Robust Optimization using Machine Learning for Uncertainty Sets

Theja Tulabandhula and **Cynthia Rudin**
MIT, Cambridge MA 02139

Abstract

Our goal is to build robust optimization problems that make decisions about the future, and where complex data from the past are used to model uncertainty. In robust optimization (RO) generally, the goal is to create a policy for decision-making that is robust to our uncertainty about the future. In particular, we want our policy to best handle the the worst possible situation that could arise, out of an *uncertainty set* of possible situations. Classically, the uncertainty set is simply chosen by the user, or it might be estimated in overly simplistic ways with strong assumptions; whereas in this work, we learn the uncertainty set from complex data from the past. The past data are drawn randomly from an (unknown) possibly complicated high-dimensional distribution. We propose a new uncertainty set design and show how tools from statistical learning theory can be employed to provide probabilistic guarantees on the robustness of the policy.
Keywords: machine learning, uncertainty sets, robust optimization.

1 Introduction

In this work, we consider a situation often faced by decision makers: a policy needs to be created for the future that would be a best possible reaction to the worst possible uncertain situation; this is a question of *robust optimization*. In our case, the decision maker does not know what the worst situation might be, and uses complex data to estimate the *uncertainty set*, which is the set of uncertain future situations. Here we are interested in answering questions such as: How might we construct a principled uncertainty set from these complex data? Can we ensure that with high probability our policy will be robust to whatever the future brings?

In this paper we address the important setting where detailed data (features) are available to predict each possible future situation. We turn to predictive modeling techniques from machine learning to make predictions, and to define uncertainty sets. Models created from finite data are uncertain: given a collection of historical data, there many be many predictive models that appear to be equally good, according to any measure of predictive quality. This was called the *Rashomon effect* by statistician Breiman in [1], and it is this source of uncertainty in learning that we capture while designing uncertainty sets.

Our concept is possibly best explained through an illustrative example. Consider the maximum return portfolio allocation problem where our goal is to construct a portfolio of assets. Let us temporarily say that we know exactly what the return for each of the assets in the market will be, and denote $\mathbf{r}$ as the vector of these known returns. Let the covariance of the returns be $\mathbf{\Sigma}$ which is also known in advance. We denote $\boldsymbol{\pi}$ as our choice of portfolio weights. We thus solve the basic decision-making problem:

$$\max_{\boldsymbol{\pi}} \mathbf{r}^T\boldsymbol{\pi} \quad \text{s.t. } \boldsymbol{\pi}^T\mathbf{1} = 1, \;\; \boldsymbol{\pi}^T\mathbf{\Sigma}\boldsymbol{\pi} \leq c, \;\; \boldsymbol{\pi} \geq 0,$$

where $()^T$ is the transpose operator, c is a constant and $\mathbf{1}$ is the vector of all ones. The three constraints represent that: (a) the sum of portfolio weights should be equal to one, (b) the variance of the portfolio return should be bounded and (c) the portfolio weights should be non-negative. Now let us consider the more realistic case where the returns $\mathbf{r}$ are not known in advance, and we need to make a decision about portfolio weights $\boldsymbol{\pi}$ under uncertainty. If we are able to encode our uncertainty about these forecasted returns using an uncertainty set $\mathcal{U}$, then we can take a robust optimization (RO) approach and solve the following:

$$\max_{\boldsymbol{\pi}} \min_{\mathbf{r}\in\mathcal{U}} \mathbf{r}^T\boldsymbol{\pi} \quad \text{s.t. } \boldsymbol{\pi}^T\mathbf{1} = 1, \;\; \boldsymbol{\pi}^T\mathbf{\Sigma}\boldsymbol{\pi} \leq c, \;\; \boldsymbol{\pi} \geq 0,$$

which gives us a best response to the worst possible outcome $\mathbf{r}$ in uncertainty set $\mathcal{U}$. The uncertainty set $\mathcal{U}$ can be defined in many ways, and the central goal of this work is how to model $\mathcal{U}$ from complex data from the past. These data take the form of features and labels; for instance in the portfolio allocation problem, the data are $\{(\mathbf{x}^i, \mathbf{r}^i)\}_{i=1}^n$ where an observation $\mathbf{x}^i \in \mathcal{X}$ represents information we could use to predict the returns $\mathbf{r}^i \in \mathcal{Y}$ on past day i. These data might include macroeconomic indicators such as interest rates, employment statistics, retail sales and so on, as well as features of the assets themselves. Having complex data like this is very common, but often is not considered carefully within the decision problem. Some of the different ways uncertainty sets can be constructed are:

• Using a priori assumptions: We may have *a priori* knowledge about the range of possible future situations. In the portfolio allocation problem, we can assume that we know all possible values of the returns. This knowledge can guide us in constructing the returns uncertainty set $\mathcal{U}$ using interval constraints. That is, $\mathcal{U} := \{\mathbf{r} : \;\; \forall j \;\; r_j \in [\underline{r}_j, \overline{r}_j]\}$.

Here we ignore the complex past data altogether.

• Using empirical statistics: We could create an uncertainty set using empirical statistics of the data. In the portfolio allocation problem, we might define $\mathcal{U}$ to be the set of all return vectors that are close to return vectors $\mathbf{r}_i$ that have been realized in the past. Or, $\mathcal{U}$ could be the convex hull of past returns vectors. Here we ignore the $\mathbf{x}^i$'s altogether.

• Using linear regression to model complex data: Here, we use the complex past data $\{(\mathbf{x}^i, \mathbf{r}^i)\}_{i=1}^n$, but we make strong (potentially incorrect) assumptions on the probability distribution these data are drawn from. We use these assumptions to define a class of "good" predictive models $\mathcal{B}$ from $\mathcal{X} \to \mathcal{Y}$. Then, given a new feature vector $\mathbf{x}$, we use $\mathcal{B}$ to define an intermediate uncertainty set $\mathcal{U}_\mathcal{B}$ of all possible outcomes for each situation $\mathbf{x}$, and an uncertainty set $\mathcal{U}_{-\mathcal{B}}$ to capture model residuals.

For the portfolio allocation problem, we define $\mathcal{B}$ as all linear models $\beta : \mathcal{X} \to \mathcal{Y}$ that fall in the confidence interval determined using a linear regression fit under the usual normality assumption. We then define $\mathcal{U}_\mathcal{B}$ as predicted returns from these "good" models given a new feature vector $\mathbf{x}$. Additionally, using past data and normality assumptions, we can define the set of model residuals $\mathcal{U}_{-\mathcal{B}}$. Finally, $\mathcal{U}_\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$ are used to define the set $\mathcal{U}$ in the robust portfolio allocation formulation above.

• Using machine learning to model complex data: This setting is more general than linear regression and with much weaker assumptions. Methods that make strong assumptions have limited applicability for modern datasets with thousands of features, and such assumptions may hinder prediction performance. In this work, we only make a single assumption: *with high probability, the error due to the "best in class" model β^* is bounded with a known constant.* Our policies need to be robust to β^* that we would choose if we knew the distribution of data. Thus, we make efforts to ensure that the set of good models $\mathcal{B}$ that we will construct contains β^*. Here, $\mathcal{B}$ and $\mathcal{U}_\mathcal{B}$ are chosen in a distribution-independent manner, based on learning theory results.

