Responsible AI
10716: Advanced Machine Learning
Pradeep Ravikumar

1 Preliminaries

Consider the supervised learning setting, with input random variable $X \in \mathcal{X} \subseteq \mathbb{R}^d$, output random variable $Y \in \mathcal{Y}$, and observations $S = \{(x_i, y_i)\}_{i=1}^n$ drawn from a distribution P_{data} over $\mathcal{X} \times \mathcal{Y}$. Let $\widehat{P}_{\text{data}}$ denote the empirical distribution over the samples. We also have a set H of hypothesis functions $h : \mathcal{X} \mapsto \mathcal{Y}$ from which we wish to learn the best predictor. We evaluate the goodness of a predictor via a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$, with true risk:

$$R(h) = \mathbb{E}_{P_{\text{data}}} \ell(h(x), y),$$

which we can approximate via the empirical risk (since it is decomposable):

$$\widehat{R}(h) = \mathbb{E}_{\widehat{P}_{\text{data}}} \ell(h(x), y),$$

where

$$\mathbb{E}_{\widehat{P}_{\text{data}}}(f(x, y)) = \frac{1}{n} \sum_{i=1}^{n} f(x_i, y_i).$$

In the risk definitions above, we suppress the dependence on the underlying data distribution (which is the underlying state of nature), and our goal is to extract the functional of the state of the nature as captured by the minimizer of the true risk. And since the true risk is decomposable, we can approximate the true risk minimizer pointwise (that is, for any underlying data distribution) by minimizing the empirical risk.

So far so good, but what if the decision-theoretic setup above does not fully capture what we want to extract from the state of nature (i.e. the underlying data distribution)? Say we do get a hypothesis h that has low expected risk. But does it also have low tail risk: meaning that the risk is not just low in expectation, but also some higher-level quantile of the risk is low? We might also want h to be robust to distribution shift, fair, resistant to adversarial attacks, robust in the presence of outliers, to name a few additional desiderata. None of these are changing the loss function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ used above, but what they do require is changing how we define our risk. It requires re-examining our decision-theoretic fundamentals (that we largely took for granted in recent years in ML).

2 Responsible AI Desiderata

It turns out that each of these additional requirements do not have a unique characterization, and there are rich sub-fields dedicated to each of these requirements. These sub-fields are sometimes collated under the umbrella of "Fairness, Accountability, and Transparency (FAccT)" and more recently as "responsible AI". (That means that anything outside that umbrella is possibly irresponsible AI!) Many organizations are increasingly advocating the use of responsible AI models [Microsoft, 2021, Google, 2020].

Let us consider each of these separately.

2.1 Distribution shift

In distribution shift, the "test" distribution with respect to which the test risk will be defined is different from the "train" distribution P_{data} available to us at training. This immediately poses a challenge to the typical approach of minimizing true risk with respect to P_{data}. We should be minimizing the true risk with respect to the test distribution instead! The problem being we typically do not have access to the test distribution even in the form of samples. Obviously if the test distribution can be arbitrarily specified, bounding the test risk is impossible in general, so researchers have formalized several possible restrictions. This setting is broadly referred to as out-of-distribution (OOD) generalization, and was classically explored in a setting where there is a single "source" training distribution and a different "target" test distribution. There has been considerable recent interest in moving beyond a single source distribution, instead assuming that the set of training data is comprised of a collection of "environments" [Blanchard et al., 2011, Muandet et al., 2013, Peters et al., 2016] or "groups" [Hu et al., 2018, Duchi et al., 2019, Sagawa et al., 2020], each representing a distinct distribution, where the group identity of each sample may be known. Such a setting is referred to as *domain generalization*. The hope is that by cleverly training on such a collection of groups, one can derive a robust predictor which will better transfer to unseen test data which relates to the observed distributions.[1] A critical question then is how future test distributions depend on these groups and how to learn predictors with good performance on these distributions.

