoretical and practical performance, we use SAA as the natural, "apples-to-apples" decoupled benchmark to which we compare our data-pooling procedures.

The data-pooling phenomenon for stochastic optimization is also closely related to Stein's phenomenon in statistics (Stein 1956; see also Efron and Hastie 2016 for a modern overview). Stein (1956) considered estimating the mean of K normal distributions, each with known variance σ^2, from K data sets. The kth data set is drawn i.i.d. from the kth normal distribution, and draws are independent across k. The natural decoupled solution to the problem (and the maximum likelihood estimate) is to use the kth sample mean as an estimate for the kth distribution. Surprisingly, whereas this estimate is optimal for each problem separately in a very strong sense (uniformly minimum variance unbiased and admissible), Stein (1956) describes a pooled procedure that *always* outperforms this decoupled procedure with respect to total mean-squared error whenever $K \geq 3$.

The proof of Stein's result is remarkably short, but arguably opaque. Many textbooks refer to it as "Stein's paradox," perhaps because it is not clear what drives the result. Why does it always improve upon decoupling? What is special about $K = 3$? Is the key the normality assumption? The common variance assumption? The structure of mean-squared error? All of the above?

Many authors have tried to develop simple intuition for Stein's result (e.g., Brown 1971, Efron and Morris 1977, Stigler 1990, Beran 1996, Brown and Zhao 2012) with mixed success. As a consequence, although Stein's phenomenon has had tremendous impact in statistics, it has, in our humble opinion, had a fairly limited impact on data-driven optimization. It is simply not clear how to generalize Stein's original algorithm to optimization problems different from minimizing mean-squared error. Indeed, the few data-driven optimization methods that attempt to leverage shrinkage apply either to quadratic optimization (e.g., Jorion 1986, DeMiguel et al. 2013, Davarnia and Cornuéjols 2017) or else under Gaussian or near-Gaussian assumptions (Mukherjee et al. 2015, Gupta and Rusmevichientong 2021), both of which are very close to Stein's original setting.

By contrast, our analysis of the data-pooling phenomenon requires very mild distributional assumptions and applies to constrained, potentially nonconvex, nonsmooth optimization problems. Numerical experiments in Section 6 further suggest that even our few assumptions are not crucial to the data-pooling phenomenon. Moreover, our proposed algorithm, Shrunken-SAA, is extremely versatile and can be applied in any setting in which SAA can be applied.

Finally, we note that (in)stability has been well studied in the machine-learning community (see, e.g., Bousquet and Elisseeff 2002, Shalev-Shwartz et al. 2010, Yu 2013, and references therein). Shalev-Shwartz et al. (2010), in particular, argue that stability is the fundamental feature of data-driven algorithms that enables learning. Our Sub-Optimality-Instability Trade-Off connects the data-pooling phenomenon in stochastic optimization to this larger statistical concept. To the best of our knowledge, however, existing theoretical analyses of stability focus on the large-sample regime. Ours is the first work to leverage stability concepts in the small-data, large-scale regime. From a technical perspective, this analysis requires somewhat different tools.

2. Model Setup and the Data-Pooling Phenomenon

As discussed in the introduction, we assume throughout that $\mathbb{P}_k$ has finite, discrete support, that is, $\boldsymbol{\xi}_k \in \{\boldsymbol{a}_{k1}, \ldots, \boldsymbol{a}_{kd}\}$ with $d \geq 2$.[1] Without loss of generality, d is common for all k. To streamline the notation, we write

$$p_{ki} \equiv \mathbb{P}_k(\boldsymbol{\xi}_k = \boldsymbol{a}_{ki}) \text{ and } c_{ki}(\boldsymbol{x}) \equiv c_k(\boldsymbol{x}, \boldsymbol{a}_{ki}), \quad i = 1, \ldots, d.$$

For each k, we let $\mathcal{S}_k = \{\hat{\boldsymbol{\xi}}_{kj} : j = 1, \ldots, \hat{N}_k\}$ be the kth data set with $\hat{\boldsymbol{\xi}}_{kj} \sim \mathbb{P}_k$ drawn i.i.d. Since $\mathbb{P}_k$ is discrete, we can equivalently represent the kth data set $\mathcal{S}_k$ via counts, $\hat{\boldsymbol{m}}_k = (\hat{m}_{k1}, \ldots, \hat{m}_{kd})$, where $\hat{m}_{ki}$ denotes the number of times that $\boldsymbol{a}_{ki}$ occurs in $\mathcal{S}_k$, and $\boldsymbol{e}^\top \hat{\boldsymbol{m}}_k = \hat{N}_k$. In what follows, we will use $\hat{\boldsymbol{m}}_k$ and $\mathcal{S}_k$ interchangeably to refer to the kth data set. We also use "hat" notation $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{m}}, \ldots)$ to denote an observed realization of a random variable, typically a function of $\mathcal{S}_k$.

Because $\hat{\boldsymbol{\xi}}_{kj}$ are i.i.d.,

$$\hat{\boldsymbol{m}}_k \mid \hat{N}_k \sim \text{Multinomial}\big(\hat{N}_k, \boldsymbol{p}_k\big), \quad k = 1, \ldots, K. \tag{2.1}$$

Let $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_K)$ or, equivalently, $\hat{\boldsymbol{m}} = (\hat{\boldsymbol{m}}_1, \ldots, \hat{\boldsymbol{m}}_K)$, denote all the data across all K subproblems, and let $\hat{\boldsymbol{N}} = (\hat{N}_1, \ldots, \hat{N}_K)$ denote the total observation counts. We define $\hat{N}_{\max} = \max_k \hat{N}_k$ and $\hat{N}_{\text{avg}} \equiv \frac{1}{K}\sum_{k=1}^K \hat{N}_k$. Finally, let $\hat{\boldsymbol{p}}_k \equiv \hat{\boldsymbol{m}}_k / \hat{N}_k$ be the empirical distribution for the kth subproblem.

Notice that we have used $\hat{\cdot}$ notation when denoting $\hat{N}_k$ and conditioned on its value in specifying the distribution of $\hat{\boldsymbol{m}}_k$. This is because, in our subsequent analysis, we will sometimes view the amount of data available for each problem as random (see Section 3.2). When the data are fixed and *nonrandom*, we condition on $\hat{N}_k$ explicitly to emphasize this fact.

With this notation, we can rewrite our target optimization problem:

$$Z^* \equiv \min_{\boldsymbol{x}_1 \in \mathcal{X}_1, \ldots, \boldsymbol{x}_K \in \mathcal{X}_K} \frac{1}{K}\sum_{k=1}^K \frac{\lambda_k}{\lambda_{\text{avg}}} \boldsymbol{p}_k{}^\top \boldsymbol{c}_k(\boldsymbol{x}_k). \tag{2.2}$$

Our goal is to identify a data-driven policy, that is, a function $\boldsymbol{x}(\hat{\boldsymbol{m}}) = (\boldsymbol{x}_1(\hat{\boldsymbol{m}}), \ldots, \boldsymbol{x}_K(\hat{\boldsymbol{m}}))$ mapping $\hat{\boldsymbol{m}}$ to $\mathcal{X}_{1 \times \cdots \times \mathcal{X}_K}$ for which $\frac{1}{K}\sum_{k=1}^K \frac{\lambda_k}{\lambda_{\text{avg}}} \boldsymbol{p}_k{}^\top \boldsymbol{c}_k(\boldsymbol{x}_k(\hat{\boldsymbol{m}}))$ is small. We stress that the performance of a data-driven policy is random because it depends on the data.

As mentioned, with full information of $\boldsymbol{p}_k$, Problem (2.2) decouples across k and, after decoupling, no longer depends on the frequency weights $\frac{\lambda_k}{K\lambda_{\text{avg}}}$. Our proposed algorithms will also *not* require knowledge of the weights λ_k. For convenience, we let $\lambda_{\min} = \min_k \lambda_k$ and $\lambda_{\max} = \max_k \lambda_k$.

A canonical policy to which we will compare is the *Sample Average Approximation* (SAA) policy, which proxies the solution of these decoupled problems by replacing $\boldsymbol{p}_k$ with $\hat{\boldsymbol{p}}_k$:

Definition 2.1 (Sample Average Approximation). Let $\boldsymbol{x}_k^{\mathrm{SAA}}(\hat{\boldsymbol{m}}_k) \in \arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} \hat{\boldsymbol{p}}_k^\top \boldsymbol{c}_k(\boldsymbol{x}_k)$ denote the SAA policy for the k^{th} problem, and let $\boldsymbol{x}^{\mathrm{SAA}}(\hat{\boldsymbol{m}}) = \boldsymbol{x}_1^{\mathrm{SAA}}(\hat{\boldsymbol{m}}_1), \ldots, \boldsymbol{x}_K^{\mathrm{SAA}}(\hat{\boldsymbol{m}}_K))$.

As we will see, SAA is closely related to our proposed algorithm Shrunken-SAA and hence provides a natural (decoupled) benchmark when assessing the value of data pooling.

Finally, we use the newsvendor problem as a running example in what follows. We say that the kth subproblem is a *newsvendor problem* with critical fractile $0 < s < 1$ if $c_k(x;\xi) = \max\{\frac{s}{1-s}(\xi - x), (x - \xi)\}$. Its full-information solution is the sth quantile of the kth distribution.

2.1. A Bayesian Perspective of Data Pooling

To motivate data pooling, we first consider a Bayesian approximation to our problem. Specifically, suppose that each $\boldsymbol{p}_k$ were independently drawn from a common Dirichlet prior, that is,

$$\boldsymbol{p}_k \sim \mathrm{Dir}(\boldsymbol{p}_0, \alpha_0), \quad k = 1, \ldots, K,$$

with $\alpha_0 > 0$ and $\boldsymbol{p}_0 \in \Delta_d$, the d-dimensional simplex. The Bayes-optimal decision minimizes the posterior risk, which is $\mathbb{E}[\frac{1}{K}\sum_{k=1}^K \frac{\lambda_k}{\lambda_{\mathrm{avg}}} \boldsymbol{p}_k{}^\top \boldsymbol{c}_k(\boldsymbol{x}_k) \mid \hat{\boldsymbol{m}}] = \frac{1}{K}\sum_{k=1}^K \frac{\lambda_k}{\lambda_{\mathrm{avg}}} \mathbb{E}[\boldsymbol{p}_k \mid \hat{\boldsymbol{m}}]^\top \boldsymbol{c}_k(\boldsymbol{x}_k)$, by linearity. Furthermore, by independence and conjugacy, respectively,

$$\mathbb{E}[\boldsymbol{p}_k \mid \hat{\boldsymbol{m}}] \;=\; \mathbb{E}[\boldsymbol{p}_k \mid \hat{\boldsymbol{m}}_k] \;=\; \frac{\alpha_0}{\hat{N}_k + \alpha_0}\boldsymbol{p}_0 + \frac{\hat{N}_k}{\hat{N}_k + \alpha_0}\hat{\boldsymbol{p}}_k.$$

Hence, a Bayes-optimal solution is $\boldsymbol{x}(\alpha_0, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k) = (\boldsymbol{x}_1(\alpha_0, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_1), \ldots, \boldsymbol{x}_K(\alpha_0, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_K))$, where

$$\hat{\boldsymbol{p}}_k(\alpha) = \left(\frac{\alpha}{\hat{N}_k + \alpha}\boldsymbol{p}_0 + \frac{\hat{N}_k}{\hat{N}_k + \alpha}\hat{\boldsymbol{p}}_k\right), \quad k = 1, \ldots, K, \tag{2.3}$$

$$\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k) \in \arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} \quad \hat{\boldsymbol{p}}_k(\alpha)^\top \boldsymbol{c}_k(\boldsymbol{x}_k), \quad k = 1, \ldots, K. \tag{2.4}$$

For any non-data-driven α and $\boldsymbol{p}_0$, $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ depends on $\hat{\boldsymbol{m}}_k$ but not on $\hat{\boldsymbol{m}}_l$ for $l \neq k$.

This policy has an appealing, intuitive structure. Notice that $\hat{\boldsymbol{p}}_k(\alpha)$ overloads notation slightly and is a convex combination between $\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{p}}_k(0)$, a data-based estimate of $\boldsymbol{p}_k$, and $\boldsymbol{p}_0$, an a priori estimate of $\boldsymbol{p}_k$. In traditional statistical parlance, we say that $\hat{\boldsymbol{p}}_k(\alpha)$ *shrinks* the empirical distribution $\hat{\boldsymbol{p}}_k$ toward the anchor $\boldsymbol{p}_0$. The Bayes-optimal solution is the plug-in solution when using this shrunken empirical measure; that is, it optimizes $\boldsymbol{x}_k$ as though that were the known true measure. This differs from the SAA solution, which is the plug-in solution when using the "unshrunken" $\hat{\boldsymbol{p}}_k$.

The parameter α controls the degree of shrinkage. As $\alpha \to 0$, $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ converges to an SAA solution, and as $\alpha \to \infty$, $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ converges to the (nonrandom) solution to the fully shrunken kth subproblem. In this sense, the Bayes-optimal solution "interpolates" between the SAA solution and the fully shrunken solution. The amount of data $\hat{N}_k$ attenuates the amount of shrinkage; that is, subproblems with more data are shrunk less aggressively for the same α.

Alternatively, we can give a data-pooling interpretation of $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ via the Bayesian notion of pseudocounts. Observe that $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k) \in \arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} (\frac{\alpha \boldsymbol{p}_0 + \hat{\boldsymbol{m}}_k}{\hat{N}_k + \alpha})^\top \boldsymbol{c}_k(\boldsymbol{x}_k)$ and that $\frac{\alpha \boldsymbol{p}_0 + \hat{\boldsymbol{m}}_k}{\hat{N}_k + \alpha}$ is a distribution on $\{\boldsymbol{a}_{k1}, \ldots, \boldsymbol{a}_{kd}\}$. In other words, we can interpret $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ as the solution obtained when we augment each of our original K data sets with α additional "synthetic" data points with counts $\alpha \boldsymbol{p}_0$. As we increase α, we add more synthetic data.

For $\alpha > 0$, $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \boldsymbol{0})$ is the solution to the fully shrunken kth subproblem. For emphasis, let

$$\boldsymbol{x}_k(\infty, \boldsymbol{p}_0) \in \arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} \sum_{i=1}^d p_{0i} c_{ki}(\boldsymbol{x}_k),$$

so that $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \boldsymbol{0}) = \boldsymbol{x}_k(\infty, \boldsymbol{p}_0)$ for all $\alpha > 0$. For completeness, we also define $\boldsymbol{x}_k(0, \boldsymbol{p}_0, \boldsymbol{0}) = \boldsymbol{x}_k(\infty, \boldsymbol{p}_0)$, so that $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \cdot)$ is continuous in α.

In summary, $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ has an intuitive structure that is well defined *regardless of the precise structure of the cost functions* $\boldsymbol{c}_k(\cdot)$ *or feasible region* $\mathcal{X}$. Importantly, this analysis shows that, when the $\boldsymbol{p}_k$'s follow a Dirichlet prior, data pooling by α is never worse than decoupling, and will be strictly better whenever $\boldsymbol{x}_k^{\mathrm{SAA}}(\hat{\boldsymbol{m}}_k)$ is not an optimal solution to the problem defining $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$.

2.2. Data Pooling in a Frequentist Setting

It is perhaps unsurprising that data pooling improves upon the decoupled SAA solution in the Bayesian setting, because problems $l \neq k$ contain information about α and $\boldsymbol{p}_0$, which, in turn, contain information about the $\boldsymbol{p}_k$'s. However, even in frequentist settings, that is, when the $\boldsymbol{p}_k$'s are fixed constants that may have no relationship to one another and there is no "ground-truth" values for α or $\boldsymbol{p}_0$, $\boldsymbol{x}(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ can still improve upon SAA through a careful choice of α and $\boldsymbol{p}_0$ that depend on *all* the data. This is the heart of Stein's result for Gaussian random variables and mean-squared error.

To build intuition, we first study the specific case of minimizing mean-squared error and show that data pooling can improve upon the decoupled SAA solution in the frequentist framework of Equation (2.1). This result is thus reminiscent of Stein's classical result but does not require the Gaussian assumptions. Consider the following example.