Being able to define uncertainty sets from predictive models is important: the uncertainty sets can now be specialized to a given new situation $\tilde{\mathbf{x}} \in \mathcal{X}$, and this is true even if we have never seen $\tilde{\mathbf{x}}$ before. For instance, when ordering daily supplies $\mathbf{r}^i$ for an ice cream parlor in Boston, an uncertainty set that depends on the weather $\mathbf{x}^i$ might be much smaller than one that does not: it would not be wise to budget for the largest possible summer sales in the middle of the winter. As the amount of data used for the modeling increases, the prediction models become more accurate, and can help us make quantitatively better decisions. Though there have been attempts to define uncertainty sets in the linear regression setting [2], ours is the first attempt to tackle the more general setting in a principled way.

Our goals are twofold: (i) We would like to create uncertainty sets for the more general machine learning setting discussed above. In particular, our uncertainty sets are chosen to include predictions from all models in the hypothesis set $\mathcal{B}_0$ that have low enough training error (low in-sample prediction error). The uncertainty sets we propose are generated using statistical learning theory [3]. (ii) We would like to consider the problem of *sample complexity*. In particular, we determine how much data the practitioner needs for a guarantee that their chosen policy will be robust to future situations. We will produce a probabilistic bound to determine this.

Our approach for constructing uncertainty sets is flexible, intuitive, easy to understand from a practitioner's point of view, and at the same time can bring all the rich theoretical results of learning theory to justify the data-driven methodology. Our uncertainty set designs can handle prediction models for classification, regression, ranking and other supervised learning problems. A main theme of this work is that RO is a new context in which many learning theory results naturally apply and can be directly used.

In Section 3, we formulate our problem and provide a workflow for making decisions under learning uncertainty. In Section 4, we use learning theory techniques to define uncertainty sets and conclude in Section 5.

2 Background Literature

There are many approaches to decision making under uncertainty when the uncertainty stems from learning using finite data. In the optimization literature, there has been a continued interest in modeling uncertainty sets for robust optimization (RO) using empirical statistics of data, along with (strong) a priori assumptions about the probability distribution generating the parameters of a particular model for the data (e.g., [4]). Gupta et al. [5] explore a way to specify data-driven uncertainty sets with probabilistic guarantees, where statistical hypothesis testing is used to construct sets. This approach has the weaknesses that (i) the hypothesis tests require assumptions (e.g., normality), and if these assumptions are false, then the robustness is jeopardized and the guarantees are incorrect, and (ii) the method is designed for non-complex featureless data. The closest work to ours is possibly that of Goldfarb and Iyengar [2], who provide a linear-regression-based robust decision making paradigm for portfolio allocation problems, where they assume a multivariate linear regression model for the learning step. A big departure from this approach is that in our work, we are able to design uncertainty sets for a general class of decision making problems while making weak assumptions about the distributional aspects of the historical data. We base our uncertainty set design on regularized empirical risk minimization, which allows us to include the best-in-class prediction model in our uncertainty set with high probability.

Other paradigms that also use empirical statistics of data are *chance constrained programming* [6] and various other *stochastic programming* techniques. Both stochastic programming and robust optimization have extensions, for instance, for multi-stage decision making. We focus on single stage optimization to highlight the importance of learning uncertainty.

3 Formulation

Let $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ represent a feature vector and $y \in \mathcal{Y}$ represent a label. Let $\beta : \mathcal{X} \mapsto \mathcal{Y}$ be a prediction model in the hypothesis class $\mathcal{B}_0$. For instance, $\mathcal{B}_0$ can be the set of linear predictors $\mathcal{B}_0 = \{x \mapsto \beta^T x : \|\beta\| \leq B_b\}$. Let $l(\beta(\mathbf{x}), y)$ denote the loss function. The loss measures the discrepancy between the prediction of a model and the true outcome/label. For example, $(\beta(\mathbf{x}) - y)^2$ is the least squares loss and $[1 - \beta(\mathbf{x})y]_+$ is the hinge loss. For any given model, let $l_{\mathbb{P}}(\beta) = \mathbb{E}_{\mathbf{x},y}[l(\beta(\mathbf{x}), y)]$ where the expectation is with respect to $(\mathbf{x}, y) \sim \mathbb{P}_{\mathbf{x},y}$ which is unknown. Let $\beta^* \in \arg\min_{\beta \in \mathcal{B}_0} l_{\mathbb{P}}(\beta)$ be defined as the "best in class" model with respect to our class $\mathcal{B}_0$. Note that we cannot calculate β^* as we do not have the distribution.

Our bound will depend on how much the mass of $\mathbb{P}_{\mathbf{x},y}$ concentrates around $\beta^*(\mathbf{x})$. It is always true that there exists a set E and a scalar $\delta_e \geq 0$ such that:

$$\mathbb{P}_{\mathbf{x},y}\left(\mathbf{x}, y : |y - \beta^*(\mathbf{x})| \in E\right) \geq 1 - \delta_e. \tag{1}$$

This is trivially satisfied if $E = \mathcal{Y}$. In this case, δ_e can be set to 0. The quality of the robust solution of Equation (3) depends on the set E. For a higher quality solution, we want set E to be as small as possible. The probabilistic guarantee on the robust solution that we derive in Section 4 depends on δ_e. For a better guarantee, we need δ_e to be as close as possible to 0. If our model class $\mathcal{B}_0$ is very complex and able to closely capture most y values, this could reduce the size of set E. Thus we formalize the assumption:

Assumption A: We know a pair (E, δ_e) such that Equation (1) holds.

Let $\{\tilde{\mathbf{x}}^j\}_{j=1}^m$ be the feature vectors on which we make predictions, and these predictions parameterize a decision making problem. In particular, let all the uncertain parameters of the decision problem be denoted by a vector $\mathbf{u}$ and let $\mathbf{u}_\beta$ be the part of $\mathbf{u}$ that is derived from a statistical model (subscript β is used to the corresponding statistical model β). Thus, given $\{\tilde{\mathbf{x}}^j\}_{j=1}^m$, $\mathbf{u}_\beta := [\beta(\tilde{\mathbf{x}}^1) \ \cdots \beta(\tilde{\mathbf{x}}^m)]^T$. Let the remaining part of $\mathbf{u}$, denoted by $\mathbf{u}_{-\beta}$, include the realizations of model residuals and realizations of any other set of random variables parameterizing the decision problem. Without loss of generality, let $\mathbf{u}^T = [\mathbf{u}_\beta^T \ \ \mathbf{u}_{-\beta}^T]$.