To set things up, suppose the training set comprises a set of distinct domains $\mathcal{E} = \{e_i\}_{i=1}^{E}$, each of which indexes a probability distribution p^e, and that the test environment will relate to these domains in some pre-specified way. Let us denote the set of such possible test distributions by $\mathbb{E}_{\text{test}}$. We can then cast the learner's goal as minimizing the worst-case error over the possible test distributions $\mathbb{E}_{\text{test}}$. For a set of predictors $\mathcal{F}$ and loss ℓ, our goal is thus to solve the objective

[1] Throughout this work, we use the terms "domain", "distribution", and "environment" interchangeably.

$$\min_{f \in \mathcal{F}} \max_{e \in \mathbb{E}_{\text{test}}} \mathbb{E}_e[\ell(f)].$$

Given a collection of environments, there are many possible ways to specify $\mathbb{E}_{\text{test}}$. One natural choice is $\mathbb{E}_{\text{test}} = \mathcal{E}$, so that the test environment could be an adversarially chosen single environment (i.e. the environment can pick that one single environment where the model performs the worst.) The min-max objective in that case is known as *Group DRO* Duchi et al. [2019], Sagawa et al. [2020]. This also arises in contexts of fairness as we will see in the sequel.

Another natural choice is the set of all convex combinations (i.e., mixtures) of source environments:

$$p^{\lambda} := \sum_{e \in \mathcal{E}} \lambda_e p^e, \tag{1}$$

where $\lambda \in \Delta_E$ is a vector of convex coefficients (Δ_E is the $(E-1)$-simplex). We will denote this convex hull $\text{Conv}(\mathcal{E})$.

This min-max objective is mathematically equivalent to Group DRO. This is because for any predictor, the optimal choice for the adversary will be whichever training environment produces the highest risk; that is, the adversary will *always* play a vertex of the simplex, so the region $\mathbb{E}_{\text{test}}$ remains discrete. It is easy to see the equivalence of interpolation and the discrete game:

$$\min_{f \in \mathcal{F}} \max_{e \in \text{Conv}(\mathcal{E})} \mathbb{E}_e[\ell(f)] \;=\; \min_{f \in \mathcal{F}} \max_{e \in \mathcal{E}} \mathbb{E}_e[\ell(f)].$$

We note that in some prior work on Group DRO, learning models that minimize worst-case sub-population risk is indeed the goal—that is, they only care about test domains that match one of the source domains. In the broader domain generalization literature, however, they do consider this form of interpolation, but as we can see from the above, and as also shown in recent work, this does provides any additional constraint beyond group DRP on OOD learning without additional regularization [Hu et al., 2018].

We can also cast the task of domain generalization as a continuous game of online learning in which the player is presented with sequential test domains and must refine their predictor at each round. We're therefore interested in the player's ability to *learn continuously* and improve in each round. We would expect that any good learning algorithm will suffer less per distribution as we observe more of them—that is, the *per-round regret* should decrease over time. Specifically, we'd like to prove a rate at which our regret goes down as a function of the number of distributions we've observed.

The full game is described in Algorithm 2.1.

Game Setup Before the game begins, we define a family of predictors $f \in \mathcal{F}$. For some observation space $\mathcal{X}$ and label space $\mathcal{Y}$, nature provides a fixed loss function $\ell : \mathcal{F} \times (\mathcal{X} \times \mathcal{Y}) \to$

Algorithm 1 : Domain Generalization Game (likelihood reweighting)

Input: Convex parameter space B, distributions $\{p^e\}_{e\in\mathcal{E}}$ over $\mathcal{X}\times\mathcal{Y}$, loss $\ell : \mathcal{F}\times(\mathcal{X}\times\mathcal{Y}) \to \mathcal{R}$.

for $t = 1 \dots T$ **do**

1. Player chooses hypothesis $f_t \in \mathcal{F}$.
2. Adversary chooses coefficients $\lambda_t \in \Delta_{|\mathcal{E}|}$.
3. Define $L_t(f) := \mathbb{E}_{(x,y)\sim p^{\lambda_t}}[\ell(f,(x,y))] = \sum_{e\in\mathcal{E}} \lambda_{t,e}\mathbb{E}_{(x,y)\sim p^e}[\ell(f,(x,y))]$.