Example 2.1 (A Priori Pooling for Mean-Squared Error)**.** Consider a special case of Problem (2.2) such that for all k we have $\lambda_k = \lambda_{\text{avg}}$, $\hat{N}_k = \hat{N} \geq 2$, $\boldsymbol{p}_k$ is supported on $\{a_{k1}, \ldots, a_{kd}\} \subseteq \mathbb{R}$, $\mathcal{X}_k = \mathbb{R}$ and $c_{ki}(x) = (x - a_{ki})^2$. In words, the k^{th} subproblem estimates the unknown mean $\mu_k = \boldsymbol{p}_k^\top \boldsymbol{a_k}$ by minimizing the mean-squared error. Let $\sigma_k^2 = \boldsymbol{p}_k^\top (\boldsymbol{a_k} - \mu_k \boldsymbol{e})^2$.

Fix any $\boldsymbol{p}_0 \in \Delta_d$ and $\alpha \geq 0$ (not depending on the data). A direct computation shows that

$$x_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k) \quad \equiv \quad \hat{\mu}_k(\alpha) \quad \equiv \quad \tfrac{\hat{N}}{\hat{N}+\alpha}\hat{\mu}_k + \tfrac{\alpha}{\hat{N}+\alpha}\mu_{k0},$$

where $\hat{\mu}_k = \frac{1}{\hat{N}}\sum_{i=1}^{\hat{N}} \hat{\xi}_{ki}$ is the usual sample mean and $\mu_{k0} = \boldsymbol{p}_0^\top \boldsymbol{a}_k$. Notice, in particular, that the decoupled SAA solution is $\boldsymbol{x}^{\mathsf{SAA}} = (\hat{\mu}_1, \ldots, \hat{\mu}_K)$, corresponding to $\alpha = 0$.

For any $\boldsymbol{p}_0$ and α, the objective value of $\boldsymbol{x}(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ is

$$\begin{aligned}
&\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{p}_k^\top \boldsymbol{c}_k\big(x_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)\big) \\
&\quad = \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\Big[\big(\hat{\mu}_k(\alpha) - \xi_k\big)^2 \mid \hat{\boldsymbol{m}}\Big] \\
&\quad = \frac{1}{K}\sum_{k=1}^{K}\Big(\sigma_k^2 + \big(\mu_k - \hat{\mu}_k(\alpha)\big)^2\Big),
\end{aligned}$$

by the usual bias-variance decomposition of mean-squared error (MSE). This objective is the average of K independent random variables. Hence, we might intuit that, under appropriate regularity conditions (see Theorem 2.1) and conditional on $\hat{N}$, as $K \to \infty$,

$$\begin{aligned}
&\frac{1}{K}\sum_{k=1}^{K}\Big(\sigma_k^2 + \big(\mu_k - \hat{\mu}_k(\alpha)\big)^2\Big) \\
&\quad - \frac{1}{K}\left(\sum_{k=1}^{K}\sigma_k^2 + \mathbb{E}\Big[\big(\mu_k - \hat{\mu}_k(\alpha)\big)^2 \mid \hat{N}\Big]\right) \quad \to_p \quad 0. \qquad (2.5)
\end{aligned}$$

Moreover, $\frac{1}{K}\left(\sum_{k=1}^{K}\sigma_k^2 + \mathbb{E}\big[(\mu_k - \hat{\mu}_k(\alpha))^2 \mid \hat{N}\big]\right) = \frac{1}{K}\sum_{k=1}^{K}$ $(\sigma_k^2 + (\frac{\alpha}{\hat{N}+\alpha})^2)$ $(\mu_k - \mu_{k0})^2 + (\frac{\hat{N}}{\hat{N}+\alpha})^2 \frac{\sigma_k^2}{\hat{N}}$, again using the bias-variance decomposition of MSE. We can minimize the right-hand side over α explicitly, yielding the value

$$\alpha_{\boldsymbol{p}_0}^{\mathsf{AP}} \quad = \quad \frac{\sum_{k=1}^{K}\sigma_k^2}{\sum_{k=1}^{K}\big(\mu_k - \mu_{k0}\big)^2} \; > 0,$$

where AP stands for a priori, meaning $\alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}$ is the on-average-best a priori choice of shrinkage before observing any data. In particular, substituting $\alpha = 0$ and $\alpha = \alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}$ into the second term of Example 2.5 shows that, up to a term that is vanishing as $K \to \infty$, shrinking by $\alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}$ decreases the MSE by

$$\begin{aligned}
&\left(\frac{1}{K}\sum_{k=1}^{K}\frac{\sigma_k^2}{\hat{N}}\right)\frac{\alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}}{\hat{N} + \alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}} \\
&\quad = \frac{\left(\frac{1}{K\hat{N}}\sum_{k=1}^{K}\sigma_k^2\right)^2}{\frac{1}{K\hat{N}}\sum_{k=1}^{K}\sigma_k^2 + \frac{1}{K}\sum_{k=1}^{K}\big(\mu_k - \mu_{k0}\big)^2} > 0. \qquad (2.6)
\end{aligned}$$

This benefit is strictly positive for any values of $\boldsymbol{p}_k$ and $\boldsymbol{p}_0$, and increasing in $\alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}$.

Unfortunately, we cannot implement $x(\alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ in practice, because $\alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}$ is not computable from the data; it depends on the unknown μ_k and σ_k^2. The next theorem shows that we can, however, estimate $\alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}$ from the data in a way that achieves the same benefit as $K \to \infty$, even if $\hat{N}$ is fixed and small. See Online Appendix A for proof.

Theorem 2.1 (Data Pooling for MSE)**.** *Consider a sequence of subproblems, indexed by $k = 1, 2, \ldots$. Suppose for each k that the k^{th} subproblem minimizes mean-squared error; that is, $\boldsymbol{p}_k$ is supported on $\{a_{k1}, \ldots, a_{kd}\} \subseteq \mathbb{R}$, $\mathcal{X}_k = \mathbb{R}$, and $c_{ki}(x) = (x - a_{ki})^2$. Suppose further that there exists λ_{avg}, $\hat{N} \geq 2$, and $a_{\max} < \infty$ such that $\lambda_k = \lambda_{\text{avg}}$, $\hat{N}_k = \hat{N}$, and $\|\boldsymbol{a}_k\|_\infty \leq a_{\max}$ for all k. Fix any $\boldsymbol{p}_0 \in \Delta_d$, and let*

$$\alpha_{\boldsymbol{p}_0}^{\mathsf{JS}} = \frac{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{\hat{N}-1}\sum_{i=1}^{\hat{N}}\big(\hat{\xi}_{ki} - \hat{\mu}_k\big)^2}{\frac{1}{K}\sum_{k=1}^{K}\big(\mu_{k0} - \hat{\mu}_k\big)^2 - \frac{1}{K\hat{N}}\sum_{k=1}^{K}\frac{1}{\hat{N}-1}\sum_{i=1}^{\hat{N}}\big(\hat{\xi}_{ki} - \hat{\mu}_k\big)^2}.$$

Then, conditional on $\hat{N}$, as $K \to \infty$,

$$\begin{aligned}
&\underbrace{\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{p}_k^\top \boldsymbol{c}_k\big(\boldsymbol{x}_k^{\mathsf{SAA}}\big) - \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{p}_k^\top \boldsymbol{c}_k\Big(\boldsymbol{x}_k\Big(\alpha_{\boldsymbol{p}_0}^{\mathsf{JS}}, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k\Big)\Big)}_{\textit{Benefit over decoupling of } \alpha=\alpha_{\boldsymbol{p}_0}^{\mathsf{JS}}} \\
&\quad - \underbrace{\frac{\left(\frac{1}{K}\sum_{k=1}^{K}\sigma_k^2/\hat{N}\right)^2}{\frac{1}{K}\sum_{k=1}^{K}\sigma_k^2/\hat{N} + \frac{1}{K}\sum_{k=1}^{K}\big(\mu_k - \mu_{k0}\big)^2}}_{\textit{Expected benefit over decoupling of } \alpha=\alpha_{\boldsymbol{p}_0}^{\mathsf{AP}}} \quad \to_p \quad 0.
\end{aligned}$$

Note that $x_k(\alpha_{\boldsymbol{p}_0}^{\mathsf{JS}}, \boldsymbol{p}_0, \hat{\boldsymbol{m}}) = (1 - \theta)\hat{\mu}_k + \theta\hat{\mu}_{k0}$, where $\theta = \frac{1}{\hat{N}}\frac{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{\hat{N}-1}\sum_{i=1}^{\hat{N}}(\hat{\xi}_{ki}-\hat{\mu}_k)^2}{\frac{1}{K}\sum_{k=1}^{K}(\mu_{k0}-\hat{\mu}_k)^2}$. In this form, we can see that the resulting estimator with pooling $\alpha_{\boldsymbol{p}_0}^{\mathsf{JS}}$ strongly resembles the classical James-Stein mean estimator (see Efron and Hastie 2016, equation (7.51)), with the exception that we have replaced the variance σ_k^2, which is assumed to be 1 in Stein's setting, with the usual, unbiased estimator of that variance.

This resemblance motivates our "JS" notation. Theorem 2.1 is neither stronger nor weaker than the James-Stein theorem. Our result applies to non-Gaussian random variables and holds in probability, but is asymptotic; the James-Stein theorem requires Gaussian distributions and holds in expectation, but applies to any fixed $K \geq 3$.

Theorem 2.1 shows that data pooling for mean-squared error always offers a benefit over decoupling for sufficiently large K, no matter what the $\boldsymbol{p}_k$ may be. Data pooling for general optimization problems, however, exhibits more subtle behavior. In particular, as shown in the following example and theorem, there exist instances where data pooling offers no benefit over decoupling, as well as instances where data pooling may be worse than decoupling.

Example 2.2 (Data Pooling for Simple Newsvendor). Consider a special case of Problem (2.2) such that, for all k, $\lambda_k = \lambda_{\text{avg}}$, ξ_k is supported on $\{1, 0\}$, $\mathcal{X}_k = [0, 1]$, and $c_k(x, \xi_k) = |x - \xi_k|$ so that $\boldsymbol{p}_k^\top \boldsymbol{c}_k(x) = p_{k1} + x(1 - 2p_{k1})$. In words, the kth subproblem estimates the median of a Bernoulli random variable by minimizing mean absolute deviation or, equivalently, is a newsvendor problem with critical fractile 0.5 for Bernoulli demand. We order the support so that $p_{k1} = \mathbb{P}(\xi_k = 1)$, as is typical for a Bernoulli random variable. Suppose further that, for each k, $p_{k1} > \frac{1}{2}$, and fix any $p_{01} < \frac{1}{2}$.

Note that $x_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k) = \mathbb{I}[\hat{p}_{k1} \geq \frac{1}{2} + \frac{\alpha}{\hat{N}_k}(\frac{1}{2} - p_{01})]$.[2] Further, for any α (possibly depending on $\hat{\boldsymbol{m}}$),

$$\begin{aligned}
&\boldsymbol{p}_k^\top\big(\boldsymbol{c}_k(\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)) - \boldsymbol{c}_k(\boldsymbol{x}_k(0, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k))\big)\\
&\quad = (2p_{k1} - 1)\left(\mathbb{I}[\hat{p}_{k1} \geq 1/2] - \mathbb{I}\left[\hat{p}_{k1} \geq \frac{1}{2} + \frac{\alpha}{\hat{N}_k}\left(\frac{1}{2} - p_{01}\right)\right]\right)\\
&\quad = (2p_{k1} - 1)\mathbb{I}\left[1/2 \ \leq \ \hat{p}_{k1} \ < \ \frac{1}{2} + \frac{\alpha}{\hat{N}_k}\left(\frac{1}{2} - p_{01}\right)\right],
\end{aligned}$$

where the last equality follows since $\hat{p}_{k1} < 1/2 \Rightarrow \hat{p}_{k1} < \frac{1}{2} + \frac{\alpha}{2}(\frac{1}{2} - p_{01})$. Notice that $p_{k1} > \frac{1}{2} \Rightarrow (2p_{k1} - 1) > 0$, so this last expression is nonnegative. It follows that, path by path, shrinkage by any $\alpha > 0$ cannot improve upon the decoupled solution ($\alpha = 0$). Moreover, if $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k) \neq \boldsymbol{x}_k(0, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$, then the performance is strictly worse. If we had chosen $p_{01} \geq \frac{1}{2}$ and $p_{k1} < \frac{1}{2}$, then a similar result holds.

We summarize this example in the following theorem.

Theorem 2.2 (Data Pooling Does Not Always Offer Benefit). *Given any $\boldsymbol{p}_0$, there exist instances of Problem (2.2) such that shrinkage does not outperform the (decoupled) SAA solution. Moreover, if $\boldsymbol{x}(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ performs the same as SAA, then $\boldsymbol{x}(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ is, itself, an SAA solution.*

On the other hand, there exist examples where the James-Stein estimator and traditional statistical reasoning might suggest the benefits of pooling are marginal, but, by data pooling in a way that exploits the optimization structure, we can achieve significant benefits. Specifically, our Bayesian motivation in Section 2.1 suggests pooling offers little benefit when the $\boldsymbol{p}_k$'s are very dispersed; that is, the Dirichlet prior has high variance and α_0 is small. Similarly, Theorem 2.1 and Efron and Morris (1977) both suggest that the benefits of pooling over decoupling for MSE are marginal if the subproblem means are quite dispersed (see Equation (2.6)). Nonetheless, for general optimization problems, we observe that pooling might still offer substantive benefits in these situations.

Example 2.3. (Pooling Can Offer Benefit Even When $\boldsymbol{p}_k$'s Are Dispersed). Let $d > 3$, and fix some $0 < s < 1$. Suppose that the k^{th} subproblem is a newsvendor problem with critical fractile $f_k > s$ and demand distribution supported on the integers $1, \ldots, d$. For each k, let $p_{k1} = 0$, $p_{kd} = 1 - s$, and $p_{kj_k} = s$ for some $1 < j_k < d$. Consider the fixed anchor $p_{01} = s$, $p_{0d} = 1 - s$, and $p_{0j} = 0$ for $1 < j < d$. Notice that typical $\boldsymbol{p}_k$'s are very far from $\boldsymbol{p}_0$ since $\|\boldsymbol{p}_k - \boldsymbol{p}_0\|_2 = \sqrt{2}s$. For s sufficiently close to 1, this value is close to $\sqrt{2}$, which is the maximal distance between two points on the simplex. In other words, the $\boldsymbol{p}_k$'s are not very similar. Moreover, the means are also dispersed for s close to 1 since $\frac{1}{K}\sum_{k=1}^K (\mu_k - \mu_0)^2 = s^2 \frac{1}{K}\sum_{k=1}^K (j_k - 1)^2 \approx s^2 d/2$ if the j_k's are chosen uniformly.

Consequently, the James-Stein estimator does not shrink very much in this example. A straightforward computation shows that, for K sufficiently large, $\alpha_{\boldsymbol{p}_0}^{\mathsf{JS}} \leq \frac{(1-s)d^2}{s}$ with high probability, which is close to 0 for s close to 1. However, the full-information solution for the k^{th} problem is $\boldsymbol{x}_k^* = d$, which *also* equals the fully pooled ($\alpha = \infty$) solution, $\boldsymbol{x}_k(\infty, \boldsymbol{p}_0)$. Hence, pooling in an optimization-aware way can achieve full-information performance, whereas both decoupling and an "estimate-then-optimize" approach using James-Stein shrinkage *necessarily* perform worse. In other words, pooling offers significant benefits despite the $\boldsymbol{p}_k$'s being as dispersed as possible, because of the optimization structure, and leveraging this structure is necessary to obtain the best shrinkage. □

Theorems 2.1 and 2.2 and Examples 2.2 and 2.3 highlight the fact that data pooling for general optimization is more complex than Stein's phenomenon. In particular, in Stein's classical result for mean-squared error and Gaussian data, data pooling *always* offers a benefit for $K \geq 3$. For other optimization problems and data distributions, data pooling may *not* offer a benefit, or it may offer a benefit but requires a new way of choosing the pooling amount.

An interplay between $\boldsymbol{p}_0$, $\boldsymbol{p}_k$, and $\boldsymbol{c}_k$ determines if data pooling can improve upon decoupling and how much pooling is best.

This raises two important questions: First, how do we identify if an instance of Problem (2.2) would benefit from data pooling? Second, if it does, how do we compute the "optimal" amount of pooling? In the next sections, we show how our Shrunken-SAA algorithm can be used to address both questions in the relevant regime, where K is large but the average amount of data per subproblem remains small. Indeed, we show that Shrunken-SAA achieves the best-possible shrinkage in an optimization-aware fashion for many types of problems and choices of anchor.