Given a realization of $\mathbf{u}$, let the (basic non-robust) decision making problem be written as:

$$\min_{\boldsymbol{\pi}} f(\boldsymbol{\pi}, \mathbf{u}) \tag{2}$$
$$\text{subject to}$$
$$F(\boldsymbol{\pi}, \mathbf{u}) \in \mathcal{K}$$

Here $\boldsymbol{\pi} \in \Pi \subseteq \mathbb{R}^{d_1}$ is the decision vector, $\mathbf{u} \in \mathbb{R}^{d_2}$ is the parameter and $f : \Pi \times \mathbb{R}^{d_2} \to \mathbb{R}$ is the objective function. Function $F : \Pi \times \mathcal{U} \to \mathcal{K}$ and convex cone $\mathcal{K} \subseteq \mathbb{R}^{d_3}$ describe the constraints of the problem.

The robust version of the decision problem in Equation (2) is thus:

$$\min_{\boldsymbol{\pi}} \max_{\mathbf{u} \in \mathcal{U}} f(\boldsymbol{\pi}, \mathbf{u}) \tag{3}$$
$$\text{subject to}$$
$$F(\boldsymbol{\pi}, \mathbf{u}) \in \mathcal{K} \text{ for all } \mathbf{u} \in \mathcal{U}$$

where $\mathcal{U} \subset \mathbb{R}^{d_2}$ represents the uncertainty set.

Let $\mathcal{B}$ represent a set of "good" prediction models. Let $\mathcal{U} = \mathcal{U}_\mathcal{B} \times \mathcal{U}_{-\mathcal{B}}$ such that $\mathbf{u}_\beta \in \mathcal{U}_\mathcal{B}$ and $\mathbf{u}_{-\beta} \in \mathcal{U}_{-\mathcal{B}}$. Here, $\mathcal{U}_\mathcal{B}$ corresponds to $\mathcal{B}$ in the following way: $\mathcal{U}_\mathcal{B} := \{\mathbf{u}_\beta : \beta \in \mathcal{B}\}$. On the other hand, $\mathcal{U}_{-\mathcal{B}}$ corresponds to a set that captures the support of most model error residuals and other random variables.

In Section 1, the maximum return portfolio allocation problem is a specific instance of the decision problem in Equation (2). The robust portfolio allocation problem is an instantiation of the robust formulation in Equation (3), where $\mathcal{U}_\mathcal{B}$ captures all the predictions of a set of "good" models and $\mathcal{U}_{-\mathcal{B}}$ captures the support of model residuals.

If we know $\mathbf{u}_\beta$ and $\mathbf{u}_{-\beta}$ beforehand with certainty (where they are not random anymore), then we do not need to construct $\mathcal{U}_\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$. The interesting case is when the "best in class" model is not known and as a result modeling $\mathcal{U}_\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$ is essential. To solve Equation (3), we prescribe the following steps:

Step 1: Construct $\mathcal{U}_\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$.

(a) Define $\mathcal{B}$ using $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$. We will propose procedures for designing $\mathcal{B}$ using learning theory results in Section 4. Our sets will be of the form:

$$\mathcal{B} = \{\beta : g(\beta) \leq g(\beta^{Alg}) + c\}$$

where g is some function, β^{Alg} is a specific model and c is a parameter. These quantities will depend on the learning algorithm and $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$.

(b) Define $\mathcal{U}_\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$: Recall that $\mathcal{U}_\mathcal{B} := \{\mathbf{u}_\beta : \beta \in \mathcal{B}\}$ where $\mathbf{u}_\beta = [\beta(\tilde{\mathbf{x}}^1) \ \cdots \beta(\tilde{\mathbf{x}}^m)]^T$. $\mathcal{U}_{-\mathcal{B}}$ captures the support of model residuals and other sources of randomness in the decision problem. This is defined in Section 4 using the property of the model residuals in Equation (1). The cartesian product of $\mathcal{U}_\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$ is $\mathcal{U}$.

Step 2: Obtain a robust solution.

Option 1: If $\mathcal{U}_\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$ are "nice" sets that can be bounded using simple sets (such as a box or an ellipsoid) or if they admit transforming the generic semi-infinite formulation in Equation (3) to a finite formulation, then solve the transformed problem to obtain a robust solution $\boldsymbol{\pi}^*$.

Option 2: If $\mathcal{U}_\mathcal{B}$ is not a "nice" set, the do the following: sample L elements $\{\beta\}_{l=1}^L$ from $\mathcal{B}$ uniformly. For instance, this can be done using geometric random walks if $\mathcal{B}$ is convex. This defines a finite set $\mathcal{U}_\mathcal{B}$. If $\mathcal{U}_{-\mathcal{B}}$ is also not a "nice" set, then sample L' elements from it. Solve the sampled version of Equation (3) to obtain a robust solution $\boldsymbol{\pi}^*$ (this assumes we have a procedure to sample from $\mathcal{B}$ and/or $\mathcal{U}_{-\mathcal{B}}$).

4 Uncertainty Sets

We will construct the uncertainty sets $\mathcal{U}_\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$ for both the general machine learning and linear regression settings. Since $\mathcal{U}_\mathcal{B}$ is defined using the set $\mathcal{B}$, we will focus our discussion on constructing $\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$. We call $\mathcal{B}$ the precursor uncertainty set, or the set of "good" models. While constructing $\mathcal{U}_{-\mathcal{B}}$, we will assume that it only captures the support

of model residuals and there is no other randomness. This is without loss of generality as other sources of uncertainty in the decision problem can be captured using one of the four ways mentioned in Section 1.

Let $S := \{(\mathbf{x}^i, y^i)\}_{i=1}^n$ be the training data which are independent and identically distributed. Let algorithm A represent a generic learning procedure. That is, it takes S as an input and outputs $\beta^{Alg} \in \mathcal{B}_0$. Let $l_S(\beta) = \frac{1}{n}\sum_{i=1}^n l(\beta(\mathbf{x}^i), y^i)$ be a function of our sample S. Let A produce β^{Alg} according to $\beta^{Alg} \in \arg\min_{\beta\in\mathcal{B}_0} l_S(\beta)$. That is, the algorithm A is minimizing the empirical loss.

4.1 Using machine learning to construct $\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$:

First we will propose sets $\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$ and then describe the probabilistic guarantees a robust solution of Equation (3) enjoys when $\mathcal{U} = \mathcal{U}_B \times \mathcal{U}_{-B}$.

Constructing $\mathcal{B}$: To construct $\mathcal{B}$, we will need the following quantity known as the empirical Rademacher average. For a set $\mathcal{H}$ of functions, the *empirical Rademacher average* is defined with respect to a given random sample $S' = \{z^i\}_{i=1}^n$ as

$$\mathcal{R}_{S'}(\mathcal{H}) = \mathbb{E}_{\sigma^1,\ldots,\sigma^n}\left[\frac{1}{n}\sup_{h\in\mathcal{H}}\sum_{i=1}^n \sigma^i h(z^i)\right]$$

where for each $i = 1,..,n$, $\sigma^i = \pm 1$ with equal probability. The interpretation of the Rademacher average is that it measures the ability of function class $\mathcal{H}$ to fit noise, coming from the random $\sigma_i's$. If the function class can fit noise well, it is a highly complex class. The Rademacher average is one of many ways to measure the richness of a function class, including covering numbers, fat-shattering dimensions and the Vapnik-Chervonenkis dimension.