end for

Player suffers regret

$$R_T = \sum_{t=1}^{T} L_t(f_t) - \min_{f\in\mathcal{F}} \sum_{t=1}^{T} L_t(f).$$

$\mathcal{R}$, as well as a set of E environments $\mathcal{E} = \{e_i\}_{i=1}^{E}$, each of which indexes a distribution p^e over $\mathcal{X}\times\mathcal{Y}$. The game proceeds as follows:

On round t, the player chooses parameters $f_t \in \mathcal{F}$. Next, the adversary chooses a set of coefficients $\lambda_t := \{\lambda_{t,e}\}_{e\in\mathcal{E}}$, which defines the distribution p^{λ_t} as the weighted combination of the distributions of the environments in $\mathcal{E}$ with coefficients λ_t, as in Equation 1. We assume that every choice of λ by the adversary is a set of convex coefficients, which ensures that p^{λ_t} is a valid probability distribution. At the end of the round, the player suffers loss $L_t(f_t) = \mathcal{R}^{\lambda_t}(f_t)$, defined as the risk of the predictor f_t on the adversary's chosen distribution:

$$\mathcal{R}^{\lambda_t}(f) := \mathbb{E}_{(x,y)\sim p^{\lambda_t}}[\ell(f,(x,y))]$$

As in standard online learning, our goal is to minimize *regret* with respect to the best fixed predictor in hindsight after T rounds. That is, we hope to minimize

$$\sum_{t=1}^{T} L_t(f_t) - \min_{f\in\mathcal{F}} \sum_{i=1}^{T} L_t(f). \tag{2}$$

As we saw in the learning and games lectures, we can achieve sub-linear regret using FTRL strategies (if the losses L_t are convex, if not, we can achieve sub-linear expected regret via FTRL in probability space). Recall that this takes the form:

$$f_t = \arg\min_{f\in\mathcal{F}} \sum_{s=1}^{t-1} L_s(f) + \eta R(f),$$

for some strongly convex regularizer $R(\cdot)$. It is instructive to see the strong similarity of this to regularized pooled ERM: where we pool the data from all environments together and learn a pooled model via ERM. Here, instead of individual environments, we pool a set of adversarially chosen interpolated distributions p^{λ_t}.

2.2 Sub-population Shift

A particular instance of the above is where the training data is a specific mixture of some sub-populations, so that

$$P_{\text{data}} = p_{\lambda_{\text{train}}},$$

and where $\{p^e\}_{e \in \mathcal{E}}$ are the sub-population distributions. This arises naturally in fairness [Hashimoto et al., 2018, Hu et al., 2018, Sagawa et al., 2019, Zhai et al., 2021], where we want to ensure that the performance on the worst-performing group is maximized. This is a specific notion of fairness known as minimax group fairness. There are many other notions of fairness such as individual fairness [Dwork et al., 2012, Zemel et al., 2013], group fairness notions such as Demographic Parity, Equality of Odds, Equality of Opportunity [Hardt et al., 2016, Zafar et al., 2017], counterfactual fairness [Kusner et al., 2017], and Rawlsian max-min fairness [Rawls, 2020, Hashimoto et al., 2018], among others [Barocas et al., 2017, Chouldechova and Roth, 2018, Mehrabi et al., 2021]. It's an interesting open problem to cast all of these disparate notions under one umbrella.

Another application of sub-population shift is learning on class-imbalanced datasets [Cao et al., 2019, Menon et al., 2021, Kini et al., 2021]. Suppose there are K classes, then we would have the K sub-populations: $p(X|Y = e)$, for $e \in [K]$. Let us define the class-conditional risk:

$$R_e(f) = \mathbb{E}_{X \sim p^e}[\ell(f(X), e),$$

given a classifier $f : \mathcal{X} \mapsto [K]$ and some loss function $\ell : [K] \times [K] \mapsto \mathbb{R}$ (for instance the zero-one loss). It can be seen that the usual expected risk is given as:

$$R(f) = \sum_{e \in [K]} p_e R_e(f),$$

where $p_e = P[Y = e]$. But this is not as useful a measure when some of the classes are imbalanced. If p_e is relatively small, $R(f)$ is not going to be affected a lot by $R_e(f)$. In such a case, we might be more interested in the worst case risk:

$$\max_{e \in [K]} R_e(f),$$

which however need not correspond to the Bayes optimal classifier (which minimizes the expected risk).