3. The Shrunken SAA Algorithm

Algorithm 1 (The Shrunken-SAA Algorithm)

```
Input: Data $\mathcal{S}_k = \{\hat{\xi}_{k1}, \dots, \hat{\xi}_{k\hat{N}_k}\}$, $k = 1, \dots, K$, and an anchor distribution $h(\mathcal{S})$
Fix a finite grid $\mathcal{A} \subseteq [0, \infty)$
for $\alpha \in \mathcal{A}$, $k = 1, \dots, K$, $j = 1, \dots, \hat{N}_k$ define:
$\boldsymbol{x}_{k,-j}(\alpha, h(\mathcal{S})) \leftarrow \arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} \sum_{\ell \neq j} c_k(\boldsymbol{x}_k, \hat{\xi}_{k\ell}) + \alpha \mathbb{E}_{\xi_k \sim h(\mathcal{S})}[c_k(\boldsymbol{x}_k, \xi_k)]$ // Compute leave-one-out (LOO) solutions
end for
$\alpha_h^{\text{S-SAA}} \leftarrow \arg\min_{\alpha \in \mathcal{A}} \sum_{k=1}^{K} \sum_{j=1}^{\hat{N}_k} c_k(\boldsymbol{x}_{k,-j}(\alpha, h(\mathcal{S})), \hat{\xi}_{kj})$ // Modified LOO Cross-Validation
for all $k = 1, \dots, K$ do
    $\boldsymbol{x}_k^{\text{S-SAA}} \leftarrow \arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} \sum_{j=1}^{\hat{N}_k} c_k(\boldsymbol{x}_k, \hat{\xi}_{kj}) + \alpha_h^{\text{S-SAA}} \mathbb{E}_{\xi_k \sim h(\mathcal{S})}[c_k(\boldsymbol{x}_k, \xi_k)]$ // Compute pooled solution
end for
return $(\boldsymbol{x}_1^{\text{S-SAA}}, \dots, \boldsymbol{x}_K^{\text{S-SAA}})$
```

Algorithm 1 formally defines Shrunken-SAA. The crucial step is the "Modified LOO Cross-Validation," which we discuss in detail in Sections 3.2 and 3.3. To highlight similarities to SAA, we have stated the algorithm in terms of the data sets $\mathcal{S}_k$ and $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_K)$. Here, $h(\mathcal{S})$ represents an arbitrary, possibly data-driven anchor distribution (we present some examples, shortly). Recall that we can equivalently express $\mathcal{S}_k$ in terms of the counts $\hat{\boldsymbol{m}}_k$. In that notation, we recognize that if the j^{th} data point of $\mathcal{S}_k$ is $\boldsymbol{a}_{ki}$, then $\boldsymbol{x}_{k,-j}(\alpha, h(\hat{\boldsymbol{m}})) = \boldsymbol{x}_k(\alpha, h(\hat{\boldsymbol{m}}), \hat{\boldsymbol{m}}_k - \boldsymbol{e}_i)$ and $\boldsymbol{x}_k^{\text{S-SAA}} = \boldsymbol{x}_k(\alpha^{\text{S-SAA}}, h(\hat{\boldsymbol{m}}), \hat{\boldsymbol{m}}_k)$. In other words, Shrunken-SAA retains the particular pooling structure of Equation (2.4) suggested by our Bayesian argument, but it allows for a data-dependent anchor $h(\mathcal{S})$ (equivalently, $h(\hat{\boldsymbol{m}})$) and chooses the amount of pooling via a particular cross-validation scheme. We present Algorithm 1 using a finite grid $\mathcal{A}$, but our theory also pertains to $\mathcal{A} = [0, \infty)$.

Remark 3.1 (Computational Complexity)**.** Computationally, Algorithm 1 does not depend on d, the size of the support of ξ_k. Its bottleneck is computing $\boldsymbol{x}_{k,-j}$, which is similar to solving the k^{th} subproblem by SAA with an augmented data set described by $h(\mathcal{S})$. More specifically, Algorithm 1 depends on the data only through $h(\mathcal{S})$ and averages of functions over subsets of $\mathcal{S}$, neither of which explicitly depends upon d. Consequently, although our setup and analysis assume that ξ_k has finite discrete support, from an implementation perspective, we can apply Shrunken-SAA when ξ_k has continuous support *without* discretization, so long as we can efficiently solve these augmented SAA problems (see our empirical study in Section E.6 of Online Appendix E). From a theoretical perspective, some of our analysis extends to this continuous setting (see Section 4.6). In the remainder, we follow Section 2 and treat the data as discrete, referring to the data by $\hat{\boldsymbol{m}}_k$ and $\hat{\boldsymbol{m}}$.

We consider Shrunken-SAA to be *roughly* as tractable as SAA. We say "roughly" because, in the worst case, one must solve at most $|\mathcal{A}| \sum_{k=1}^{K} \min(d, \hat{N}_k)$ problems in the LOO cross-validation step, which, if we sample from $h(\hat{\boldsymbol{m}})$, have a similar structure to SAA. Fortunately, we can parallelize these problems in distributed computing environments and use previous iterations to "warm-start" solvers. Moreover, in Section E.8 of Online Appendix E, we observe empirically that less computationally expensive κ-fold cross-validation procedures can be used in place of LOO with similar performance. □

For clarity, the $\alpha_h^{\text{S-SAA}}$ parameter (with $\mathcal{A} = [0, \infty)$) computed by Algorithm 1 is

$$\alpha_h^{\text{S-SAA}} \in \arg\min_{\alpha \geq 0} \sum_{k=1}^{K} \hat{\boldsymbol{m}}_k^{\top} \boldsymbol{c}_k(\boldsymbol{x}_k(\alpha, h(\hat{\boldsymbol{m}}), \hat{\boldsymbol{m}}_k - \boldsymbol{e}_i)). \quad (3.1)$$

The Anchor Distribution $h(\hat{\boldsymbol{m}})$

As stated, the anchor in Algorithm 1, $h(\hat{\boldsymbol{m}})$, is an input. We think of $h(\hat{\boldsymbol{m}})$ as a function that selects an anchor distribution from a candidate set of distributions $\mathcal{P}$. In what follows, we will focus on two types of anchors and corresponding candidate sets $\mathcal{P}$:

- **Fixed anchors**: Here, $h(\hat{\boldsymbol{m}}) = \boldsymbol{p}_0$, $\mathcal{P} = \{\boldsymbol{p}_0\}$ for some fixed $\boldsymbol{p}_0$, for example, the uniform distribution $\boldsymbol{p}_0 = \boldsymbol{e}/d$. Fixed anchors might be used for computational/statistical simplicity or when there is strong a priori knowledge of a good anchor. In this case, we abuse notation slightly, replacing the map $h : \hat{\boldsymbol{m}} \mapsto \boldsymbol{p}_0$ with the constant $\boldsymbol{p}_0$ when clear from context; for exampe, we write $\alpha_{\boldsymbol{p}_0}^{\text{S-SAA}}$ for $\alpha_h^{\text{S-SAA}}$.
- **Data-driven anchors**: Here, $h(\hat{\boldsymbol{m}})$ is any procedure that uses the data $\hat{\boldsymbol{m}}$ to select a distribution, and $\mathcal{P}$ is the image of $h(\cdot)$. One example might be to use all the data to fit a parametric distribution, for example, a lognormal distribution, via maximum likelihood

and use this fitted distribution as the anchor. Then, $\mathcal{P}$ would be the set of lognormal distributions.

We also pay particular focus to two special cases of data-driven anchors in what follows:

- **LOO-optimized anchor**: For a given $\mathcal{P} \subseteq \Delta_d$, let

$$h_{\mathcal{P}}(\hat{\boldsymbol{m}}) \in \arg\min_{\boldsymbol{q}\in\mathcal{P}} \min_{\alpha\in\mathcal{A}} \sum_{k=1}^{K} \hat{\boldsymbol{m}}_k^\top \boldsymbol{c}_k\big(\boldsymbol{x}_k(\alpha, \boldsymbol{q}, \hat{\boldsymbol{m}}_k - \boldsymbol{e}_i\big). \qquad (3.2)$$

We will see later that $h_{\mathcal{P}}$ satisfies stronger optimality properties than general data-driven anchors and, hence, we treat it separately. From an implementation point of view, when applying Algorithm 1, we only ever require the value of $h_{\mathcal{P}}(\hat{\boldsymbol{m}})$, not the full-function $h_{\mathcal{P}}(\cdot)$. Thus, Algorithm 1 with $h_{\mathcal{P}}(\cdot)$ amounts to replacing the "Modified LOO Cross-Validation" step by a joint optimization over anchor and pooling amount:

$$\left(\alpha^{\text{S-SAA}}_{h_{\mathcal{P}}}, h_{\mathcal{P}}(\hat{\boldsymbol{m}})\right) \leftarrow \arg\min_{\alpha\in\mathcal{A}, \boldsymbol{q}\in\mathcal{P}} \sum_{k=1}^{K} \sum_{j=1}^{\hat{N}_k} c_k\left(\boldsymbol{x}_{k,-j}(\alpha, \boldsymbol{q}), \hat{\boldsymbol{\xi}}_{kj}\right). \qquad (3.3)$$

We note that the multivariate optimization problem in Section 3.3 may be challenging depending on the structure of $\mathcal{P}$, motivating our second special case:

- **GM-anchor:** We also consider a computationally simpler "grand-mean" anchor $h(\hat{\boldsymbol{m}}) = \hat{\boldsymbol{p}}^{\text{GM}}$, where $\hat{\boldsymbol{p}}^{\text{GM}} \equiv \sum_{k=1}^{K} \hat{\boldsymbol{p}}_k \mathbb{I}\big[\hat{N}_k > 0\big] / \sum_{k=1}^{K} \mathbb{I}\big[\hat{N}_k > 0\big]$ if $\hat{N}_{\max} > 0$ and $\boldsymbol{e}/d$ otherwise. (For this data-driven anchor, $\mathcal{P} = \Delta_d$.) This choice is motivated by our Bayesian perspective on data pooling from Section 2.1. In the Bayesian setting, $\hat{\boldsymbol{p}}^{\text{GM}}$ is an unbiased estimator of the prior mean. We observe empirically in Section 6 that $\hat{\boldsymbol{p}}^{\text{GM}}$ is a strong and computationally efficient heuristic.

3.1. Oracle Benchmarks

From Theorem 2.2, data pooling need not improve upon decoupling for a given $h(\cdot)$. To establish appropriate benchmarks, we first define the *oracle* pooling for given $h(\cdot)$; that is,

$$\alpha_h^{\text{OR}} \in \arg\min_{\alpha\geq 0} \overline{Z}_K(\alpha, h(\hat{\boldsymbol{m}})), \quad \text{where}$$
$$\overline{Z}_K(\alpha, \boldsymbol{q}) = \tfrac{1}{K}\sum_{k=1}^{K} Z_k(\alpha, \boldsymbol{q}),$$
$$Z_k(\alpha, \boldsymbol{q}) = \tfrac{\lambda_k}{\lambda_{\text{avg}}} \boldsymbol{p}_k{}^\top \boldsymbol{c}_k\big(\boldsymbol{x}_k(\alpha, \boldsymbol{q}, \hat{\boldsymbol{m}}_k)\big). \qquad (3.4)$$

Notice that α_h^{OR} is random, depending on the entire data-sequence. By construction, $\overline{Z}_K(\alpha_h^{\text{OR}}, h(\hat{\boldsymbol{m}}))$ lower bounds the performance of *any* other data-driven pooling policy with anchor $h(\hat{\boldsymbol{m}})$ path by path. Hence, it serves as a strong performance benchmark. However, α_h^{OR} also depends on the unknown $\boldsymbol{p}_k$ and λ_k, and, hence, is not implementable in practice. In this sense, it is an oracle.

Given any α (possibly depending on the data), we measure the suboptimality of pooling by α relative to the oracle pooling for $h(\cdot)$ on a particular data realization by

$$\text{SubOpt}_{h,K}(\alpha) = \overline{Z}_K(\alpha, h(\hat{\boldsymbol{m}})) - \overline{Z}_K\big(\alpha_h^{\text{OR}}, h(\hat{\boldsymbol{m}})\big).$$

Good pooling procedures have small suboptimality with high probability with respect to the data. We allow for $\alpha_h^{\text{OR}} = 0$. Thus, procedures with small suboptimality still have good performance in instances when data pooling is not beneficial. Studying when $\alpha_h^{\text{OR}} > 0$ further gives intuition into when and why data pooling is helpful, a task we take up in Section 5.

The aforementioned oracle is defined with respect to a given anchor. One might also seek to benchmark performance relative to the best-possible anchor. Given any $\mathcal{P} \subseteq \Delta_d$, we define the oracle choice of anchor and pooling amount for anchors in $\mathcal{P}$ and for a particular data realization by

$$\big(\alpha_{\mathcal{P}}^{\text{OR}}, \boldsymbol{q}_{\mathcal{P}}^{\text{OR}}\big) \in \arg\min_{\alpha\geq 0, \boldsymbol{q}\in\mathcal{P}} \overline{Z}_K(\alpha, \boldsymbol{q}). \qquad (3.5)$$

Then, given any anchor $\boldsymbol{q} \in \mathcal{P}$ and pooling amount α (both possibly depending the data), we measure the suboptimality of shrinking by α toward $\boldsymbol{q}$ by

$$\text{SubOpt}_{\mathcal{P},K}(\alpha, \boldsymbol{q}) \;=\; \overline{Z}_K(\alpha, \boldsymbol{q}) - \overline{Z}_K\big(\alpha_{\mathcal{P}}^{\text{OR}}, \boldsymbol{q}_{\mathcal{P}}^{\text{OR}}\big).$$

For clarity, we observe that, by construction, $\alpha_{\mathcal{P}}^{\text{OR}} = \alpha^{\text{OR}}_{\boldsymbol{q}_{\mathcal{P}}^{\text{OR}}}$.

3.2. Motivating $\alpha^{\text{S-SAA}}$ Through Unbiased Estimation

We first consider a fixed anchor $h(\hat{\boldsymbol{m}}) = \boldsymbol{p}_0$. In this case, we abuse notation slightly and write

$$\alpha_{\boldsymbol{p}_0}^{\text{OR}} \in \arg\min_{\alpha\geq 0} \overline{Z}_K(\alpha, \boldsymbol{p}_0). \qquad (3.6)$$

One approach to choosing $\alpha_{\boldsymbol{p}_0}$ might be to construct a suitable proxy for $\overline{Z}_K(\alpha, \boldsymbol{p}_0)$ in Equation (3.6) based only on the data and then choose the $\alpha_{\boldsymbol{p}_0}$ that optimizes this proxy.

If we knew the values of λ_k, then a natural proxy might be to replace the unknown $\boldsymbol{p}_k$ with $\hat{\boldsymbol{p}}_k$; that is, optimize $\frac{1}{K}\sum_{k=1}^{K} \frac{\lambda_k}{\lambda_{\text{avg}}} \hat{\boldsymbol{p}}_k{}^\top \boldsymbol{c}_k(\boldsymbol{x}_k(\alpha, \boldsymbol{q}, \hat{\boldsymbol{m}}_k))$. Unfortunately, even for a fixed, non-data-driven α, this proxy is *biased*, that is, $\mathbb{E}[\frac{1}{K}\sum_{k=1}^{K} \frac{\lambda_k}{\lambda_{\text{avg}}} \hat{\boldsymbol{p}}_k{}^\top \boldsymbol{c}_k(\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k))] \neq \mathbb{E}\big[\overline{Z}_K(\alpha, \boldsymbol{p}_0)\big]$, since both $\hat{\boldsymbol{p}}_k$ and $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ depend on the data $\hat{\boldsymbol{m}}_k$. Worse, this bias wrongly suggests that $\alpha = 0$, that is, decoupling, is always a good policy, because $\boldsymbol{x}_k(0, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ always optimizes this proxy, by construction. By contrast, Theorem 2.1 shows that data pooling can offer significant benefits. This type of bias and its consequences are well known in other contexts and are often termed the "optimizer's curse"—in-sample costs are optimistically biased and may not generalize well.