With the above definition we can define $\mathcal{B}$ as follows:

$$\mathcal{B} := \Bigg\{ \beta \in \mathcal{B}_0 : \\ l_S(\beta) \le l_S(\beta^{Alg}) + 2\mathcal{R}_S(l\circ\mathcal{B}_0) + 4M\sqrt{\frac{\log\frac{3}{\delta}}{2n}} \Bigg\}, \tag{4}$$

where M is a bound on the range of the loss function l and δ is pre-specified, as we discuss below. Also, n is the number of examples in our data S, $l_S(\beta^{Alg})$ is the empirical loss of the predictive model β^{Alg} and $\mathcal{R}_S(l\circ\mathcal{B}_0)$ is the empirical Rademacher complexity of the function class $l\circ\mathcal{B}_0 := \{\beta \mapsto l(\beta(\cdot),\cdot) : \beta \in \mathcal{B}_0\}$.

By plugging in different loss functions, we obtain different uncertainty sets. For instance, when $l(\beta(\mathbf{x}, y)) = (\beta(\mathbf{x}) - y)^2$, we have:

$$\mathcal{B} = \Big\{\beta : \sum_{i=1}^n (y^i - \beta(\mathbf{x}^i))^2 \le \\ \sum_{i=1}^n (y^i - \beta^{Alg}(\mathbf{x}^i))^2 + 2\mathcal{R}_S(l\circ\mathcal{B}_0) + 4M\sqrt{\frac{\log\frac{3}{\delta}}{2n}}\Big\}.$$

One of the advantages of defining precursor uncertainty set $\mathcal{B}$ in this way is that it directly links the uncertainty in decision making to the loss function $l(\beta(\mathbf{x}), y)$ and S of the machine learning step. We chose, among other choices, the empirical Rademacher average in defining $\mathcal{B}$ because the other choices such as covering number and VC-dimension do not make use of the data sample S in their definition, whereas the empirical Rademacher average can reflect the properties of the particular unknown distribution $\mathbb{P}_{\mathbf{x},y}$ of the data source.

Constructing $\mathcal{U}_{-\mathcal{B}}$: We define $\mathcal{U}_{-\mathcal{B}} := E^m$ (m copies of E) where E is satisfies Equation (1) for a given δ_e and m is the number of predictions (equal to the length of the vector $\mathbf{u}_\beta$). Intuitively, $\mathcal{U}_{-\mathcal{B}}$ is capturing the support of prediction errors if we knew the "best in class" model β^*.

Our construction of $\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$ leads us to a theoretical guarantee on the robust optimal solution $\boldsymbol{\pi}^*$ if the loss function that we pick is $\mathcal{L}$-Lipschitz. We will not require assumptions on the unknown data distribution to state our guarantee. For instance, we will not assume that the data came from a linear model with normal noise.

Theorem 4.1. *If $\mathcal{U}_{\mathcal{B}}$ is defined using $\mathcal{B}$ described in Equation (4) and $\mathcal{U}_{-\mathcal{B}}$ is defined as described above, using Equation (1) and Assumption A, then the following hold:*

1. *With probability at least $1-\delta$, $\beta^* \in \mathcal{B}$.*
2. *Robust optimal solution $\boldsymbol{\pi}^*$ of Equation (3) is feasible for unknown $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$ with probability at least $1-(\delta + m\delta_e)$. That is,*

$$\mathbb{P}_{S,\{(\tilde{\mathbf{x}}^j,\tilde{y}^j)\}_{j=1}^m}(F(\boldsymbol{\pi}^*, \mathbf{u}) \in \mathcal{K}) \ge 1 - (\delta + m\delta_e),$$

where $\mathbf{u}^T = [\mathbf{u}_{\beta^*}^T \;\; \mathbf{u}_{-\beta^*}^T]$.

This theorem provides a guarantee that the robust optimal solution we find will be robust to the unknown $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$. In particular, π^* will be robust to $\mathbf{u}$ with components

$$\mathbf{u}_\beta = [\beta^*(\tilde{\mathbf{x}}^1)\ldots\beta^*(\tilde{\mathbf{x}}^j)\ldots\beta^*(\tilde{\mathbf{x}}^m)]^T$$

and

$$\mathbf{u}_{-\beta} = [\tilde{y}^1\ldots\tilde{y}^j\ldots\tilde{y}^m]^T - [\beta^*(\tilde{\mathbf{x}}^1)\ldots\beta^*(\tilde{\mathbf{x}}^j)\ldots\beta^*(\tilde{\mathbf{x}}^m)]^T.$$

The theorem holds for any choice of loss function l_S obeying the Lipschitz and boundedness properties.

We attempt to insure against all possible predictions made by the "best in class" model β^* in a particular way: by first ensuring β^* belongs to $\mathcal{B}$ with high probability in Theorem 4.1 and then ensuring that the random errors $\tilde{y}^j - \beta^*(\tilde{\mathbf{x}}^j)$ are in $\mathcal{U}_{-\mathcal{B}}$ also with high probability via Equation (1). Thus the true $\{\tilde{y}^j\}_{j=1}^m$ belong to the cartesian product $\mathcal{U}_{\mathcal{B}} \times \mathcal{U}_{-\mathcal{B}}$ with high probability.

Remark 4.2. This theorem tells us how the choice of $\mathcal{B}_0$ affects the size of our precursor uncertainty set $\mathcal{B}$. Interestingly enough, if we work with a (possibly infinite) set of predictive models $\mathcal{B}_0$ such that their empirical Rademacher average $\mathcal{R}_S(l\circ\mathcal{B}_0)$ scales as $O(n^{-\frac{1}{2}})$, then we have similar quantitative dependence on n compared to that of confidence-interval based approaches (that make explicit distributional

assumptions; see Section 4.4). In fact, for many interesting model classes the scaling of the empirical Rademacher complexity is indeed $O(n^{-\frac{1}{2}})$ which we will show shortly.

Computing terms in the definition of $\mathcal{B}$: The terms appearing in the expression for $\mathcal{B}$ in Equation (4) are all computable; however, it may sometimes be difficult to compute the value of $\mathcal{R}_S(l \circ \mathcal{B}_0)$ efficiently. In these cases, we have two options. The first one involves finding upper bounds on $\mathcal{R}_S(l \circ \mathcal{B}_0)$. This can be tricky as $\mathcal{R}_S$ depends on the data. The second one involves defining $\mathcal{B}$ directly in terms of *Rademacher average* $\mathcal{R}(l \circ \mathcal{B}_0)$ and using it in the robust decision making problem:

$$\mathcal{B} := \Bigg\{ \beta \in \mathcal{B}_0 : \\ l_S(\beta) \leq l_S(\beta^{Alg}) + 2\mathcal{R}(l \circ \mathcal{B}_0) + 3M\sqrt{\frac{\log \frac{2}{\delta}}{2n}} \Bigg\}, \quad (5)$$

where the *Rademacher average* is defined to be the expectation of the empirical Rademacher average over the random sample S:

$$\mathcal{R}(\mathcal{H}) = \mathbb{E}_{z^1,\ldots,z^n}\left[\mathcal{R}_S(\mathcal{H})\right]. \quad (6)$$

It can be shown that the optimal robust solution obtained using the set in (5) enjoys a guarantee similar to the solution obtained using the set in (4) with slightly different constants. Thus we can either bound the Rademacher average $\mathcal{R}(l \circ \mathcal{B}_0)$ or its empirical version $\mathcal{R}_S(l \circ \mathcal{B}_0)$ in order to define $\mathcal{B}$. In this regard, we can make use of the various relationships in Theorem 12 of [7]. The following are some example upper bounds of $\mathcal{R}(l \circ \mathcal{B}_0)$ and $\mathcal{R}_S(l \circ \mathcal{B}_0)$.