2.3 Distributionally Robust Optimization (DRO)

Suppose the samples we see are not coming from the "true" distribution, but from a noisy variant of it. In that case, a natural estimator would not minimize expected risk but take this noisiness of the distribution into account. A natural approach is what is known as

Distributionally Robust Optimization (DRO). Suppose P is the noisy distribution to which we have sampling access, but is not the true distribution Q of which all we know is that it satisfies $D(Q, P) \leq \rho$, for some divergence D. Then DRO computes the worst-case risk over such plausible true distributions:

$$\max_{\{} Q : D(Q, P) \leq \rho\} \mathbb{E}_{Z \sim Q} \ell(f, Z),$$

DRO has been studied under various uncertainty sets including f-divergence based uncertainty sets [Namkoong and Duchi, 2017, Duchi and Namkoong, 2018, Sagawa et al., 2019], Wasserstein uncertainty sets [Sinha et al., 2017, Gao et al., 2022], Maximum Mean Discrepancy uncertainty sets [Staib and Jegelka, 2019], more general uncertainty sets in the RKHS space [Zhu et al., 2020].

2.3.1 Tail Risk

In expected risk, we care about the predictor that performs well on average. Suppose I tell you that one pond you are thinking of crossing is 2 feet deep on average, and suppose you don't know how to swim. And another pond is 3 feet deep on average. Does this tell you enough about which pond to cross without getting drowned? No, it doesn't. Because what you care about how deep the pond is at its deepest, not its average depth. Similarly, in high-stakes settings, we may care about the model performance in the the tails of the risk distribution.

A popular choice, known as Conditional Value at Risk (CVAR) is to focus on the worst-alpha proportion of the data:

$$\max_{Q: P = \alpha Q + (1-\alpha) Q'} R(f, Q).$$

Let us consider the sample setting where P is simply the uniform distribution over the n training samples S. In such a case, we can write the above as:

$$\max_{w: \|w\|_0 \leq n\alpha, w \in \Delta_n} \sum_{i=1}^{n} w_i \ell(f, Z_i).$$

This in turn is commonly relaxed to the convex set:

$$\max_{w: \|w\|_i nfty \leq 1/(n\alpha), w \in \Delta_n} \sum_{i=1}^{n} w_i \ell(f, Z_i),$$

and is in fact also commonly referred to as the CVAR risk.

2.4 Boosting

Recall the boosting game, where we aimed to solve for the classifier f that minimizes the max loss over distributions over the n training samples S:

$$\max_{w \in \Delta_n} \sum_{i=1}^{n} w_i \ell(f, Z_i).$$

The problem with this is that if there is a single noisy sample, or a single sample where the predictor f is bad, we are giving it the maximum possible risk. This is too conservative. One way to address is a softening of this:

$$\max_{w \in \Delta_n : KL(w \| \text{Unif}([n])) \leq \rho_n} w_i \ell(f, Z_i).$$

2.5 Adversarial Robustness

It is natural to expect that if we perturb the test input by an infinitesimal amount, for instance by changing a few pixels in an input image, the output of the model should not change. This however turned to not be the case for many state of the art deep neural network models [Goodfellow et al., 2015, Szegedy et al., 2013], which led to a decade long quest to learn models that are robust to such perturbations. This is referred to as adversarial robustness, since the perturbations can be adversarially chosen. We can cast the learning of adversarially robust models as a two player game: the adversary outputs a perturbation function that maps each data point to a perturbation, and the learner selects a model. Alternatively, we aim to minimize the adversarial risk:

$$\mathbb{E}_{(X,Y) \sim P} \max_{X' : d(X,X') \leq \epsilon} \ell(f(X'), Y)),$$

for some perturbation distance $d : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. It turns out that this can be compactly stated as:

$$\mathbb{E}_{Q : W_d(P,Q) \leq \epsilon} \mathbb{E}_{(X,Y) \sim Q} \ell(f(X), Y)),$$

where W_d is the 1-Wasserstein distance with cost d.