These features motivate us to seek an unbiased estimate of $\overline{Z}_K(\alpha, \boldsymbol{p}_0)$. At first glance, however, $Z_K(\alpha, \boldsymbol{p}_0)$, which depends on both the unknown $\boldsymbol{p}_k$ and unknown λ_k, seems particularly intractable unless $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ admits a closed-form solution as in Example 2.1. A key observation is that, in fact, $\overline{Z}_K(\alpha, \boldsymbol{p}_0)$ does more generally admit an unbiased estimator, *if* we also introduce an additional assumption on our data-generating mechanism, that is, that the amount of data is random.

Assumption 3.1 (Randomizing Amount of Data)**.** *There exists an N such that $\hat{N}_k \sim Poisson(N\lambda_k)$ for each $k = 1, \ldots, K$.*

Under Assumption 3.1, (unconditional) expectations and probabilities should be interpreted as over both the random draw of $\hat{N}_k$ and the counts $\hat{\boldsymbol{m}}_k$.

Analytically, the benefit of Assumption 3.1 is that it breaks the dependence across i in $\hat{\boldsymbol{m}}_k$. Namely, by the Poisson-splitting property, under Assumption 3.1,

$$\hat{m}_{ki} \sim Poisson(m_{ki}) \quad \text{where} \quad m_{ki} \equiv N\lambda_k p_{ki}, \qquad i = 1, \ldots, d, \quad k = 1, \ldots, K,$$

and the $\hat{m}_{ki}$'s are independent across i and k. If $\hat{N}_k$ were nonrandom, then these $\hat{m}_{ki}$ would be dependent.

Beyond its analytical convenience, we consider Assumption 3.1 to be reasonable in many applications. Consider, for instance, a retailer optimizing the price of k distinct products; that is, x_k represents the price of product k, ξ_k represents the (random) valuation of a typical customer, and $c_k(x_k, \xi_k)$ is the (negative) profit earned. In such settings, one frequently ties data collection to time; that is, one might collect $N = 6$ months worth of data. To the extent that customers arrive seeking product k in a random fashion, the number of arrivals $\hat{N}_k$ that one might observe in N months is, itself, random and is reasonably modeled as Poisson with rate proportional to N. Similar statements apply whenever data for problem k are generated by an event that occurs randomly, for example, when observing the response time of emergency responders (disasters occur intermittently), effectiveness of a new medical treatment (patients with the relevant disease arrive sequentially), or any aspect of a customer service interaction (customers arrive randomly to service).

In some ways, this perspective tacitly underlies the formulation of Problem (2.2) itself. Indeed, one way to interpret the subproblem weights $\frac{\lambda_k}{K\lambda_{\text{avg}}} = \frac{\lambda_k}{\sum_{j=1}^K \lambda_j}$ is that the decision maker incurs costs $c_k(x_k, \xi_k)$ at rate λ_k, so that problems of type k contribute a $\frac{\lambda_k}{\sum_{j=1}^K \lambda_j}$ fraction of the total long-run costs. However, if problems of type k occur at rate λ_k, it should be that observations of type k, that is, realizations of $\boldsymbol{\xi}_k$, also occur at rate λ_k, supporting Assumption 3.1.

In settings where data collection is not tied to randomly occurring events, modeling $\hat{N}_k$ as Poisson may still be a reasonable approximation if d is large relative to $\hat{N}_k$ and each of the individual p_{ki}'s are small. Indeed, under such assumptions, a *Multinomial*$(\hat{N}_k, \boldsymbol{p}_k)$ is well approximated by independent Poisson random variables with rates $\hat{N}_k p_{ki}$, $i = 1, \ldots d$ (see McDonald 1980, Deheuvels and Pfeifer 1988 for a formal statement). In this sense, we can view the consequence of Assumption 3.1 as a useful approximation to the setting where the $\hat{N}_k$'s are fixed, even if it is not strictly true.

In any case, under Assumption 3.1, we develop an unbiased estimate for $\overline{Z}_K(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$. We use the following identity (Chen 1975). For any $f : \mathbb{Z}_+ \to \mathbb{R}$, for which the expectations exist,

$$W \sim \text{Poisson}(\lambda) \Rightarrow \lambda\mathbb{E}\big[f(W+1)\big] = \mathbb{E}\big[Wf(W)\big]. \quad (3.7)$$

The proof of the identity is immediate from the Poisson probability mass function.[3]

Now, for any $\alpha \geq 0$ and $\boldsymbol{q} \in \Delta_d$, define

$$Z_k^{\text{LOO}}(\alpha, \boldsymbol{q}) \equiv \frac{1}{N\lambda_{\text{avg}}} \sum_{i=1}^d \hat{m}_{ki} c_{ki}\big(\boldsymbol{x}_k(\alpha, \boldsymbol{q}, \hat{\boldsymbol{m}}_k - \boldsymbol{e}_i)\big),$$
$$\text{and} \quad \overline{Z}_K^{\text{LOO}}(\alpha, \boldsymbol{q}) \equiv \frac{1}{K} \sum_{k=1}^K Z_k^{\text{LOO}}(\alpha, \boldsymbol{p}_0). \quad (3.8)$$

Lemma 3.1. (An Unbiased Estimator for $\overline{Z}_K(\alpha, \boldsymbol{p}_0)$). *Under Assumption* 3.1, *we have for any $\alpha \geq 0$ and $\boldsymbol{q} \in \Delta_d$ that $\mathbb{E}[Z_k^{\text{LOO}}(\alpha, \boldsymbol{q})] = \mathbb{E}[Z_k(\alpha, \boldsymbol{q})]$. In particular, $\mathbb{E}\Big[\overline{Z}_K^{\text{LOO}}(\alpha, \boldsymbol{q})\Big] = \mathbb{E}\big[\overline{Z}_K(\alpha, \boldsymbol{q})\big]$.*

Proof. Recall that $Z_k(\alpha, \boldsymbol{q}) = \frac{1}{N\lambda_{\text{avg}}} \sum_{i=1}^d m_{ki} c_{ki}(\boldsymbol{x}_k(\alpha, \boldsymbol{q}, \hat{\boldsymbol{m}}_k))$ and that, under Assumption 3.1, $\hat{m}_{ki} \sim Poisson(m_{ki})$ independently over $i = 1, \ldots, d$. Let $\hat{m}_{k,-i}$ denote $(\hat{m}_{k,j})_{j \neq i}$. Then, by Equation (3.7),

$$\mathbb{E}\big[m_{ki} c_{ki}\big(\boldsymbol{x}_k(\alpha, \boldsymbol{q}, \hat{\boldsymbol{m}}_k)\big) \mid \hat{m}_{k,-i}\big] = \mathbb{E}\big[\hat{m}_{ki} c_{ki}\big(\boldsymbol{x}_k(\alpha, \boldsymbol{q}, \hat{\boldsymbol{m}}_k - \boldsymbol{e}_i)\big) \mid \hat{m}_{k,-i}\big].$$

Taking expectations of both sides, summing over $i = 1, \ldots, d$, and scaling by $N\lambda_{\text{avg}}$ proves $\mathbb{E}[Z_k^{\text{LOO}}(\alpha, \boldsymbol{q})] = \mathbb{E}\big[Z_k(\alpha, \boldsymbol{q})\big]$. Finally, averaging this last equality over k completes the lemma.

We therefore propose selecting α by minimizing the estimate $\overline{Z}_K^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0)$. As written, $\overline{Z}_K^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0)$ still depends on the unknown N and λ_{avg}; however, these values occur multiplicatively and are positive, and so do not affect the optimizer. Hence, the optimizer is exactly $\alpha_h^{\mathsf{S-SAA}}$, as in Equation (3.1).

3.3. Motivating $\alpha^{\mathsf{S-SAA}}$ via Modified Leave-One-Out Cross-Validation

Although we motivated Equation (3.1) via an unbiased estimator, we can alternatively motivate it through leave-one-out cross-validation. This latter perspective informs our "LOO" notation. Indeed, consider again our decision maker, and assume in line with Assumption 3.1 that subproblems of type k arrive randomly according to a Poisson process with rate λ_k, independently across k. When a problem of type k arrives, she incurs a cost $c_k(\boldsymbol{x}_k, \boldsymbol{\xi})$. Again, the objective of Problem (2.2) thus represents her expected long-run costs.

We can alternatively represent her costs via the modified cost function $C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K, \kappa, \boldsymbol{\xi}) = c_\kappa(\boldsymbol{x}_\kappa, \boldsymbol{\xi})$, where κ is a random variable indicating which of the k subproblems she is currently facing. In particular, letting $\mathbb{P}(\kappa = k) = \frac{\lambda_k}{K\lambda_{\text{avg}}}$ and $\mathbb{P}(\boldsymbol{\xi} = \boldsymbol{a}_{ki} \mid \kappa = k) = p_{ki}$, the objective of Problem (2.2) can be more compactly written $\mathbb{E}[C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K, \kappa, \boldsymbol{\xi})]$.

Now consider pooling all the data into a single "grand" data set of size $\hat{N}_1 + \cdots + \hat{N}_K$:

$$\left\{\left(k, \boldsymbol{\xi}_{kj}\right) : j = 1, \ldots, \hat{N}_k,\ k = 1, \ldots, K\right\}.$$

The grand data set can be seen as i.i.d. draws of $(\kappa, \boldsymbol{\xi})$.

For a fixed α and $\boldsymbol{p}_0$, the leave-one-out estimate of $\mathbb{E}\big[C(\boldsymbol{x}_1(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}), \ldots, \boldsymbol{x}_K(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}), \kappa, \boldsymbol{\xi})\big]$ is given by removing one data point from the grand data set, training $\boldsymbol{x}_1(\alpha, \boldsymbol{p}_0, \cdot), \ldots, \boldsymbol{x}_K(\alpha, \boldsymbol{p}_0, \cdot)$ on the remaining data, and evaluating $C(\cdot)$ on the left-out point using these policies. We repeat for each point in the grand data set and average. We can rewrite this leave-one-out estimate as

$$\frac{1}{\sum_{k=1}^K \hat{N}_k} \sum_{k=1}^K \sum_{i=1}^d \hat{m}_{ki} c_{ki}\big(\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k - \boldsymbol{e}_i)\big),$$

which agrees with the objective of Equation (3.1) up to a positive multiplicative constant. Although this multiplicative constant does not affect the choice of $\alpha^{\mathsf{S-SAA}}$, it *does* cause the traditional leave-one-out estimator to be *biased*. This bias agrees with folklore results in machine learning that assert that leave-one-out does generally exhibit a small bias (Hastie et al. 2001).

For data-driven anchors, we stress that, unlike traditional leave-one-out validation, we do *not* use one fewer points when computing the anchor in Algorithm 1; we use $h(\hat{\boldsymbol{m}})$ for all iterations. Hence, Shrunken-SAA is *not* strictly a leave-one-out procedure, motivating our qualifier "Modified."

4. Performance Guarantees for Shrunken-SAA

In this section, we show that, in the limit where the number of subproblems K grows, shrinking by $\alpha_h^{\mathsf{S-SAA}}$ is essentially best possible. More precisely, for any $K \geq 2$ and any $0 < \delta < 1/2$, with probability at least $1 - \delta$, we prove that

$$\mathsf{SubOpt}_{h,K}(\alpha_h^{\mathsf{S-SAA}}) \leq \tilde{\mathcal{O}}\left(\frac{\log^\beta(1/\delta)}{\sqrt{K}}\right), \tag{4.1}$$

where the $\tilde{\mathcal{O}}(\cdot)$ notation suppresses logarithmic factors in K, and $1 < \beta < 2$ is a constant that depends on the particular class of optimization problems under consideration. Imporantly, by the Borel-Cantelli lemma, Equation (4.1) implies $\mathsf{SubOpt}_{h,K}(\alpha_h^{\mathsf{S-SAA}}) \to 0$, almost surely as $K \to \infty$, *even* if the expected amount of data per subproblem remains fixed.

Equation (4.1) asserts that for a *given* anchor $h(\cdot)$, Shrunken-SAA achieves the best possible shrinkage amount as $K \to \infty$. We will also prove similar bounds on $\mathsf{SubOpt}_{\mathcal{P},K}(\alpha_h^{\mathsf{S-SAA}}, h_{\mathcal{P}}(\hat{\boldsymbol{m}}))$. Such bounds assert that, for a given class $\mathcal{P}$, Shrunken-SAA with $h_{\mathcal{P}}(\cdot)$ achieves the best possible anchor and shrinkage amount *simultaneously*.

4.1. Overview of Proof Technique

To prove performance guarantees like Equation (4.1), we first bound the suboptimality of Shrunken-SAA in terms of the maximal stochastic deviations of $\overline{Z}_K(\alpha, h)$ and $\overline{Z}_K^{\mathsf{LOO}}(\alpha, h)$ from their means.

Lemma 4.1 (Bounding Sub-Optimality)**.** *Suppose that Assumption 3.1 holds.*

For a non-data-driven anchor $h(\hat{\boldsymbol{m}}) = \boldsymbol{p}_0$,

$$\begin{aligned}
&\mathsf{SubOpt}_{\boldsymbol{p}_0,K}\left(\alpha_{\boldsymbol{p}_0}^{\mathsf{S-SAA}}\right)\\
&\quad\leq 2 \underbrace{\sup_{\alpha\geq 0}\left|\overline{Z}_K(\alpha, \boldsymbol{p}_0) - \mathbb{E}[\overline{Z}_K(\alpha, \boldsymbol{p}_0)]\right|}_{\textit{Maximal Stochastic Deviation in } \overline{Z}_K(\cdot, \boldsymbol{p}_0)} +\\
&\quad\quad 2\underbrace{\sup_{\alpha\geq 0}\left|\overline{Z}_K^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0,) - \mathbb{E}\left[\overline{Z}_K^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0)\right]\right|}_{\textit{Maximal Stochastic Deviation in } \overline{Z}_K^{\mathsf{LOO}}(\cdot, \boldsymbol{p}_0)}.
\end{aligned}$$

Similarly, for a general data-driven anchor with $h(\hat{\boldsymbol{m}}) \in \mathcal{P}$*,*

$$\begin{aligned}&\mathsf{SubOpt}_{h,K}(\alpha_h^{\mathsf{S\text{-}SAA}}) \\ &\quad\le 2 \underbrace{\sup_{\substack{\alpha\ge 0\\ \boldsymbol{q}\in\mathcal{P}}} \left|\overline{Z}_K(\alpha,\boldsymbol{q}) - \mathbb{E}[\overline{Z}_K(\alpha,\boldsymbol{q})]\right|}_{\textit{Maximal Stochastic Deviation in } \overline{Z}_K(\cdot,\cdot)} + \\ &\quad\quad 2 \underbrace{\sup_{\substack{\alpha\ge 0\\ \boldsymbol{q}\in\mathcal{P}}} \left|\overline{Z}_K^{\mathsf{LOO}}(\alpha,\boldsymbol{q}) - \mathbb{E}\left[\overline{Z}_K^{\mathsf{LOO}}(\alpha,\boldsymbol{q})\right]\right|}_{\textit{Maximal Stochastic Deviation in } \overline{Z}_K^{\mathsf{LOO}}(\cdot,\cdot)}.\end{aligned} \tag{4.2}$$

Finally, for $h = h_{\mathcal{P}}$*,* $\mathsf{SubOpt}_{\mathcal{P},K}(\alpha_{h_{\mathcal{P}}}^{\mathsf{S\text{-}SAA}}, h_{\mathcal{P}}(\hat{\boldsymbol{m}}))$ *is also bounded by the right-hand side of* Equation (4.2).