- Linear function class with squared loss:

  $$\mathcal{R}(\mathcal{B}_0) \leq \frac{X_b B_b}{\sqrt{n}}, \text{ and } \mathcal{R}(l \circ \mathcal{B}_0) \leq 4X_b B_b \frac{X_b B_b}{\sqrt{n}}$$

  where the latter inequality is obtained using Corollary 3.17 in [8], which relates $\mathcal{R}(l \circ \mathcal{B}_0)$ and $\mathcal{R}(\mathcal{B}_0)$. That is, when the loss function $l(\beta(\mathbf{x}), y)$ is $\mathcal{L}$-Lipschitz we have: $\mathcal{R}(l \circ \mathcal{B}_0) \leq \mathcal{L} \cdot \mathcal{R}(\mathcal{B}_0)$. For the squared loss function, $\mathcal{L} = 4X_b B_b$ if $\forall \mathbf{x} \in \mathcal{X}, \|\mathbf{x}\|_2 \leq X_b$ and $\forall \beta \in \mathcal{B}_0, \|\beta\|_2 \leq B_b$. This bound does not depend on data sample S. In general, bounds for $\mathcal{R}(l \circ \mathcal{B}_0)$ can be precomputed given a choice of loss function and model class.

- Kernel based function classes: In this setting, the function class $\mathcal{B}_0$ is:

  $$\mathcal{B}_0 = \Bigg\{ x \mapsto \sum_{i=1}^{n} \alpha^i k(x, x^i) : \\ n \in \mathbb{N}, x \in \mathcal{X}, \sum_{i,j} \alpha^i \alpha^j K(x^i, x^j) \leq B_b \Bigg\}$$

  where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a bounded kernel. (Here k is called a kernel if an $n \times n$ Gram matrix K with entries $(K)_{i,j} = k(x^i, x^j)$ is positive semi-definite.) This function class is used in Support Vector Machines (SVMs) (e.g., see [9]) when the loss function is chosen to be the hinge-loss. The following result (see Lemma 22 in [7]) can be used assuming the loss function is $\mathcal{L}$-Lipschitz as before to upper bound $\mathcal{R}_S(l \circ \mathcal{B}_0)$:

  $$\mathcal{R}_S(\mathcal{B}_0) \leq \frac{B_b}{n}\sqrt{\sum_{i=1}^{n} k(x^i, x^j)},$$

  $$\text{and} \quad \mathcal{R}_S(l \circ \mathcal{B}_0) \leq \mathcal{L}\frac{B_b}{n}\sqrt{\sum_{i=1}^{n} k(x^i, x^j)}.$$

  This upper bound is similar to the previous case (linear function class and squared loss) when we choose the appropriate kernel and loss function. In particular, using the dot product kernel $k(x^i, x^j) = (x^i)^T x^j$ we get:

  $$\mathcal{R}_S(l \circ \mathcal{B}_0) \leq \mathcal{L}\frac{B_b}{n}\sqrt{\sum_{i=1}^{n} k(x^i, x^j)} = \mathcal{L}\frac{B_b}{n}\sqrt{\sum_{i=1}^{n} (x^i)^T x^j} \\ \leq \mathcal{L}\frac{B_b}{n}\sqrt{nX_b^2} = 4X_b B_b \frac{X_b B_b}{\sqrt{n}}.$$

Proof of Theorem 4.1:

Consider the random variable $l_S(\beta^*) - l_S(\beta^{Alg})$, which depends on the random sample S. We can upper bound it by:

$$\begin{aligned} l_S(\beta^*) - l_S(\beta^{Alg}) \\ &= l_S(\beta^*) - l_{\mathbb{P}}(\beta^*) + l_{\mathbb{P}}(\beta^*) - l_S(\beta^{Alg}) \\ &\leq l_S(\beta^*) - l_{\mathbb{P}}(\beta^*) + l_{\mathbb{P}}(\beta^{Alg}) - l_S(\beta^{Alg}) \\ &\leq l_S(\beta^*) - l_{\mathbb{P}}(\beta^*) + \max_{\beta \in \mathcal{B}_0} \left(l_{\mathbb{P}}(\beta) - l_S(\beta)\right) \quad (7) \end{aligned}$$

where we added and subtracted $l_{\mathbb{P}}(\beta^*)$ in the first step, then in the second step substituted β^{Alg} for β^* in the third term to increase the value of the right hand side, and finally in the last step, replaced the last two terms with a max operation over $\mathcal{B}_0$.

The first term in the expression on the right hand side of (7) will go to zero in probability as $n \to \infty$ due to concentration, and this can be quantified for finite n via Hoeffding's inequality.

Lemma 4.3. (Hoeffding's inequality.) *Let $z^1, ..., z^n$ be n i.i.d. random variables and let h be a bounded function, $a \leq h(z) \leq b$. Then for all $\epsilon > 0$ we have*

$$\mathbb{P}_{z^1,\ldots,z^n}\left(\frac{1}{n}\sum_{i=1}^{n} h(z^i) - \mathbb{E}_{z^1}[h(z^1)] > \epsilon\right) \\ \leq \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right).$$

In our case, the sample S is represented by $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$. For the fixed function $\beta^*(x)$, the function $l(\beta^*(x), y)$ is bounded in the interval $[0, M]$. The empirical average $\frac{1}{n}\sum_{i=1}^{n} l(\beta^*(x^i), y^i)$ $(= l_S(\beta^*))$ thus gets close to its mean

$\mathbb{E}[l(\beta^*(x), y)]$ $(= l_{\mathbb{P}}(\beta^*))$ as n increases. Using the one-sided version of Hoeffding's inequality above, we see that with probability at least $1 - \delta_1$,

$$l_S(\beta^*) - l_{\mathbb{P}}(\beta^*) \leq M\sqrt{\frac{\log \frac{1}{\delta_1}}{2n}}. \quad (8)$$

The second term in (7), $\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta))$, is a random variable that depends on the sample S in a more complicated way than in the first term. We can use the (one-sided) McDiarmid's inequality to claim that this random variable is close to its mean as n increases.