3 RAI Risk

Our development here follows Gupta et al. [2023]. Given a set of samples $\{(x_i, y_i)\}_{i=1}^n$, we define the class of empirical RAI risks (for Responsible AI risks) as: $\widehat{R}_{W_n}(h) = \sup_{w \in W_n} \mathbb{E}_w(h(x), y)$,

where $W_n \subseteq \Delta_n$, is some set of sample weights (a.k.a uncertainty set), and $\mathbb{E}_w(f(x,y)) = \sum_{i=1}^n w_i f(x_i, y_i)$.

It can be seen that the examples we discussed above are all instances of RAI risks, for different uncertainty sets W_n.

Given the empirical RAI risk $\widehat{R}_{W_n}(h)$ of a hypothesis, and set of hypotheses H, we naturally wish to obtain the hypothesis that minimizes the empirical RAI risk: $\min_{h \in H} \widehat{R}_{W_n}(h)$. This can be seen as solving a zero-sum game.

Definition 1 (RAI Games) *Given a set of hypothesis H, and a RAI sample weight set W_n, the class of RAI games is given as:* $\min_{h \in H} \max_{w \in W_n} \mathbb{E}_w(h(x), y)$.

We now know that in many high-stakes settings, what we want to minimize is not the expected risk but a RAI risk. Accordingly, there has been a lot of effort over the past decade in developing algorithms to solve such RAI games. Unfortunately the RAI game above need not have a Nash Equilibrium in general, and their min-max and max-min game values need not coincide, so that this is a difficult problem to solve. Particularly so when the hypothesis class is large, such as deep neural networks. Accordingly, there are heuristic approaches to solve these, which however turn out to not be as responsible as we might hope. This has seen the most development likely in adversarial robustness, where there are "defenses" which are heuristic approaches to solve the RAI game, and "attacks" which essentially provide a witness that they haven't actually solved the game well.

For example, Adversarial Training [Madry et al., 2018] (AT) is a notable technique that trains a robust model by two alternative steps: 1) finding adversarial examples of training data against the current model; 2) updating the model to correctly classify the adversarial examples and returning to step 1). This procedure can be connected to an alternating best-response strategy in the 2-player zero-sum RAI game. However, [REF] show that even in simple settings, the alternating best-response strategy may not converge.

What do we do? From the learning and games lecture, we know that one strategy is to solve the linearized problem instead. There is also a statistical, rather than purely computational reason to do so. Looking at the definition of the RAI game, good worst-case performance over the sample weight set W_n is generally harder, and for a simpler set of hypotheses H, there *may not exist* $h \in H$ that can achieve such good worst-case performance. Thus it is natural to consider deterministic ensemble models over H, which effectively gives us more powerful hypothesis classes.

Given a hypothesis class H, a **randomized ensemble** is specified by some distribution $Q \in \Delta_H$, and is given by: $\mathbb{P}[h_{\text{rand;Q}}(x) = y] = \mathbb{E}_{h \sim Q} \mathbb{I}[h(x) = y]$. Similarly, we can define its corresponding randomized ensemble RAI risk: $\widehat{R}_{\text{rand};W_n}(Q) = \max_{w \in W_n} \mathbb{E}_{h \sim Q} \mathbb{E}_w \ell(h(x), y)$.