Proof. By definition of $\alpha_{\boldsymbol{p}_0}^{\mathsf{S\text{-}SAA}}$, $\overline{Z}_K^{\mathsf{LOO}}(\alpha_{\boldsymbol{p}_0}^{\mathsf{OR}}, \boldsymbol{p}_0) - \overline{Z}_K^{\mathsf{LOO}}(\alpha_{\boldsymbol{p}_0}^{\mathsf{S\text{-}SAA}}, \boldsymbol{p}_0) \ge 0$. Therefore,

$$\begin{aligned}&\mathsf{SubOpt}_{\boldsymbol{p}_0,K}\left(\alpha_{\boldsymbol{p}_0}^{\mathsf{S\text{-}SAA}}\right) \\ &\quad\le \overline{Z}_K\left(\alpha_{\boldsymbol{p}_0}^{\mathsf{S\text{-}SAA}}, \boldsymbol{p}_0\right) - \overline{Z}_K\left(\alpha_{\boldsymbol{p}_0}^{\mathsf{OR}}, \boldsymbol{p}_0\right) \\ &\quad\quad + \overline{Z}_K^{\mathsf{LOO}}\left(\alpha_{\boldsymbol{p}_0}^{\mathsf{OR}}, \boldsymbol{p}_0\right) - \overline{Z}_K^{\mathsf{LOO}}\left(\alpha_{\boldsymbol{p}_0}^{\mathsf{S\text{-}SAA}}, \boldsymbol{p}_0\right) \\ &\quad\le 2\sup_{\alpha\ge 0}\left|\overline{Z}_K(\alpha,\boldsymbol{p}_0) - \overline{Z}_K^{\mathsf{LOO}}(\alpha,\boldsymbol{p}_0)\right| \\ &\quad\le 2\sup_{\alpha\ge 0}\left|\overline{Z}_K(\alpha,\boldsymbol{p}_0) - \mathbb{E}\overline{Z}_K(\alpha,\boldsymbol{p}_0)\right| \\ &\quad\quad + 2\sup_{\alpha\ge 0}\left|\overline{Z}_K^{\mathsf{LOO}}(\alpha,\boldsymbol{p}_0) - \mathbb{E}\overline{Z}_K^{\mathsf{LOO}}(\alpha,\boldsymbol{p}_0)\right| \\ &\quad\quad + 2\sup_{\alpha\ge 0}\left|\mathbb{E}\overline{Z}_K(\alpha,\boldsymbol{p}_0) - \mathbb{E}\overline{Z}_K^{\mathsf{LOO}}(\alpha,\boldsymbol{p}_0)\right|.\end{aligned}$$

By Lemma 3.1, the last term is zero, which establishes the first statement. The proof of the second statement is similar, but, in the second inequality, we take an additional supremum over $\boldsymbol{q} \in \mathcal{P}$ to replace $h(\hat{\boldsymbol{m}})$. The proof of the third statement is similar, using $\overline{Z}_K^{\mathsf{LOO}}(\alpha_{h_{\mathcal{P}}}^{\mathsf{S\text{-}SAA}}, h_{\mathcal{P}}(\hat{\boldsymbol{m}})) \le \overline{Z}_K^{\mathsf{LOO}}(\alpha_{\mathcal{P}}^{\mathsf{OR}}, \boldsymbol{q}_{\mathcal{P}}^{\mathsf{OR}})$, and taking a supremum over $\alpha \ge 0$, $\boldsymbol{q} \in \mathcal{P}$ in the second inequality.

Proving a performance guarantee for $\alpha_h^{\mathsf{S\text{-}SAA}}$ thus reduces to bounding the maximal deviations in the lemma. Recall that $\overline{Z}_K(\alpha,\boldsymbol{q}) = \frac{1}{K}\sum_{k=1}^K Z_k(\alpha,\boldsymbol{q})$ and $\overline{Z}_K^{\mathsf{LOO}}(\alpha,\boldsymbol{q}) = \frac{1}{K}\sum_{k=1}^K Z_k^{\mathsf{LOO}}(\alpha,\boldsymbol{q})$. Both processes have a special form: they are the empirical average of K independent stochastic processes (indexed by k). Fortunately, there exist standard tools to bound the maximal deviations of such empirical processes that rely on bounding their metric entropy.

To keep our paper self-contained, we summarize one such approach presented in Pollard (1990), specifically in equation (7.5) of that work. Recall that, for any set $S \subseteq \mathbb{R}^d$, the ϵ-packing number of S, denoted by $D(\epsilon, S)$, is the largest number of elements of S that can be chosen so that the Euclidean distance between any two is at least ϵ. Intuitively, packing numbers describe the size of S at scale ϵ.

Theorem 4.1 (A Maximal Inequality; Pollard 1990)**.** *Let* $\mathbf{W}(t) = (W_1(t), \ldots, W_K(t)) \in \mathbb{R}^K$ *be a stochastic process indexed by* $t \in \mathcal{T}$*, and let* $\overline{W}_K(t) = \frac{1}{K}\sum_{k=1}^K W_k(t)$*. Let* $\mathbf{F} \in \mathbb{R}_+^K$ *be a random variable such that* $|W_k(t)| \le F_k$ *for all* $t \in \mathcal{T}$, $k = 1, \ldots, K$*. Finally, define the random variable*

$$\begin{aligned} J &\equiv J(\{\mathbf{W}(t) : t \in \mathcal{T}\}, \boldsymbol{F}) \\ &\equiv 9\|\mathbf{F}\|_2 \int_0^1 \sqrt{\log D(\|\mathbf{F}\|_2 u, \{\mathbf{W}(t) : t \in \mathcal{T}\})}\,du. \end{aligned} \tag{4.3}$$

Then, for any $p \ge 1$ *and any* $0 < \delta < 1$*, with probability at least* $1 - \delta$*,*[4]

$$\sup_{t\in\mathcal{T}}\left|\overline{W}_K(t) - \mathbb{E}[\overline{W}_K(t)]\right| \le 5^{1/p}\sqrt{p}\|J\|_p K^{-1}\delta^{-1/p}.$$

If $\mathcal{T}$ is finite, then one can bound the maximal deviation with a union bound. Theorem 4.1 extends beyond this simple case to cases where $|\mathcal{T}| = \infty$. The random variable $\boldsymbol{F}$ in the theorem is called an *envelope* for the process $\mathbf{W}(t)$. The random variable J is often called the *Dudley integral*. Whereas packing numbers describe the size of a set at scale ϵ, the Dudley integral roughly describes the size of the set at varying scales. We again refer the reader to Pollard (1990) for discussion.

Our overall proof strategy is to use Theorem 4.1 to bound the two suprema in Lemma 4.1 and thus obtain a bound on the suboptimality. Specifically, define the following stochastic processes:

$$\begin{aligned}\mathbf{Z}(\alpha,\boldsymbol{q}) &= (Z_1(\alpha,\boldsymbol{q}), \ldots, Z_K(\alpha,\boldsymbol{q})), \\ \mathbf{Z}^{\mathsf{LOO}}(\alpha,\boldsymbol{q}) &= \left(Z_1^{\mathsf{LOO}}(\alpha,\boldsymbol{q}), \ldots, Z_K^{\mathsf{LOO}}(\alpha,\boldsymbol{q})\right).\end{aligned}$$

Our proof strategy will be to (1) compute envelopes for both processes; (2) compute the packing numbers and Dudley integrals for the relevant aforementioned sets; (3) apply Theorem 4.1 to bound the relevant maximal deviations; and (4) use these bounds in Lemma 4.1 to bound the suboptimality. We execute this strategy for several special cases in the remainder of the section.

As a first step, we identify envelopes for each process. We restrict attention to the case where the optimal value of each subproblem is bounded for any choice of anchor and shrinkage.

Assumption 4.1 (Bounded Optimal Values)**.** *There exists* C *such that for all* $i = 1, \ldots, d$*, and* $k = 1 \ldots, K$*,* $\sup_{\boldsymbol{q}\in\Delta_d} |c_{ki}(\boldsymbol{x}_k(\infty, \boldsymbol{q}))| \le C$*.*

Notice that $\sup_{\alpha\geq 0,\, \boldsymbol{q}\in\Delta_d}|c_{ki}(\boldsymbol{x}_k(\alpha,\boldsymbol{q}))| = \sup_{\boldsymbol{q}\in\Delta_d}|c_{ki}(\boldsymbol{x}_k(\infty,\boldsymbol{q})|$, so that the assumption bounds the optimal value associated to every policy. Assumption 4.1 is a mild assumption and follows, for example, if $c_{ki}(\cdot)$ is continuous and $\mathcal{X}_k$ is compact. However, the assumption also holds, for example, if $c_{ki}(\cdot)$ is unbounded but coercive. With it, we can easily compute envelopes. Recall that $\hat{N}_{\max} \equiv \max_k \hat{N}_k$.

Lemma 4.2. (Envelopes for $\mathbf{Z}, \mathbf{Z}^{\mathsf{LOO}}$). *Under Assumption* 4.1,

1. *the vector* $\mathbf{F}^{\mathsf{Perf}} \equiv C\boldsymbol{\lambda}/\lambda_{\text{avg}}$ *is an envelope for* $\mathbf{Z}(\alpha,\boldsymbol{q})$ *with* $\|\mathbf{F}^{\mathsf{Perf}}\|_2 = \frac{C}{\lambda_{\text{avg}}}\|\boldsymbol{\lambda}\|_2$;
2. *the random vector* $\mathbf{F}^{\mathsf{LOO}} = C\frac{\hat{\mathbf{N}}}{N\lambda_{\text{avg}}}$ *is an envelope for* $\mathbf{Z}^{\mathsf{LOO}}(\alpha,\boldsymbol{q})$ *with* $\|\mathbf{F}^{\mathsf{LOO}}\|_2 = \frac{C}{N\lambda_{\text{avg}}}\|\hat{\mathbf{N}}\|_2$.

The proof is immediate from the definitions and omitted.

Our next step is to bound the packing numbers (and Dudley integrals) for the sets $\{\mathbf{Z}(\alpha,\boldsymbol{q}) : \alpha \geq \mathbf{0},\ \boldsymbol{q}\in\mathcal{P}\} \subseteq \mathbb{R}^{\mathbf{K}}$, and $\{\mathbf{Z}^{\mathsf{LOO}}(\alpha,\boldsymbol{p}_0) : \alpha\geq \mathbf{0}\}\subseteq\mathbb{R}^K$, for the case of fixed anchors, and the sets $\{\mathbf{Z}(\alpha,\boldsymbol{q}) : \alpha \geq \mathbf{0},\ \boldsymbol{q}\in\mathcal{P}\}\subseteq\mathbb{R}^K$, and $\{\mathbf{Z}^{\mathsf{LOO}}(\alpha,\boldsymbol{q}) : \alpha\geq\mathbf{0},\ \boldsymbol{q}\in\mathcal{P}\}\subseteq\mathbb{R}^K$, for the case of data-driven anchors. Bounding these packing numbers is subtle and requires exploiting the specific structure of the optimization problem (2.2). We *separately* consider two general classes of optimization problems—strongly convex optimization problems and discrete optimization problems—in the remainder of the article. Although we focus on these classes, we expect that a similar proof strategy and technique might be employed to attack other classes of optimization problems.

Remark 4.1. (Performance of $\alpha^{\mathsf{S-SAA}}$ in the Large-Sample Regime). Although we focus on performance guarantees for $\alpha^{\mathsf{S-SAA}}$ in settings where K is large and the expected amount of data per problem is fixed, one could also ask how $\alpha^{\mathsf{S-SAA}}$ performs in the large-sample regime, that is, where K is fixed and $\hat{N}_k\to\infty$ for all k. Using similar techniques, namely, reducing the problem to bounding a certain maximal stochastic deviation, one can show that $\boldsymbol{x}_k(\alpha^{\mathsf{S-SAA}},\boldsymbol{p}_0,\hat{\boldsymbol{m}})$ performs comparably to the full-information solution in Problem (2.2) in this limit. The proof uses somewhat standard arguments for empirical processes. Moreover, the result is perhaps unsurprising; many data-driven methods converge to full-information performance in the large-sample regime (see, e.g., Kleywegt et al. 2002 for the case of SAA) since $\hat{\boldsymbol{p}}_k$ is consistent for $\boldsymbol{p}_k$ for all k in this regime. Consequently, we focus on the small-data, large-scale regime, where Shrunken SAA enjoys strong suboptimality guarantees not enjoyed by SAA. This small-data, large-scale focus, however, causes the N dependence in our bounds to be looser than that obtained from a direct large-sample analysis. Developing a unified analysis of data pooling for *any* sequence of N, K remains an open question. □

4.2. Fixed Anchors and Strongly Convex Optimization Problems

In this section, we treat the case where the K subproblems are smooth enough so that $\boldsymbol{x}_k(\alpha,\boldsymbol{q},\hat{\boldsymbol{m}}_k)$ is smooth in α and $\boldsymbol{q}$ for each k. Specifically, in this section we assume the following.

Assumption 4.2 (Lipschitz, Strongly-Convex Optimization). *There exists L,γ such that the $c_{ki}(\boldsymbol{x})$'s are γ-strongly convex and L-Lipschitz over $\mathcal{X}_k$, and, moreover, $\mathcal{X}_k$ is nonempty and convex, for all $k = 1,\ldots,K$, and $i = 1,\ldots,d$.*

Theorem 4.2 (Shrunken-SAA with Fixed Anchors for Strongly Convex Problems). *Fix any $\boldsymbol{p}_0$. Suppose that Assumptions* 3.1, 4.1, *and* 4.2 *hold, $K\geq 2$, and $N\lambda_{\min}\geq 1$. Then, there exists a universal constant* A *such that, for any $0<\delta<1/2$, with probability at least $1-\delta$, we have that*

$$\begin{aligned}&\mathsf{SubOpt}_{\boldsymbol{p}_0,K}\left(\alpha^{\mathsf{S-SAA}}_{\boldsymbol{p}_0}\right)\\ &\quad\leq \mathrm{A}\cdot\max\left(C, L\sqrt{\frac{C}{\gamma}}\right)\cdot\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{5/4}\\ &\quad\times\frac{\log^2(1/\delta)\cdot\log^{3/2}(K)}{\sqrt{K}}.\end{aligned}$$

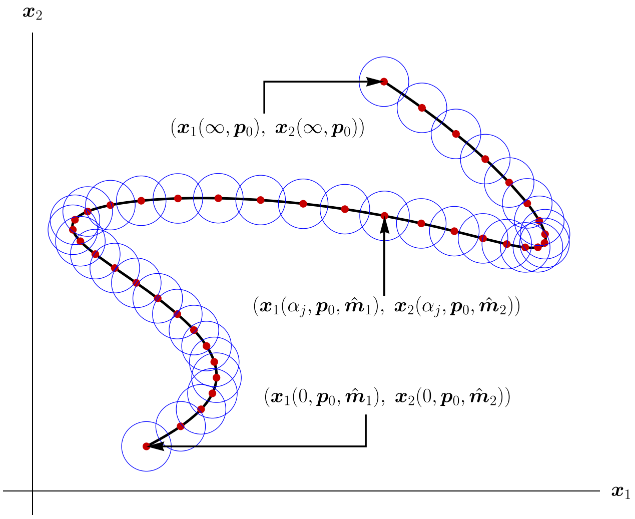
The proof follows our strategy from Section 4.1 (see Section C.1 in Online Appendix C). We sketch the main ideas:

We first bound the packing numbers of $\{\mathbf{Z}(\alpha,\boldsymbol{p}_0) : \alpha\geq 0\}$ and $\left\{\mathbf{Z}^{\mathsf{LOO}}(\alpha,\boldsymbol{p}_0) : \alpha\geq\mathbf{0}\right\}$. The key observation is that, since the subproblems are strongly convex, the optimal solutions $\boldsymbol{x}_k(\alpha,\boldsymbol{p}_0,\hat{\boldsymbol{m}}_k)$ are continuous as functions of α. We utilize this continuity to construct a packing.

Specifically, consider $\left\{\mathbf{Z}(\alpha,\boldsymbol{p}_0) : \alpha\geq\mathbf{0}\right\}$. Continuity in α implies that, by evaluating $\boldsymbol{x}(\alpha,\boldsymbol{p}_0,\hat{\boldsymbol{m}})$ on a sufficiently dense grid of α's, we can construct a covering of $\{(\boldsymbol{x}_k(\alpha,\boldsymbol{p}_0,\hat{\boldsymbol{m}}_k))_{k=1}^K : \alpha\geq 0\}$, which in turn yields a covering of $\left\{\mathbf{Z}(\alpha,\boldsymbol{p}_0) : \alpha\geq\mathbf{0}\right\}$. By carefully choosing the initial grid of α's, we can ensure that this last covering is a valid $(\epsilon/2)$-covering. By (Pollard 1990, p. 10), the size of this covering bounds the ϵ-packing number as desired. Figure 2 illustrates this intuition and further argues that the initial grid of α's should be of size $\mathcal{O}(1/\epsilon^2)$. A similar argument holds for $D(\epsilon,\left\{\mathbf{Z}^{\mathsf{LOO}}(\alpha,\boldsymbol{p}_0) : \alpha\geq\mathbf{0}\right\})$, using a grid of α's to cover $\{(x_k(\alpha,\boldsymbol{p}_0,\hat{\boldsymbol{m}}_k-\boldsymbol{e}_i) : i = 1,\ldots,d,\ k=1,\ldots,K) : \alpha\geq 0\}$. The packing is also of size $\mathcal{O}(1/\epsilon^2)$.