Lemma 4.4. (McDiarmid's inequality.) *Let $z^1, ..., z^n$ be n i.i.d. random variables in a set A and $h(z^1, ..., z^n)$ be a function such that for all $i = 1, ..., n$*

$$\sup_{(z^1,...,z^n,\tilde{z}) \in A^{n+1}} |h(z^1, ..., z^i, ..., z^n) - h(z^1, ..., \tilde{z}, ..., z^n)| \leq c.$$

Then for all $\epsilon > 0$,

$$\mathbb{P}_{z^1,...,z^n}\left(h(z^1, ..., z^n) - \mathbb{E}[h(z^1, ..., z^n)] > \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{nc^2}\right).$$

In our case, the function h is $\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta))$. We can show that if the i^{th} instance in the sample S is perturbed, the maximum change in the function value is $\frac{M}{n}$. Let $\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta)) \geq \max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_{S^i}(\beta))$ where $l_{S^i}(\beta)$ is the same as $l_S(\beta)$ except for the i^{th} example, which is changed from $(\mathbf{x}^i, y^i)$ to a new example $\mathbf{x}^i_\circ, y^i_\circ$. Also let $\beta^\circ \in \arg\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta))$. Then,

$$\begin{aligned}
&\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta)) - \max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_{S^i}(\beta)) \\
&\leq (l_{\mathbb{P}}(\beta^\circ) - l_S(\beta^\circ)) - (l_{\mathbb{P}}(\beta^\circ) - l_{S^i}(\beta^\circ)) \\
&\leq - l_S(\beta^\circ) + l_{S^i}(\beta^\circ) \\
&= \frac{1}{n}\left(l(\beta^\circ(\mathbf{x}^i_\circ), y^i) - l(\beta^\circ(\mathbf{x}^i), y^i)\right) \leq \frac{M}{n}.
\end{aligned}$$

We can do an identical calculation to get the same upper bound $\frac{M}{n}$ if $\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta)) \leq \max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_{S^i}(\beta))$. Thus, with probability at least $1 - \delta_2$,

$$\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta)) \leq \mathbb{E}[\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta))] + M\sqrt{\frac{\log \frac{1}{\delta_2}}{2n}}. \quad (9)$$

The quantity $\mathbb{E}[\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta))]$ captures the complexity or size of $\mathcal{B}_0$ (or actually, its composition with the loss function l, which is the set $l \circ \mathcal{B}_0$). We can upper bound this quantity in terms of a Rademacher average (see Equation (6)) using a symmetrization trick.

Lemma 4.5. *(Upper bound)*

$$\mathbb{E}[\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta))] \leq 2\mathcal{R}(l \circ \mathcal{B}_0). \quad (10)$$

Proof. See Theorem 8 in [7].

The empirical Rademacher average also concentrates around its mean and this can be proved again by McDiarmid's inequality. In this case, from Lemma 4.4, the function h is represented by $\mathcal{R}_S(l \circ \mathcal{B}_0)$. We can again show (Theorem 11 in [7]) that if the i^{th} instance in the sample S is perturbed, the maximum change in the function value is $\frac{M}{n}$. Thus, with probability at least $1 - \delta_3$,

$$\mathcal{R}(l \circ \mathcal{B}_0) \leq \mathcal{R}_S(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_3}}{2n}}. \quad (11)$$

In summary we have the following statements for the terms on the right hand side of (7):

1. With probability $1 - \delta_1$ over S, $l_S(\beta^*) - l_{\mathbb{P}}(\beta^*) \leq M\sqrt{\frac{\log \frac{1}{\delta_1}}{2n}}$ from (8).
2. With probability $1 - \delta_2$ over S,
$$\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta)) \leq 2\mathcal{R}(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_2}}{2n}},$$
where we have substituted the value of $\mathbb{E}[\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta))]$ from (10) into (9).
3. With probability $1 - \delta_3$ over S, $\mathcal{R}(l \circ \mathcal{B}_0) \leq \mathcal{R}_S(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_3}}{2n}}$ from (11).

Combining these gives us the following key lemma.

Lemma 4.6. *With probability at least $1 - \delta$,*

$$l_S(\beta^*) - l_S(\beta^{Alg}) \leq 2\mathcal{R}_S(l \circ \mathcal{B}_0) + 4M\sqrt{\frac{\log \frac{3}{\delta}}{2n}}. \quad (12)$$

Proof. Consider the three events:

$$E_1 = \left\{ l_S(\beta^*) - l_{\mathbb{P}}(\beta^*) \leq M\sqrt{\frac{\log \frac{1}{\delta_1}}{2n}} \right\}$$

$$E_3 = \left\{ \max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta)) \leq 2\mathcal{R}(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_2}}{2n}} \right\}$$

$$E_2 = \left\{ \mathcal{R}(l \circ \mathcal{B}_0) \leq \mathcal{R}_S(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_3}}{2n}} \right\}$$

We know that with probabilities $\delta_1, \delta_2, \delta_3$ over the random sample S, these events do not happen. Thus using the union bound,

$$\begin{aligned}
\mathbb{P}_S(\bar{E}_1 \cup \bar{E}_2 \cup \bar{E}_3) &\leq \mathbb{P}_S(\bar{E}_1) + \mathbb{P}_S(\bar{E}_2) + \mathbb{P}_S(\bar{E}_3) \\
&= \delta_1 + \delta_2 + \delta_3 \\
\Rightarrow \mathbb{P}_S(E_1 \cap E_2 \cap E_3) &\geq 1 - \delta_1 + \delta_2 + \delta_3.
\end{aligned}$$

Substituting $\frac{\delta}{3}$ for δ_1, δ_2 and δ_3 and using (7), we get the result as stated. □

The implication of this lemma is that the empirical risk for the 'best-in-class' function β^* is less than the right hand side quantities, all of which are computable. This implies that even though we do not know β^*, we know it belongs to our precursor uncertainty set with high probability. As the number of instances n increases, the size of the set decreases and we are more sure that we have captured β^* within our precursor uncertainty set.

We need one more lemma that extends Equation (1) to when we have m simultaneous errors using a union bound argument.

Lemma 4.7. *With probability at least $1 - m\delta_e$ over m examples $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$,*

$$\max_{j=1,...,m} |\tilde{y}^j - \beta^*(\tilde{\mathbf{x}}^j)| \in E$$

Proof. From Equation (1),

$$\mathbb{P}_{\tilde{\mathbf{x}}^j, \tilde{y}^j}(|\tilde{y}^j - \beta^*(\tilde{\mathbf{x}}^j)| \notin E) \leq \delta_e \;\; j = 1, ..., m$$

Summing up these probabilities give us an upper bound $m\delta_e$ on the probability that at least one of these errors is outside E. The complement of this event is the event where none of the m errors are outside E simultaneously. Computing the probability of this event with respect to the randomness of $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$ gives us the desired result. □

Coming back to proving Theorem 4.1, we now make the following observations:

- Using the definition of set $\mathcal{B}$ and Lemma 4.6, we see that $\beta^* \in \mathcal{B}$ with probability $1 - \delta$. This implies that with probability at least $1 - \delta$, $u_{\beta^*} \in \mathcal{U}_{\mathcal{B}}$.
- Using the definition of set $\mathcal{U}_{-\mathcal{B}}$, which is equal to E^m, and Lemma 4.7 we see that $u_{-\beta*} \in \mathcal{U}_{-\mathcal{B}}$ with probability at least $1 - m\delta_e$.

Robust optimal solution $\boldsymbol{\pi}^*$ of Equation (3) is robust to any element of $\mathcal{U} = \mathcal{U}_{\mathcal{B}} \times \mathcal{U}_{-\mathcal{B}}$. In particular, it is robust to the random vector $[\mathbf{u}_{\beta^*}^T \;\; \mathbf{u}_{-\beta^*}^T]^T$ generated by the "best in class" model β^* with high probability. That is, we can combine the observations above using a union bound to get a guarantee of robustness of π^* to unknown future situations determined by the unknown model β^*:

$$\mathbb{P}_{S,\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m} \left(F(\boldsymbol{\pi}^*, [\mathbf{u}_{\beta^*}^T \;\; \mathbf{u}_{-\beta^*}^T]^T) \in \mathcal{K}\right) \geq 1 - (\delta + m\delta_e).$$

This concludes the proof of Theorem 4.1. □

4.2 Using machine learning with a finite hypothesis set $\mathcal{B}_0$ to construct $\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$:

When $\mathcal{B}_0$ consists of a finite number of models, we can define $\mathcal{B}$ without using the notion of Rademacher averages. Let $|\mathcal{B}_0|$ represent the size of the set $\mathcal{B}_0$. Then we can define the set of good models as:

$$\mathcal{B} := \Bigg\{ \beta \in \mathcal{B}_0 : \\ l_S(\beta) \leq l_S(\beta^{Alg}) + M\sqrt{\frac{\log|\mathcal{B}_0| + \log\frac{2}{\delta}}{2n}} + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}} \Bigg\}, \tag{13}$$

where $n, \delta, M, l_S(\cdot)$ and β^{Alg} are the same as before.

Theorem 4.8. *For finite $\mathcal{B}_0$, the conclusion of Theorem 4.1 holds if $\mathcal{U}_{\mathcal{B}}$ is defined using $\mathcal{B}$ described in Equation (13).*

Proof. It is sufficient to show is that with probability at least $1 - \delta$, $\beta^* \in \mathcal{B}$ where $\mathcal{B}$ is defined in Equation (13). To see this, consider the second term of Equation (7) again. It is a function of random sample S. We can bound the probability of the event $\{\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta)) > \epsilon\}$ as follows:

$$\begin{aligned} &\mathbb{P}_S\Big(\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta)) > \epsilon\Big) \\ &\quad = \mathbb{P}_S\Big(\cup_{i=1}^{|\mathcal{B}_0|} \{l_{\mathbb{P}}(\beta^i) - l_S(\beta^i) > \epsilon\}\Big) \\ &\quad \overset{(a)}{\leq} \sum_{i=1}^{|\mathcal{B}_0|} \mathbb{P}_S\Big(l_{\mathbb{P}}(\beta^i) - l_S(\beta^i) > \epsilon\Big) \\ &\quad \overset{(b)}{=} \sum_{i=1}^{|\mathcal{B}_0|} e^{-\frac{2n\epsilon^2}{M^2}} = e^{\log|\mathcal{B}_0| - \frac{2n\epsilon^2}{M^2}}. \end{aligned}$$

Here, (a) follows from taking a union bound, and (b) follows from applying Hoeffding's inequality to each fixed model $\beta^i, i = 1, ..., |\mathcal{B}_0|$. Setting $\delta_2 = e^{\log|\mathcal{B}_0| - \frac{2n\epsilon^2}{M^2}}$ and replacing ϵ gives us the following equivalent way to state the same result: with probability at least $1 - \delta_2$ over S,

$$\max_{\beta \in \mathcal{B}_0}(l_{\mathbb{P}}(\beta) - l_S(\beta)) \leq M\sqrt{\frac{\log|\mathcal{B}_0| + \log(\frac{1}{\delta_2})}{2n}}.$$

From Equation (8), we have the following statement for the first term on the right hand side of (7): with probability $1 - \delta_1$ over S, $l_S(\beta^*) - l_{\mathbb{P}}(\beta^*) \leq M\sqrt{\frac{\log\frac{1}{\delta_1}}{2n}}$.

Using a union bound with these two observations about the first and second terms of Equation (7) gives us the following statement when $\delta_1 = \delta_2 = \delta/2$: with probability at least $1 - \delta$ over S, $l_S(\beta^*) - l_S(\beta^{Alg}) \leq M\sqrt{\frac{\log|\mathcal{B}_0| + \log(\frac{2}{\delta})}{2n}} + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}$. Thus $\beta^* \in \mathcal{B}$ with probability $1 - \delta$ as desired. □

4.3 Using PAC-Bayes theory for classification to construct $\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$:

If the learning step is a classification task, we can also define $\mathcal{B}$ using the PAC-Bayes framework [10] where PAC means "probably approximately correct". An important distinction of this framework is that it does not seek a single empirically good classifier β^{Alg} and instead the objective is to find a good "posterior" distribution Q over the hypothesis set $\mathcal{B}_0$. The theory provides a probabilistic guarantee that holds uniformly over all posterior distributions. The framework then picks a Q using data S so that the corresponding a Q-weighted deterministic classifier (or a Q-based randomized classifier) has the optimal probabilistic guarantee.

In particular, consider the Q-based Gibbs classifier G_Q, which makes each prediction by choosing a classifier from $\mathcal{B}_0$ according to Q. Let the Q-based Gibbs classifier have the following risks: (a) expected risk $R(G_Q) := \mathbb{E}_{\beta \in Q}[l_{\mathbb{P}}(\beta)]$,

and (b) empirical risk $R_S(G_Q) := \mathbb{E}_{\beta\in Q}[l_S(\beta)]$ where $l_{\mathbb{P}}(\beta)$ and $l_S(\beta)$ were defined previously. The PAC-Bayes framework guarantees that for all Q, $R(G_Q)$ is bounded by $R_S(G_Q)$ and a term which captures the deviation of Q from a pre-specified 'prior' distribution P over $\mathcal{B}_0$.

Theorem 4.9. *(Theorem 2.1 [11]) Let $l(\beta(\mathbf{x}), y) := \mathbf{1}[\beta(\mathbf{x}) \neq y]$. For any $\mathbb{P}_{\mathbf{x},y}$, any $\mathcal{B}_0$, any prior P on $\mathcal{B}_0$, any $\delta \in (0,1]$ and any convex function $\mathcal{D} : [0,1]^2 \to \mathbb{R}$, we have*

$$\mathbb{P}_S\left(\forall Q \text{ on } \mathcal{B}_0 : \mathcal{D}(R_S(G_Q), R(G_Q)) \leq \frac{1}{n}\left[KL(Q||P) + \log\left(\frac{1}{\delta}\mathbb{E}_S\mathbb{E}_{\beta\sim P}e^{m\mathcal{D}(l_S(\beta),l_S(\beta))}\right)\right]\right) \geq 1-\delta, \quad (14)$$

where $KL(Q||P) := \mathbb{E}_{\beta\sim Q}[\log\frac{Q(\beta)}{P(\beta)}]$.