To complete the proof, we use these packing numbers in Theorem 4.1 to bound the maximal deviations

Figure 2. (Color online) Covering a Continuous Process


x_2
$(x_1(\infty, p_0),\ x_2(\infty, p_0))$
$(x_1(\alpha_j, p_0, \hat{m}_1),\ x_2(\alpha_j, p_0, \hat{m}_2))$
$(x_1(0, p_0, \hat{m}_1),\ x_2(0, p_0, \hat{m}_2))$
x_1

Notes. The set $\{(x_k(\alpha, p_0, \hat{m}_k))_{k=1}^K : \alpha \geq 0\}$ can be thought of as a parametric curve indexed by α in the space $\prod_{k=1}^K \mathcal{X}$. Because of the continuity in α (see part iii of Lemma C.1 in Online Appendix C), to cover this curve for any compact set $\alpha \in [0, \alpha_{\max}]$ requires $\mathcal{O}(1/\epsilon)$ balls of size ϵ. Because of the continuity at $\alpha = \infty$ (see part iv of Lemma C.1 in Online Appendix C), it suffices to take $\alpha_{\max} = \mathcal{O}(1/\epsilon)$. This yields a packing number bound of $\mathcal{O}(1/\epsilon^2)$ (see Lemma C.2 in Online Appendix C).

of $\overline{Z}_K(\cdot, p_0), \overline{Z}_K^{\mathrm{LOO}}(\cdot, p_0)$. Substituting into Lemma 4.1 proves Theorem 4.2.

4.3. Data-Driven Anchors and Strongly Convex Problems

We next consider the case of a data-driven anchor $h(\hat{m}) \in \mathcal{P}$. Our performance guarantees will depend on the complexity of $\mathcal{P}$ as measured by the size of its ℓ_1-packing numbers. Namely, we let $D_1(\epsilon, \mathcal{P})$ be the largest number of elements of $\mathcal{P}$ that can be chosen so that the ℓ_1-distance between any two is at least ϵ.[5] Then, we have the following.

Theorem 4.3 (Shrunken-SAA with Data-Driven Anchors for Strongly Convex Problems)**.** *Suppose that Assumptions 3.1, 4.1, and 4.2 hold, $K \geq 2$. Let $d_0 \geq 1$ be such that for any $0 < \epsilon < 1/2$, $\log D_1(\epsilon, \mathcal{P}) \leq d_0 \log(1/\epsilon)$. Then, there exists a universal constant A such that for any $0 < \delta < 1/2$, with probability at least $1 - \delta$, we have that*

$$\mathrm{SubOpt}_{h,K}(\alpha_h^{\mathrm{S\text{-}SAA}}) \leq \mathrm{A} \cdot \max\left(C, \frac{L^2}{\gamma} + L\sqrt{\frac{C}{\gamma}}\right)\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{5/4} \times \frac{d_0^2 \log^{7/2}(K) \log^2(1/\delta)}{\sqrt{K}}.$$

In the special case of $h_{\mathcal{P}}(\cdot)$, we can prove an even stronger result, namely, that Shrunken-SAA with $h_{\mathcal{P}}$ performs comparably to pooling in an optimal way to the best anchor within the class $\mathcal{P}$.

Theorem 4.4. (Shrunken-SAA with $h_{\mathcal{P}}$ for Strongly Convex Problems). *Under the assumptions of Theorem 4.3, there exists a universal constant A such that for any $0 < \delta < 1/2$, with probability at least $1 - \delta$, we have that*

$$\mathrm{SubOpt}_{\mathcal{P},K}\left(\alpha_{h_{\mathcal{P}}}^{\mathrm{S\text{-}SAA}}, h_{\mathcal{P}}(\hat{m})\right) \leq \mathrm{A} \cdot \max\left(C, \frac{L^2}{\gamma} + L\sqrt{\frac{C}{\gamma}}\right)\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{5/4} \times \frac{d_0^2 \log^{7/2}(K) \log^2(1/\delta)}{\sqrt{K}}.$$

In both theorems, the constant d_0 measures the complexity of $\mathcal{P}$. Without loss of generality, $d_0 \leq 3d$ since $\mathcal{P} \subseteq \Delta_d$ and $\log D_1(\epsilon, \Delta_d) \leq 3d \log(1/\epsilon)$ (Pollard 1990, lemma 4.1). In practice, we might choose flexible, parametric families for $\mathcal{P}$ with small d_0 that do not scale with d. An example might be when $\mathcal{P}$ consists of all (truncated) Poisson distributions with mean at most Λ, in which case, one can take $d_0 = 2\max(1, \log(\Lambda))$, *independently* of d (and the truncation). Another example is given in Section 6 using beta distributions. In general, we expect that our performance bounds must depend on the complexity of $\mathcal{P}$ in some way, because we impose no assumptions on the function $h(\hat{m})$ that selects the anchor and, hence, must control behavior across all of $\mathcal{P}$.

Both proofs follow the strategy of Section 4.1 (see Section C.2 of Online Appendix C). The key idea to bounding the packing numbers is again to leverage continuity and cover the set $\{(\alpha, q) : \alpha \geq 0, q \in \mathcal{P}\}$. Since both proofs leverage Lemma 4.1, the right-hand sides of the bounds are the same.

By contrast, the left-hand sides of Theorems 4.3 and 4.4 are different: the first measures suboptimality relative to an oracle with a prespecified anchor, whereas the second is relative to an oracle that can optimize the choice of anchor. This distinction mirrors the difference between "estimate-then-optimize" procedures and those which choose parameters in an optimization-aware fashion. Continuing our example where $\mathcal{P}$ is a set of Poisson distributions, Theorem 4.3 bounds the suboptimality of Shrunken-SAA when using (all) the data to fit a Poisson distribution without regard to the downstream optimization, for example, by maximum likelihood, and then choosing α and $x_k(\cdot)$ to optimize. By contrast, Theorem 4.4 bounds the performance of Shrunken-SAA when choosing the anchor, α and $x_k(\cdot)$ simultaneously to optimize the downstream optimization.

4.4. Fixed Anchors and Discrete Optimization Problems

In this section, we consider the case where the K subproblems are discrete optimization problems. Specifically, we require $|\mathcal{X}_k| < \infty$ for each $k = 1, \dots, K$. This encompasses, for example, binary linear or nonlinear optimization and linear optimization over a polytope, since we may restrict to its vertices.

Unlike the case of strongly convex problems, the optimization defining $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ (see Equation (2.4)) may admit multiple optima, and, hence, $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ requires a tie-breaking rule. For our results, we assume that this tie-breaking rule is consistent in the sense that if the set of minimizers to Equation (2.4) is the same for two distinct values of $(\alpha, \boldsymbol{p}_0)$, then the tie-breaking minimizer is also the same for both. We express this requirement by representing the tie-breaking rule as a function from a set of minimizers to a chosen minimizer.

Assumption 4.3 (Consistent Tie-Breaking). *For each k, there exists $\sigma_k : 2^{\mathcal{X}_k} \to \mathcal{X}_k$ such that*

$$\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k) = \sigma_k\left(\arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} \ \hat{\boldsymbol{p}}_k(\alpha)^\top \boldsymbol{c}_k(\boldsymbol{x}_k)\right).$$

Then we have the following.

Theorem 4.5 (Shrunken-SAA with Fixed Anchors for Discrete Problems). *Suppose that $|\mathcal{X}_k| < \infty$ for each k, $K \geq 2$, and that Assumptions 3.1, 4.1, and 4.3 hold. Then, there exists a universal constant* A *such that for any $0 < \delta < 1/2$ we have that, with probability at least $1 - \delta$,*

$$\begin{aligned} &\mathrm{SubOpt}_{\boldsymbol{p}_0,K}\left(\alpha^{\mathrm{S-SAA}}_{\boldsymbol{p}_0}\right) \\ &\quad \leq \mathrm{A} \cdot C\frac{\lambda_{\max}}{\lambda_{\min}} \cdot \sqrt{\log\left(2N_{\max}\sum_{k=1}^{K}|\mathcal{X}_k|\right)} \cdot \\ &\quad \times \frac{\log^{3/2}(K) \cdot \log^{3/2}(1/\delta)}{\sqrt{K}}. \end{aligned}$$

We stress that $|\mathcal{X}_k|$ occurs logarithmically in the bound, so that the bound is reasonably tight, even when the number of feasible solutions per subproblem may be large. For example, consider binary optimization. Then, $|\mathcal{X}_k|$ often scales exponentially in the number of binary variables, so that $\log(|\mathcal{X}_k|)$ scales like the number of binary variables. Thus, as long as the number of binary variables per subproblem is much smaller than K, the suboptimality will be small with high probability.

We also note that, unlike Theorem 4.2, the aforementioned bound depends on $\log(N_{\max})$. This mild dependence stems from the fact that we have made *no assumptions of continuity* on the functions $\boldsymbol{c}_k(\boldsymbol{x}, \boldsymbol{\xi})$ in $\boldsymbol{x}$ or $\boldsymbol{\xi}$. Since these functions could be arbitrarily nonsmooth, we need to control their behavior separately across all of the LOO iterations, which introduces the $N_{\max}$ dependence. With stronger assumptions, it might be possible to remove this dependence. However, since we are mostly interested in the setting where N_k is moderate to small for all k, we do not pursue this idea.

To prove Theorem 4.5, we again follow the approach outlined in Section 4.1. However, since the policy $\boldsymbol{x}(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ need not be smooth in α, we adopt a different strategy than in Section 4.2. Specifically, we bound the cardinality of $\{\mathbf{Z}(\alpha, \boldsymbol{p}_0) : \alpha \geq \mathbf{0}\}$, $\{\mathbf{Z}^{\mathrm{LOO}}(\alpha, \boldsymbol{p}_0) : \alpha \geq \mathbf{0}\}$, directly. (Recall that the cardinality of a set bounds its ϵ-packing number for any ϵ.)

First note that the cardinality of $\{\mathbf{Z}(\alpha, \boldsymbol{p}_0) : \alpha \geq \mathbf{0}\}$ is at most that of $\{(\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k))_{k=1}^K : \alpha \geq 0\}$. A trivial bound on this latter set's cardinality is $\prod_{k=1}^K |\mathcal{X}_k|$. This bound is too crude for our purposes; it grows exponentially in K, even if $|\mathcal{X}_k|$ is bounded for all k. Intuitively, this bound is crude, because it supposes that we can vary each solution $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ independently of the others to achieve all $\prod_{k=1}^K |\mathcal{X}_k|$ possible combinations. In reality, we can only vary a single parameter, α, that simultaneously controls all K solutions, rather than varying them separately. We use this intuition to show that a much smaller bound, that is, $2\sum_{k=1}^K |\mathcal{X}_k|$, is valid.

To this end, we fix k and study the dependence of $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ on α. In the trivial case $\hat{N}_k = 0$, $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ takes only one value: $\boldsymbol{x}_k(\infty, \boldsymbol{p}_0)$. Hence, we focus on the case $\hat{N}_k \geq 1$.

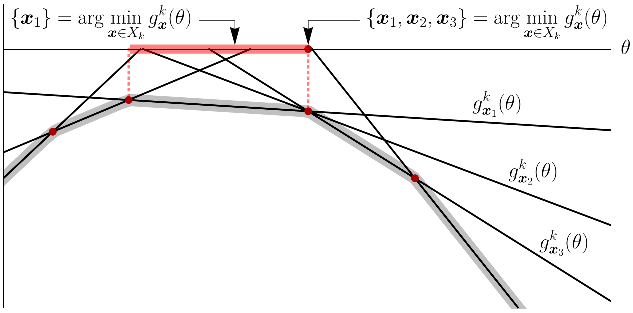
Consider reparameterizing the solution in terms of $\theta = \frac{\alpha}{\alpha + \hat{N}_k} \in [0, 1)$, and let $\alpha(\theta) = \frac{\theta}{1-\theta}\hat{N}_k$. Then for any $\boldsymbol{x} \in \mathcal{X}_k$, define the linear function

$$g_{k\boldsymbol{x}}(\theta) = \left((1-\theta)\hat{\boldsymbol{p}}_k + \theta \boldsymbol{p}^0\right)^\top \boldsymbol{c}_k(\boldsymbol{x}), \qquad \theta \in [0, 1).$$

Since $g_{k\boldsymbol{x}}(\cdot)$ is linear, the function $\theta \mapsto \min_{\boldsymbol{x} \in \mathcal{X}_k} g_{k\boldsymbol{x}}(\theta)$ is concave and piecewise linear with at most $|\mathcal{X}_k|$ -1 break points. By construction, $\boldsymbol{x}_k(\alpha(\theta), \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k) \in \arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} g_{k\boldsymbol{x}}(\theta)$. More precisely, for any θ, the set of active supporting hyperplanes of $\min_{\boldsymbol{x} \in \mathcal{X}_k} g_{k\boldsymbol{x}}(\cdot)$ at θ is $\{(\boldsymbol{p}^0 - \hat{\boldsymbol{p}}_k)^\top \boldsymbol{c}_k(\boldsymbol{x}) \ \boldsymbol{x} \in \arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} g_{k\boldsymbol{x}}(\theta)\}$.

Since the set of active supporting hyperplanes is constant between break points, the set of minimizers $\arg\min_{\boldsymbol{x}_k \in \mathcal{X}_k} g_{k\boldsymbol{x}}(\theta)$ is also constant between break points. By Assumption 4.3, this implies that $\theta \mapsto \boldsymbol{x}_k(\alpha(\theta), \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ is piecewise constant with at most $|\mathcal{X}_k| - 1$ points of discontinuity (see also Figure 3). Viewed in the original parameterization in terms of α, it follows that $\alpha \mapsto \boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ is also piecewise constant with

Figure 3. (Color online) Counting Discrete Solutions


$\{\boldsymbol{x}_1\} = \arg\min_{\boldsymbol{x}\in X_k} g^k_{\boldsymbol{x}}(\theta)$
$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\} = \arg\min_{\boldsymbol{x}\in X_k} g^k_{\boldsymbol{x}}(\theta)$
θ
$g^k_{\boldsymbol{x}_1}(\theta)$
$g^k_{\boldsymbol{x}_2}(\theta)$
$g^k_{\boldsymbol{x}_3}(\theta)$

Notes. A concave piecewise-linear function consisting of $|\mathcal{X}_k|$ lines has at most $|\mathcal{X}_k| - 1$ break points, between which the set of active supporting lines is constant. Any function of this set of active supporting lines is piecewise constant with at most $|\mathcal{X}_k| - 1$ discontinuities.

at most $|\mathcal{X}_k| - 1$ points of discontinuity. Thus we have the following.

Lemma 4.3. *Suppose that Assumption 4.3 holds. Fix any $\boldsymbol{p}_0$ and $\hat{\boldsymbol{m}}_k$. Then, the function $\alpha \mapsto \boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ is piecewise constant with at most $|\mathcal{X}_k| - 1$ points of discontinuity.*

Taking the union of all these points of discontinuity over k proves that $\left(\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)\right)_{k=1}^K$ is also piecewise constant with at most $\sum_{k=1}^K (|\mathcal{X}_k| - 1)$ points of discontinuity. Therefore, it takes at most $2\sum_{k=1}^K |\mathcal{X}_k| - 2K + 1$ different values—a distinct value for each of the $\sum_{k=1}^K (|\mathcal{X}_k| - 1)$ break points plus a distinct value for the $\sum_{k=1}^K (|\mathcal{X}_k| - 1) + 1$ regions between break points. This gives the desired cardinality bound on $|\{\mathbf{Z}(\alpha, \boldsymbol{p}_0) : \alpha \geq \mathbf{0}\}|$. A similar argument considering the larger $\left(\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k - \boldsymbol{e}_i)\right)_{i\in\mathcal{I}_k, k=1,\ldots,K}$, where $\mathcal{I}_k = \{i = 1, \ldots, d : \hat{m}_{ki} > 0\}$, gives a corresponding cardinality bound on $|\{\mathbf{Z}^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0) : \alpha \geq \mathbf{0}\}|$. Noting $|\mathcal{I}_k| \leq \min(d, \hat{N}_k)$ gives the following (proof omitted):

Corollary 4.1 (Size of Discrete Solutions Sets)**.** *Suppose that Assumption 4.3 holds. Then,*

$$\left|\{\mathbf{Z}(\alpha, \boldsymbol{p}_0) : \alpha \geq \mathbf{0}\}\right| \leq 2\sum_{k=1}^K |\mathcal{X}_k|,$$

$$\left|\{\mathbf{Z}^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0) : \alpha \geq \mathbf{0}\}\right| \leq 1 + 2\sum_{k=1}^K \min\left(d, \hat{N}_k\right)|\mathcal{X}_k|.$$

The additional "1" in the case of $\left|\{\mathbf{Z}^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0) : \alpha \geq \mathbf{0}\}\right|$ covers the case where $\hat{N}_{\max} = 0$ and $\{\mathbf{Z}^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0) : \alpha \geq 0\} = \{\mathbf{0}\}$. Although these bounds may appear large, an important feature is that they are only linear in K as long as the $|\mathcal{X}_k|$'s are bounded over k.