For a certain choice of the metric $\mathcal{D}$ as shown in [11], the above theorem gives a bound on $R(G_Q)$ that is proportional to $CnR_S(G_Q) + KL(Q||P)$ where C is a pre-specified constant. We can minimize this quantity to get an optimal distribution Q^{Alg} with a closed form expression: $Q^{\text{Alg}}(\beta) = \frac{1}{Z}P(\beta)e^{-Cnl_S(\beta)}$ where Z is a normalizing constant.

Our construction of $\mathcal{B}$ for the model uncertainty set $\mathcal{U}_{\mathcal{B}}$ uses Q^{Alg} as follows:

$$\mathcal{B} = \left\{\beta \in \mathcal{B}_0 : l_S(\beta) \leq \frac{\log P(\beta) - \alpha}{nC}\right\},$$

where $\alpha > 0$ is a fixed constant, $P(\beta)$ is the prior probability density of model β, and C is a constant that appears in the objective when we solve for Q^{Alg}. Intuitively, the set $\mathcal{B}$ includes all models such that their empirical error is bounded appropriately using their scaled log prior density values. By our construction, if $\beta \in \mathcal{B}$, then $Q^{\text{Alg}}(\beta)$ is greater than the threshold $\frac{e^{\alpha}}{Z}$. There is no notion of a best-in-class model β^* in the PAC-Bayes setting and thus we do not have guarantees similar to the case where we used uniform convergence results to define $\mathcal{B}$. Nonetheless, $\mathcal{B}$ is data driven and captures those models which have a high posterior density in $\mathcal{B}_0$. $\mathcal{U}_B$ and $\mathcal{U}_{-\mathcal{B}}$ are defined using $\mathcal{B}$ and Equation (1) in the same way as before and used in the decision problem to obtain robust solution π^*.

4.4 Using linear regression to construct $\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$:

As suggested in [2] in the specific context of robust portfolio selection problems, we can assume distributional properties on $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$ (**Assumption 1**) in addition to assuming the functional form of the map $\mathbf{x} \mapsto \beta(\mathbf{x})$ (**Assumption 2**). In particular, let $y = \beta(\mathbf{x}) + \epsilon$ where $\beta(\mathbf{x}) = \boldsymbol{\beta}^T\mathbf{x}$ be the functional form of the model. Let us also assume that $\mathbf{x}$ is not necessarily random. The only source of randomness is through ϵ which is independent from example to example and is distributed according to $\mathcal{N}(0,\sigma^2)$. Then an estimator of $\boldsymbol{\beta}^*$ (the "best" model) is given by:

$$\boldsymbol{\beta}^{\text{Alg}} = (X^TX)^{-1}X^TY$$

where X is a matrix with n rows, one for each $\mathbf{x}^i$ and Y is an $n \times 1$ vector with the ith element being y^i. We assume that the rank of X is d. Substituting $Y = X\boldsymbol{\beta}^* + \boldsymbol{\epsilon}$ above gives us:

$$\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta}^* = (X^TX)^{-1}X^T\boldsymbol{\epsilon}$$

which is then distributed as $\mathcal{N}(0, \sigma^2(X^TX)^{-1})$. Thus, the real-valued function $g(\boldsymbol{\beta}^*, S) := \frac{1}{\sigma^2}(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta}^*)(X^TX)(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta}^*)$ is a χ^2_d distributed random variable. We can find a range such that with high probability the χ^2_d distributed random variable $g(\boldsymbol{\beta}^*, S)$ belongs to it. Choosing $\mathcal{B}$ based on this interval gives us an ellipsoid centered at $\boldsymbol{\beta}^{\text{Alg}}$ as follows:

$$\mathcal{B} = \left\{\boldsymbol{\beta} : \frac{1}{\sigma^2}(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta})(X^TX)(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta}) \leq c\right\}$$

where c is a constant that determines how much of the probability mass of χ^2_d is within the set $\mathcal{B}$.

Set $\mathcal{U}_{-\mathcal{B}}$ can be defined using our assumption about the model residuals: $\epsilon = (y - \boldsymbol{\beta}^T x) \sim \mathcal{N}(0,\sigma^2)$. In particular, using Equation (1), we get interval $E = [-e, e]$ for any desired value of δ_e by solving the equation:

$$\int_{-e}^{e} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{s}{2\sigma^2}} ds = 1 - \delta_e$$

Using $\mathcal{B}$ and $\mathcal{U}_{\mathcal{B}}$ as defined above in the robust problem of Equation (3) gives us a natural guarantee on the robustness of π^* to the "best in class" model $\boldsymbol{\beta}^*$.

If σ^2 is unknown, regression theory provides the following fix. We obtain an unbiased estimator of σ^2 given by $s^2 = \frac{\|Y - X\boldsymbol{\beta}^{\text{Alg}}\|_2^2}{n-d}$. The resulting scaled random variable $\frac{1}{ds^2}(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta})(X^TX)(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta})$ is $F-$distributed with d degrees of freedom in the numerator and $n-d$ degrees of freedom in the denominator [12]. We can again define $\mathcal{B}$ similarly to the previous case. The constant c now determines how much of the probability mass of an $F_{d,n-d}$-distributed random variable is within $\mathcal{B}$.

Note that both **Assumption 1** and **Assumption 2** (or their variations for similar models) are needed to justify this construction. Contrast this with the setting of Section 4.1 where a much weaker assumption was made. In the above illustration for multivariate regression, we assumed a fixed design ($\mathbf{x}^i$ were not random) whereas in the setting of Section 4.1 $\mathbf{x}^i$ are random.

5 Conclusion

In this paper, we considered a class of single-stage decision making problems where the uncertainty is derived from statistical modeling. We present a principled way to design uncertainty sets in the robust optimization framework for these problems using statistical learning theory. The core idea in the construction of the uncertainty sets is to construct $\mathcal{B}$ so that it contains the "best in class" model β^* with high probability. Then if we solve the optimization problem to be robust to all of $\mathcal{B}$, it will be robust to β^*.

References

[1] Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199–231, 2001.

[2] Donald Goldfarb and Garud Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*, 28(1):1–38, 2003.

[3] Vladimir N Vapnik. *Statistical learning theory*. Wiley, 1998.

[4] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

[5] Vishal Gupta et al. Data-Driven Robust Optimization. *Unpublished manuscript*, 2013.

[6] Abraham Charnes and William W Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.

[7] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

[8] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*, volume 23. Springer, 1991.

[9] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[10] David A McAllester. PAC-Bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pages 164–170. ACM, 1999.

[11] Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 353–360. ACM, 2009.

[12] Theodore Wilbur Anderson. *An introduction to multivariate statistical analysis*, volume 2. Wiley New York, 1958.