We use these cardinality bounds to bound the packing numbers and then apply our usual strategy via Theorem 4.1 and Lemma 4.1 to prove Theorem 4.5. The details are in Section C.3 of Online Appendix C.

4.5. Data-Driven Anchors and Discrete Optimization Problems

We next extend the results of Section 4.4 to the case of a data-driven anchor, $h(\hat{\boldsymbol{m}})$. As in Section 4.3, our bounds will depend on a measure of complexity of $\mathcal{P}$, namely, the dimension of $\mathrm{spn}(\mathcal{P}) \equiv \{\sum_{\ell=1}^d \theta_\ell \boldsymbol{q}_\ell \theta_\ell \in \mathbb{R}, \boldsymbol{q}_\ell \in \mathcal{P}, \ell = 1, \ldots, d\}$ when viewed as a linear subspace. Denote this dimension by d_0, and note $1 \leq d_0 \leq d$. A canonical example might be when $\mathcal{P}$ consists of mixture distributions with d_0 (specified) components. We prove the following.

Theorem 4.6 (Shrunken-SAA with Data-Driven Anchors for Discrete Problems)**.** *Suppose that $|\mathcal{X}_k| < \infty$ for each k, that span($\mathcal{P}$) has dimension d_0, and that Assumptions 3.1, 4.1, and 4.3 hold. Then, there exists a universal constant A such that, for all $0 < \delta < 1/2$, we have that, with probability at least $1 - \delta$,*

$$\mathsf{SubOpt}_{h,K}(\alpha_h^{\mathsf{S\text{-}SAA}}) \leq \mathrm{A} \cdot C\frac{\lambda_{\max}}{\lambda_{\min}}\sqrt{d_0 \log\left(N_{\max}\sum_{k=1}^K |\mathcal{X}_k|\right)} \cdot \frac{\log^{3/2}(K)\log^2(1/\delta)}{\sqrt{K}}.$$

Theorem 4.7 (Shrunken-SAA with $h_{\mathcal{P}}$ for Discrete Problems)**.** *Under the assumptions of Theorem 4.6, there exists a universal constant A such that, for any $0 < \delta < 1/2$ with probability at least $1 - \delta$, we have that*

$$\mathsf{SubOpt}_{\mathcal{P},K}\left(\alpha_{h_{\mathcal{P}}}^{\mathsf{S\text{-}SAA}}, h_{\mathcal{P}}(\hat{\boldsymbol{m}})\right) \leq \mathrm{A} \cdot C\frac{\lambda_{\max}}{\lambda_{\min}}\sqrt{d_0 \log\left(N_{\max}\sum_{k=1}^K |\mathcal{X}_k|\right)} \cdot \frac{\log^{3/2}(K)\log^2(1/\delta)}{\sqrt{K}}.$$

Both proofs follow the strategy from Section 4.1 (see Section C.4 of Online Appendix C) and, hence, lead to the same right-hand sides. However, the left-hand sides are distinct. We sketch the main ideas of the proof:

We first bound $\left|\{\mathbf{Z}(\alpha, \boldsymbol{q}) : \alpha \geq \mathbf{0}, \boldsymbol{q} \in \mathcal{P}\}\right|, |\{\mathbf{Z}^{\mathsf{LOO}}(\alpha, \boldsymbol{q}) : \alpha \geq 0, \boldsymbol{q} \in \mathcal{P}\}|$. The key is to generalize the argument of Section 4.4 from counting break points in a univariate piecewise-affine function to counting the pieces in a multivariate piecewise affine function.

First, we reparameterize our policies. Let the columns of $\boldsymbol{V} \in \mathbb{R}^{d\times d_0}$ be a basis of spn($\mathcal{P}$). Then, interpreting $\mathbf{0}/\mathbf{0}$ as $\boldsymbol{e}/d \in \Delta_d$,

$$\begin{aligned}
&\left|\left\{\mathbf{Z}(\alpha,\boldsymbol{q}) : \alpha \geq \mathbf{0}, \boldsymbol{q} \in \mathcal{P}\right\}\right| \\
&\leq \left|\left\{\left(\boldsymbol{x}_k(\alpha,\boldsymbol{q},\hat{\boldsymbol{m}}_k)\right)_{k=1}^K : \boldsymbol{q} \in \mathcal{P}, \alpha \geq 0\right\}\right| \\
&\leq \left|\left\{(\boldsymbol{x}_k(\|\boldsymbol{w}\|_1, \boldsymbol{w}/\|\boldsymbol{w}\|_1, \hat{\boldsymbol{m}}_k))_{k=1}^K : \right.\right. \\
&\qquad \left.\left. \boldsymbol{w} \in \operatorname{span}(\mathcal{P}) \cap \mathbb{R}_+^d\right\}\right| \qquad (4.4)\\
&= \left|\left\{(\boldsymbol{x}_k(\|\boldsymbol{V\theta}\|_1, \boldsymbol{V\theta}/\|\boldsymbol{V\theta}\|_1, \hat{\boldsymbol{m}}_k))_{k=1}^K : \right.\right. \\
&\qquad \left.\left. \boldsymbol{\theta} \in \mathbb{R}^{d_0}, \boldsymbol{V\theta} \in \mathbb{R}_+^d\right\}\right|.
\end{aligned}$$

Hence, it suffices to bound the right-most side of Equation (4.4). An advantage of this $\boldsymbol{\theta}$-parameterization over the original $(\alpha, \boldsymbol{q})$-parameterization is that, for $\hat{N}_k > 0$,

$$\boldsymbol{x}_k(\|\boldsymbol{V\theta}\|_1, \boldsymbol{V\theta}/\|\boldsymbol{V\theta}\|_1, \hat{\boldsymbol{m}}_k) \in \arg\min_{\boldsymbol{x}\in\mathcal{X}_k} (\boldsymbol{V\theta}+\hat{\boldsymbol{m}}_k)^\top \boldsymbol{c}_k(\boldsymbol{x}), \qquad (4.5)$$

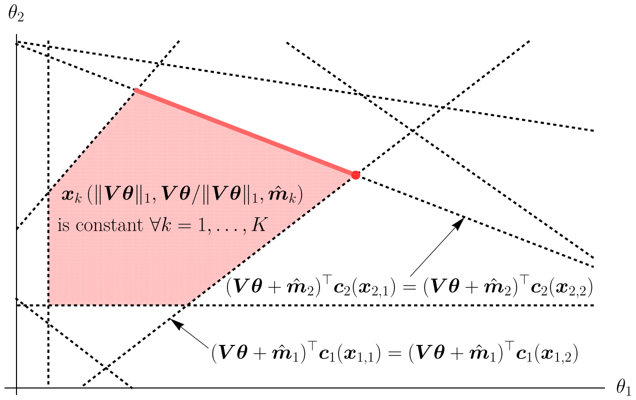
and $\boldsymbol{\theta}$ occurs linearly in this representation.

The set of $\boldsymbol{\theta}$ where we are indifferent between $\boldsymbol{x}_{ki}, \boldsymbol{x}_{kj} \in \mathcal{X}_k$ in Equation (4.5) is the hyperplane

$$H_{kij} = \left\{\boldsymbol{\theta} \in \mathbb{R}^{d_0} : (\boldsymbol{V\theta} + \hat{\boldsymbol{m}}_k)^\top \left(\boldsymbol{c}_k(\boldsymbol{x}_{ki}) - \boldsymbol{c}_k(\boldsymbol{x}_{kj})\right) = \mathbf{0}\right\}. \qquad (4.6)$$

Consider drawing all $\sum_{k=1}^K \binom{|\mathcal{X}_k|}{2}$ such hyperplanes, as in Figure 4. Then, for any $\boldsymbol{\theta} \in \mathbb{R}^{d_0}$, consider the polyhedron given by the equality constraints of those hyperplanes containing $\boldsymbol{\theta}$ and the inequality constraints defined by the side on which $\boldsymbol{\theta}$ lies for the remaining hyperplanes. The relative ordering of $\{(\boldsymbol{V\theta} + \hat{\boldsymbol{m}}_k)^\top \boldsymbol{c}_k(\boldsymbol{x}_k) : \boldsymbol{x}_k \in \mathcal{X}_k\}$ is constant for all $\boldsymbol{\theta}$ in this polyhedron's interior. Hence, $(\boldsymbol{x}_k(\|\boldsymbol{V\theta}\|_1, \boldsymbol{V\theta}/\|\boldsymbol{V\theta}\|_1, \hat{\boldsymbol{m}}_k))_{k=1}^K$ is also constant. Thus, to bound $\{\mathbf{Z}(\alpha, \boldsymbol{q}) : \alpha \geq 0, \boldsymbol{q} \in \mathcal{P}\}$, it suffices to count the number of such polyhedra. We do this counting in Section C.4 of Online Appendix C. A similar argument (with a different hyperplane arrangement) can be used to bound the cardinality of $\left\{\mathbf{Z}^{\mathsf{LOO}}(\alpha, \boldsymbol{q}) : \alpha \geq \mathbf{0}, \boldsymbol{q} \in \mathcal{P}\right\}$. We summarize the results as follows.

Figure 4. (Color online) Solution-Induced Hyperplane Arrangement


θ_2
$\boldsymbol{x}_k\left(\|\boldsymbol{V\theta}\|_1, \boldsymbol{V\theta}/\|\boldsymbol{V\theta}\|_1, \hat{\boldsymbol{m}}_k\right)$
is constant $\forall k = 1, \ldots, K$
$(\boldsymbol{V\theta} + \hat{\boldsymbol{m}}_2)^\top \boldsymbol{c}_2(\boldsymbol{x}_{2,1}) = (\boldsymbol{V\theta} + \hat{\boldsymbol{m}}_2)^\top \boldsymbol{c}_2(\boldsymbol{x}_{2,2})$
$(\boldsymbol{V\theta} + \hat{\boldsymbol{m}}_1)^\top \boldsymbol{c}_1(\boldsymbol{x}_{1,1}) = (\boldsymbol{V\theta} + \hat{\boldsymbol{m}}_1)^\top \boldsymbol{c}_1(\boldsymbol{x}_{1,2})$
θ_1

Notes. The hyperplanes H_{kij} (see Section 4.6) in $\mathbb{R}^d$ are indifference curves between solutions $\boldsymbol{x}_{ki}$ and $\boldsymbol{x}_{kj}$ in Section 4.5. The total ordering on each set $\mathcal{X}$ induced by the objective of Section 4.5 is thus constant on the interior of the fully specified polyhedra defined by the hyperplanes.

Lemma 4.4 (Size of Discrete Solutions Sets)**.** *Under the assumptions of* Theorem 4.6*,*

$$\left|\left\{\mathbf{Z}(\alpha, \boldsymbol{q}) : \alpha \geq \mathbf{0}, \boldsymbol{q} \in \mathcal{P}\right\}\right| \leq \left(\sum_{k=1}^K |\mathcal{X}_k|^2\right)^{d_0},$$

$$\left|\left\{\mathbf{Z}^{\mathsf{LOO}}(\alpha, \boldsymbol{q}) : \alpha \geq \mathbf{0}, \boldsymbol{q} \in \mathcal{P}\right\}\right| \leq 1 + \hat{N}_{\max}^{d_0} \left(\sum_{k=1}^K |\mathcal{X}_k|^2\right)^{d_0}.$$

Importantly, both bounds are polynomial in K if the $|\mathcal{X}_k|$'s are bounded over k. We then apply Theorem 4.1 to bound the maximal deviations in Lemma 4.1, proving the theorems.

4.6. Performance Guarantees for Continuous Distributions

None of the bounds in Theorems 4.2–4.7 or Algorithm 1 depend on d, the size of the support of $\boldsymbol{p}_k$, suggesting that similar guarantees might hold for continuous distributions. In Online Appendix F, we prove such results for strongly convex optimization problems by discretizing and taking a limit as the granularity tends to zero. The details are straightforward. Unfortunately, for discrete optimization problems, it is not clear that similar results hold without additional assumptions. Again, see Online Appendix F.

5. The Sub-Optimality-Instability Trade-Off: An Intuition for Data Pooling

Shrunken SAA also provides intuition into *when* and *why* data pooling improves upon decoupling. For simplicity, first consider a non-data-driven anchor $\boldsymbol{p}_0$. Lemma 3.1 shows that $\mathbb{E}\left[\overline{Z}_K(\alpha, \boldsymbol{p}_0)\right] = \mathbb{E}[\overline{Z}_K^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0)]$. Theorems 4.2 and 4.5 further establish that, under mild conditions,

$$\begin{aligned}
&\sup_{\alpha\geq 0} \Big| \underbrace{\overline{Z}_K(\alpha, \boldsymbol{p}_0)}_{\text{True Performance of } \alpha} - \underbrace{\overline{Z}_K^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0)}_{\text{LOO Performance of } \alpha} \Big| \\
&\quad \leq \underbrace{\tilde{\mathcal{O}}_p\left(1/\sqrt{K}\right)}_{\text{Stochastic Error}}.
\end{aligned}$$

Hence, optimizing $\overline{Z}_K(\alpha, \boldsymbol{p}_0)$ over α is roughly equivalent to optimizing $\overline{Z}_K^{\mathsf{LOO}}(\alpha, \boldsymbol{p}_0)$ for large K.

A simple algebraic manipulation then shows that

$$\overline{Z}_K^{\mathrm{LOO}}(\alpha, \boldsymbol{p}_0) = \frac{1}{N\lambda_{\mathrm{avg}}}(\text{SAA-SubOpt}(\alpha) + \text{Instability}(\alpha) + \text{SAA}(0)),$$

where

$$\text{SAA-SubOpt}(\alpha) \equiv \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{d} \hat{m}_{ki}(c_{ki}(x_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)) - c_{ki}(x_k(0, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)))$$

$$\text{Instability}(\alpha) \equiv \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{d} \hat{m}_{ki}(c_{ki}(x_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k - \boldsymbol{e}_i)) - c_{ki}(x_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k))),$$

$$\text{SAA}(0) \equiv \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{d} \hat{m}_{ki} c_{ki}(x_k(0, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)),$$

and SAA(0) does not depend on α. Hence, optimizing $\overline{Z}_K^{\mathrm{LOO}}(\alpha, \boldsymbol{p}_0)$ is roughly equivalent to

$$\min_{\alpha \geq 0} \quad \text{SAA-SubOpt}(\alpha) + \text{Instability}(\alpha).$$

We term this last optimization the "Sub-Optimality-Instability Trade-Off."

For intuition, SAA-SubOpt(α) is nonnegative and measures the average degree to which each $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ is suboptimal with respect to a (scaled) SAA objective. It is minimized at $\alpha = 0$, and we expect it is increasing in α. By contrast, Instability(α) measures the average degree to which the (scaled) performance of $\boldsymbol{x}_k(\alpha, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ changes in-sample if we use one fewer data point. It is minimized at $\alpha = \infty$, since $\boldsymbol{x}_k(\infty, \boldsymbol{p}_0, \hat{\boldsymbol{m}}_k)$ does not depend on the data at all. One might expect Instability(α) to be decreasing. However, although Instability(α) is often decreasing for large α, its behavior for small α can be subtle. In sum, intuitively, Shrunken-SAA improves performance by seeking an α in the "sweet spot" that balances this fundamental trade-off.

This trade-off also illuminates *when* data pooling offers an improvement. Intuitively, $\alpha^{\mathrm{S-SAA}} > 0$ only if Instability(0) is large and decreasing fast enough that the (in-sample) suboptimality incurred by choosing a small positive α is outweighed by the increased stability. Thus, we intuit that data pooling offers a benefit whenever (i) the SAA solution is unstable, (ii) the fully shrunken solution $\boldsymbol{x}_k(\infty, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ is not too suboptimal, and (iii) K is large. In particular, when $\hat{N}_k$ is small for most k, the SAA solution is likely to be *very* unstable. Hence, intuition suggests that data pooling provides a benefit whenever $\hat{N}_k$ is small but K is large, that is, the small-data, large-scale regime.

The intuition for a data-driven anchor $h(\hat{\boldsymbol{m}})$ is similar. The proofs of Theorems 4.3 and 4.6 show that the approximation $\overline{Z}_K(\alpha, \boldsymbol{p}_0) \approx \overline{Z}_K^{\mathrm{LOO}}(\alpha, \boldsymbol{p}_0)$ holds uniformly in α *and* $\boldsymbol{p}_0$. Thus, the Sub-Optimality-Instability Trade-Off also holds for the data-driven anchor $h(\hat{\boldsymbol{m}})$.

The Sub-Optimality-Instability Trade-Off resembles the bias-variance trade-off; however, it applies to general optimization problems, not just MSE. Even in the case of MSE, these trade-offs still exhibit differences (see Section D.2 of Online Appendix D).

Finally, for simple problems, we may analytically study the benefits of data pooling (see Theorem 2.1), but, for more complex problems, such a study is not possible. Fortunately, both SAA-SubOpt(α) and Instability(α) can be evaluated *directly from the data*. Thus, evaluating the Sub-Optimality-Instability Trade-Off for various α for a particular instance explains why (or why not) data pooling improves performance for that instance. We illustrate this idea in Section D.1 of Online Appendix D.

6. Computational Experiments

We next study the empirical performance of Shrunken-SAA on synthetic and real data. All code is available at https://github.com/vgupta1/JS_SAA. Our focus is assessing the degree to which Shrunken-SAA is robust to violations of the assumptions underlying Theorems 4.2–4.7. Specifically, we ask how Shrunken-SAA performs when (i) K is small to moderate; (ii) each $\hat{N}_k$ is fixed and nonrandom; (iii) the true $\mathbb{P}_k$ do not have finite, discrete support; or (iv) N grows large.

Each subproblem is a newsvendor problem with critical fractile $s = 95\%$. We use real sales data from a chain of European pharmacies to specify the distributions $\boldsymbol{p}_k$ (see Section E.3 in Online Appendix E).

We compare 10 policies: The SAA and KS polices are our decoupled benchmarks. Recall that, for the newsvendor problem, SAA is also the optimal solution to a distributionally robust formulation using a Wasserstein ambiguity set (Esfahani and Kuhn 2018). We define KS to be an optimal solution to a distributionally robust formulation of the newsvendor problem using the Kolmogorov-Smirnov ambiguity set (see Section E.2 in Online Appendix E).

The policies JS-Fixed, S-SAA-Fixed, and Oracle-Fixed each shrink toward the uniform distribution. JS-Fixed, $\boldsymbol{x}(\alpha_{\boldsymbol{p}_0}^{\mathrm{JS}}, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$, pools according to Theorem 2.1; S-SAA-Fixed, $\boldsymbol{x}(\alpha_{\boldsymbol{p}_0}^{\mathrm{S-SAA}}, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$, is our Shrunken-SAA algorithm; and Oracle-Fixed, $\boldsymbol{x}(\alpha_{\boldsymbol{p}_0}^{\mathrm{OR}}, \boldsymbol{p}_0, \hat{\boldsymbol{m}})$ is the oracle shrinkage.

The policies S-SAA-Beta and Oracle-Beta each shrink toward a data-driven anchor in $\mathcal{P}$, a set of scaled beta distributions (see Section E.3 of Online Appendix E). S-SAA, $\boldsymbol{x}(\alpha_{\mathcal{P}}^{\mathrm{S-SAA}}, h_{\mathcal{P}}(\hat{\boldsymbol{m}}), \hat{\boldsymbol{m}})$, uses $h_{\mathcal{P}}$, and Oracle-Beta, $\boldsymbol{x}(\alpha_{\mathcal{P}}^{\mathrm{OR}}, \boldsymbol{q}_{\mathcal{P}}^{\mathrm{OR}}, \hat{\boldsymbol{m}})$, uses the oracle anchor.

Finally, the policies JS-GM, S-SAA-GM, and Oracle-GM each shrink toward the grand-mean anchor, $\hat{\boldsymbol{p}}^{\mathrm{GM}}$. JS-GM pools according to Theorem 2.1, S-SAA-GM is our Shrunken-SAA Algorithm, and Oracle-GM is the oracle pooling.

Contrasting the JS policies and the decoupled policies illustrates the value of data pooling in a "generic" fashion. Contrasting the Shrunken-SAA policies and JS policies quantifies the additional benefit of data pooling in an optimization-aware fashion. Contrasting the "Beta" and "Fixed" anchor versions quantifies the value of a good anchor.

Before presenting details, we summarize our main findings. When N is moderate to large, all methods perform comparably to the full-information solution. When N is small to moderate, however, our Shrunken-SAA policies provide a significant benefit over SAA and a substantial benefit over JS variants that do not leverage the optimization structure. This is true even for moderate K ($K \leq 100$) and even when the $\hat{N}_k$'s are fixed (violating Assumption 3.1). The value of d has little effect on the performance of Shrunken-SAA; it strongly outperforms decoupling, even as $d \to \infty$. Finally, our GM heuristic has very strong performance, comparable to the Beta variants that optimize the choice of anchor, at a much smaller computational cost.

We present all results as "% Benefit over SAA"; that is, bigger values are better.

6.1. An Idealized Synthetic Data Set

We first consider an ideal setting. Specifically, after discretizing demand for each store into $d = 20$ buckets, we set $\boldsymbol{p}_k$ to be the (store-level) empirical distribution of demand. We then simulate synthetic data according to Section 2.1 under Assumption 3.1. We train each policy and then evaluate its true performance using the $\boldsymbol{p}_k$. We repeat this process 200 times. The left panel of Figure 5 shows the average results for a subset of the policies (see Table EC.1 in Online Appendix E for all policies).

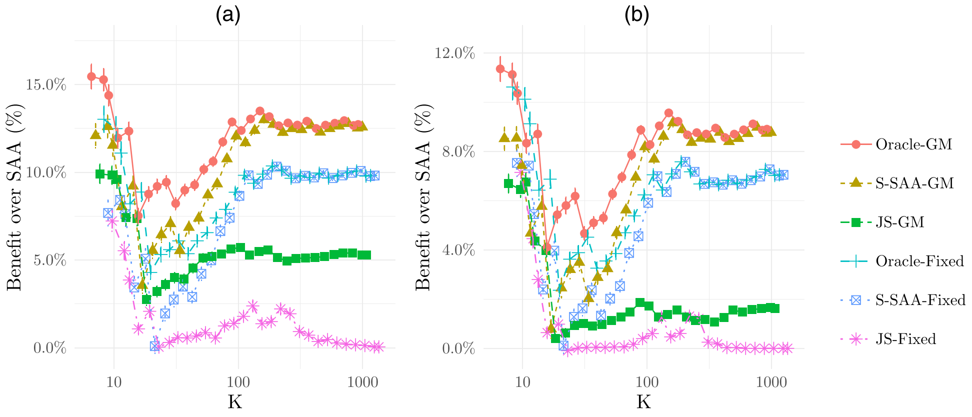
Shrunken-SAA significantly outperforms decoupling even for K as small as 10. For large K, the benefit is as large as 10%–15%. Both Shrunken-SAA policies converge quickly to their oracle benchmarks. JS policies also outperform the decoupled solutions but by a smaller amount (5%–10%). Shrinking to the grand mean outperforms shrinking to the uniform distribution, since, as observed earlier, the true distributions are far from uniform and have quantiles far from the uniform quantile. The grand-mean policies perform comparably to our Beta policies (see Table EC.1).

Section E.4 in Online Appendix E presents additional results such as the standard deviation of performance and the amount of pooling for each variant. Overall, Shrunken-SAA methods are less variable and pool more than competitors. In particular, JS variants pool very little because of the demand heterogeneity.

6.2. Relaxing Assumption 3.1

We next repeat the experiment of the previous section but now simulate data with $\hat{N}_k = 10$ for all k and all runs. Results are shown in the second panel of Figure 5 and in Section E.4 in Online Appendix E. We see the same qualitative features. Specifically, our Shrunken-SAA methods converge to oracle performance, and, even for moderate K, they significantly outperform decoupling. The JS methods offer a much smaller improvement over SAA. Many of the other features with respect to convergence in α and standard deviation of the performance are also qualitatively similar.

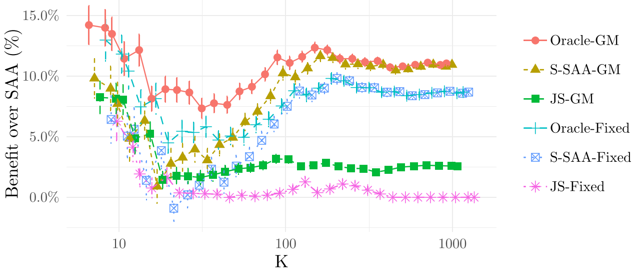
Figure 5. (Color online) Robustness to Assumption 3.1



$\hat{N}_k \sim \text{Poisson}(10)$ $\hat{N}_k = 10$ (non-random)

Notes. Simulated data. In panel (a), the amount of data per store follows Assumption 3.1. In panel (b), it is fixed (nonrandom) for all runs. Error bars show ±1 standard error.

Figure 6. (Color online) Historical Backtest



Notes. We evaluate our policies on historical data using $d = 20$. Error bars show ±1 standard error.

These results support our claim that Assumption 3.1 is not crucial to performance.

6.3. Historical Backtest

We next consider a more realistic setting. We employ repeated random subsampling validation with our data to assess each method: for each store, we select 10 days randomly from the data set for training and an additional 10 days for testing. Since store k may be missing data on these days, we train (respectively, test) with at most 10 points. We use repeated, random subsampling validation instead of five-fold cross-validation in order to limit the number of data points $\hat{N}_k$ used in each subproblem.

Figure 6 shows results with $d = 20$ for a subset of policies. See Table EC.3 in Online Appendix E for all policies. Importantly, we see the same features as in our synthetic data experiment: our Shrunken-SAA methods converge to oracle optimality and offer a substantive improvement over SAA for large enough K. They also outperform JS variants that do not leverage the optimization structure.

6.4. Other Experiments with Synthetic and Real Data

Sections E.6–E.8 in Online Appendix E study the robustness of Shrunken-SAA to the number of support points d, its performance as $N \longrightarrow \infty$, and compares computationally cheaper variants of the algorithm that substitute two-fold or five-fold cross-validation for the LOO validation step. We omit details for space. Generally, we find that (i) Shrunken-SAA is quite robust to d; (ii) as N increases, Shrunken-SAA retains many of SAA's strong large-sample properties; (iii) Other forms of cross-validation perform quite well and are viable alternatives in computationally limited settings.

7. Conclusion

We introduce the data-pooling phenomenon for stochastic optimization—when simultaneously solving many data-driven stochastic optimization subproblems, pooling data across subproblems may improve performance, even when (1) subproblems are unrelated and (2) data for each subproblem are independent. We propose Shrunken-SAA, a simple algorithm that exploits this phenomenon, and prove that, as the number of subproblems grows large, Shrunken-SAA identifies whether pooling can improve upon decoupling and, if so, the ideal amount to pool, even if the amount of data per subproblem is fixed and small. Empirical evidence further suggests that Shrunken-SAA offers significant benefits for a wide variety of problems. We hope that our work inspires researchers to think of data pooling as an "additional knob" to leverage in data-driven decision-making under uncertainty.

Acknowledgments

The authors thank the editors and three anonymous reviewers for their feedback.

Endnotes

[1] Section 4.6 discusses relaxing this discrete support assumption.

[2] This solution is nonunique, and the solution $\mathbb{I}[\hat{p}_{k1} > \frac{1}{2} + \frac{\alpha}{\hat{N}_k}(\frac{1}{2} - p_{01})]$ is also valid. We adopt the former solution in what follows, but our comments apply to either solution.

[3] In particular, $\mathbb{E}[Wf(W)] = \sum_{w=0}^{\infty} wf(w)e^{-\lambda}\frac{\lambda^w}{w!} = \lambda\sum_{w=0}^{\infty} f(w)e^{-\lambda}\frac{\lambda^{w-1}}{(w-1)!} = \lambda\mathbb{E}[f(W+1)]$.

[4] Strictly speaking, equation (7.5) of Pollard (1990) shows that $\mathbb{E}[|\sup_{t\in\mathcal{T}}|\overline{W}_K(t) - \mathbb{E}[\overline{W}_K(t)]||^p] \leq 2^p C_p^p \mathbb{E}[J^p]K^{-p}$, for some constant C_p that relates the ℓ_p norm of a random variable and a particular Orlicz norm. In Lemma B.4 in Online Appendix B, we prove that it suffices to take $C_p = 5^{1/p}\sqrt{\frac{p}{2e}}$. The result then follows from Markov's inequality.

[5] Recall that $D(\epsilon, S)$ is defined with respect to ℓ_2-distance.

References

Beran R (1996) Stein estimation in high dimensions: A retrospective. Brunner E, Denker M, eds. *Research Developments in Probability and Statistics* (VSP, Leiden, Netherlands), 91–110.

Bousquet O, Elisseeff A (2002) Stability and generalization. *J. Machine Learn. Res.* 2(March):499–526.

Brown LD (1971) Admissible estimators, recurrent diffusions, and insoluble boundary value problems. *Ann. Math. Statist.* 42(3):855–903.

Brown LD, Zhao LH (2012) A geometrical explanation of Stein shrinkage. *Statist. Sci.* 27(1):24–30.

Chen LHY (1975) Poisson approximation for dependent trials. *Ann. Probab.* 3(3):534–545.

Davarnia D, Cornuéjols G (2017) From estimation to optimization via shrinkage. *Oper. Res. Lett.* 45(6):642–646.

Deheuvels P, Pfeifer D (1988) Poisson approximations of multinomial distributions and point processes. *J. Multivariate Anal.* 25(1):65–89.

DeMiguel V, Martin-Utrera A, Nogales FJ (2013) Size matters: Optimal calibration of shrinkage estimators for portfolio selection. *J. Banking Finance* 37(8):3018–3034.

Efron B, Hastie T (2016) *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science* (Cambridge University Press, New York).

Efron B, Morris C (1977) Stein's paradox in statistics. *Sci. Amer.* 236(5):119–127.

Esfahani PM, Kuhn D (2018) Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Math. Programming* 171(1–2): 115–166.

Hastie T, Tibshirani R, Friedman J (2001) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York).

Gupta V, Rusmevichientong P (2021) Small-data, large-scale linear optimization with uncertain objectives. *Management Sci.* 67(1):220–241.

Jorion P (1986) Bayes-Stein estimation for portfolio analysis. *J. Financial Quant. Anal.* 21(3):279–292.

Kleywegt AJ, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.* 12(2):479–502.

Levi R, Perakis G, Uichanco J (2015) The data-driven newsvendor problem: New bounds and insights. *Oper. Res.* 63(6):1294–1306.

McDonald DR (1980) On the Poisson approximation to the multinomial distribution. *Canadian J. Statist.* 8(1):115–118.

Mukherjee G, Brown LD, Rusmevichientong P (2015) Efficient empirical Bayes prediction under check loss using asymptotic risk estimates. Preprint, submitted October 30, https://arxiv.org/abs/1511.00028.

Pollard D (1990) *Empirical Processes: Theory and Applications* (Institute of Mathematical Sciences, Haywood, CA; American Statistical Association, Alexandria, VA).

Shalev-Shwartz S, Shamir O, Srebro N, Sridharan K (2010) Learnability, stability and uniform convergence. *J. Machine Learn. Res.* 11:2635–2670.

Shapiro A, Dentcheva D, Ruszczyński A (2009) *Lectures on Stochastic Programming: Modeling and Theory* (SIAM, Philadelphia).

Stein C (1956) Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. *Proc. 3rd Berkeley Sympos. Math. Statist. Probab.*, vol. 1 (University of California Press, Berkeley), 197–206.

Stigler SM (1990) The 1988 Neyman Memorial Lecture: A Galtonian perspective on shrinkage estimators. *Statist. Sci.* 5(1):147–155.

Yu B (2013) Stability. *Bernoulli* 19(4):1484–1500.