We can then also define the class of ensemble RAI games:

Definition 2 (Randomized Ensemble RAI Games) *Given a set of hypothesis H, a RAI sample weight set W_n, the class of mixed RAI games is given as:*

$$\min_{Q \in \Delta_H} \max_{w \in W_n} \mathbb{E}_{h \sim Q} \mathbb{E}_w \ell(h(x), y). \tag{3}$$

This is a much better class of zero-sum games: it is linear in both the hypothesis distribution P, as well as the sample weights w, and if the sample weight set W_n is convex, is a convex-concave game. And under some mild conditions [REF], this game can be shown to have a Nash equilibrium (i.e. a mixed Nash equilibrium of the original RAI game).

One caveat with this is that in practice, the randomized ensemble risk is not what we will be evaluated by. In practice, we are expected to provide a deterministic prediction, and we will be evaluated by the loss of that prediction. In particular, when we talk about ensemble classifiers, we typically do not have a randomized ensemble in mind, rather what we are talking about is a deterministic ensemble.

Given a hypothesis class H, a deterministic ensemble is specified by some distribution $Q \in \Delta_H$, and is given by: $h_{\text{det;Q}}(x) = \arg\max_{y \in \mathcal{Y}} \mathbb{E}_{h \sim Q} \mathbb{I}[h(x) = y]$.

Alternative definitions for deterministic ensembles could be considered. For example, one could consider $h_{\text{det;Q}}(x) = \arg\min_{y \in \mathcal{Y}} \mathbb{E}_{h \sim Q} \ell(h(x), y)$. [Cotter et al., 2019, Wu et al., 2022] designed other more sophisticated strategies, but these are largely domain dependent. However the definition above is the most standard. For regression, a popular de-randomization strategy is to compute the expected prediction: $h_{\text{det;Q}}(x) = \mathbb{E}_{h \sim Q}[h(x)]$.

Correspondingly, we can write the deterministic ensemble RAI risk as $\widehat{R}_{W_n}(h_{\text{det;Q}}(x)) = \max_{w \in W_n} \mathbb{E}_w \ell(h_{\text{det;Q}}(x), y)$. This admits a class of deterministic RAI games:

Definition 3 (Deterministic Ensemble RAI Games) *Given a set of hypothesis H, a RAI sample weight set W_n, the class of RAI games for deterministic ensembles over H is given as:*

$$\min_{Q \in \Delta_H} \max_{w \in W_n} \mathbb{E}_w \ell(h_{det;Q}(x), y).$$

However, the aforementioned game is computationally less amenable because of the non-smooth nature of de-randomized predictions. Moreover, they need not have a Nash Equilibrium (NE), and in general, their min-max and max-min game values need not coincide. This poses challenges in solving the games efficiently. Which was the key reason we wanted to move to ensemble RAI games in the first place.

Interestingly, for the very specific case of binary classification, we can provide simple relationships between the risks of the randomized and deterministic ensemble.

Proposition 4 *Consider the setting with $\mathcal{Y} = \{-1, 1\}$, the zero-one loss ℓ, and $W_n = \Delta_n$. Then,*

$$\widehat{R}_{W_n}(h_{det;Q}) = \mathbb{I}[\widehat{R}_{W_n}(h_{rand;Q}) \geq 1/2].$$

Proof.

$$\begin{aligned}
\sup_{w \in \Delta_n} \widehat{\mathbb{E}}_w \mathbb{I}[h_{\mathrm{det;Q}}(x) \neq y] &= \sup_{i \in [n]} \mathbb{I}[y_i \neq \arg\max_{y \in \mathcal{Y}} \mathbb{E}_Q[h(x_i) = y]] \\
&= \mathbb{I}[\sup_{w \in \Delta_n} \mathbb{E}_w \mathbb{E}_Q \mathbb{I}[h(x) \neq y] \geq 1/2] \\
&= \mathbb{I}[\widehat{R}_{W_n}(h_{\mathrm{rand;Q}}) \geq 1/2]
\end{aligned}$$

as required. □ In this case, we can also relate the existence of a perfect deterministic ensemble ("boostability") to a weak learning condition on the set of hypotheses. Specifically, suppose H is boostable iff there exists $Q \in \Delta_H$ s.t. $\widehat{R}_{W_n}(h_{\mathrm{det;Q}}) = 0$. From the above proposition this is equivalent to requiring that $\widehat{R}_{W_n}(h_{\mathrm{rand;Q}}) < 1/2$. We thus obtain:

$$\inf_{Q \in \Delta_H} \sup_{w \in W_n} \mathbb{E}_{w,Q} \ell(h(x), y) < 1/2 \iff \sup_{w \in W_n} \inf_{h \in H} \mathbb{E}_w \ell(h(x), y) < 1/2$$

where the equivalence follows from the min-max theorem and the linearity of the objective in P. The last statement says that for any sample weights $w \in W_n$, there exists a hypothesis $h \in H$ that has w-weighted loss at most $1/2$. We can state this as a "weak-learning" condition on individual hypotheses in H. The above thus shows that for the specific case of $\mathcal{Y} = \{-1, 1\}$, the zero-one loss $\ell(y, y') = \mathbb{I}[y \neq y']$, and $W_n = \Delta_n$, we can relate boostability of H to a weak learning condition on hypothesis within H.

General Classification But in general, we do not have simple connections between $\widehat{R}_{W_n}(h_{\mathrm{det;Q}})$ and $\widehat{R}_{W_n}(h_{\mathrm{rand;Q}})$. All we can guarantee is the following upper bound:

Proposition 5 *Let $\gamma_Q = 1/\min_{i \in [n]} \max_{y \in \mathcal{Y}} \mathbb{P}_Q[h(x_i) = y]$. Then,*

$$\widehat{R}_{W_n}(h_{det;Q}) \leq \gamma_Q \widehat{R}_{W_n}(h_{rand;Q}).$$

Corollary 6 *For binary classification, we have $\gamma_P \leq 2$ and thus, we recover the well known bound $\widehat{R}_{W_n}(h_{det;Q}) \leq 2\widehat{R}_{W_n}(h_{rand;Q})$*

Proof. Denote $y_Q(x) = \arg\max_{y\in\mathcal{Y}} \mathbb{E}_Q(h(x) = y)$. Then,

$$\begin{aligned}
\widehat{R}_{W_n}(h_{\text{det;Q}}) &= \sup_{w\in W_n} \mathbb{E}_w \ell(y_Q(x), y) \\
&\leq \sup_{w\in W_n} \mathbb{E}_w \ell(y_Q(x), y) \frac{P_Q(h(x) = y_Q(x))}{1/\gamma_Q} \\
&\leq \gamma_Q \sup_{w\in W_n} \mathbb{E}_w \sum_{y'\in\mathcal{Y}} \ell(y', y) P_Q(h(x) = y') \\
&= \gamma_Q \sup_{w\in W_n} \mathbb{E}_w \mathbb{E}_Q \sum_{y'\in\mathcal{Y}} \ell(y', y) \mathbb{I}[h(x) = y'] \\
&= \gamma_Q \sup_{w\in W_n} \mathbb{E}_w \mathbb{E}_Q \ell(h(x), y) \\
&= \gamma_Q \widehat{R}_{W_n}(h_{\text{rand;Q}}),
\end{aligned}$$

as required. □

Note that these bounds might be loose in practice. Specifically, for the binary case, if $\widehat{R}_{W_n}(h_{\text{rand;Q}}) \leq \frac{1}{2}$ then we have $\widehat{R}_{W_n}(h_{\text{det;Q}}) = 0$. To this end, prior work [Lacasse et al., 2006, Germain et al., 2015, Masegosa et al., 2020] have developed tighter bounds using second-order inequalities. Note that these might suggest *second-order RAI games*, which might be a good course project for an intrepid team. As such, we can cast minimizing randomized RAI risk as minimizing an upper bound on the deterministic ensemble RAI risk. Thus, the corresponding randomized RAI game can be cast as a relaxation of the deterministic RAI game. In the sequel, we thus focus on this randomized ensemble RAI game, which we can then use to obtain a deterministic ensemble. Following the bounds above, the corresponding deterministic ensemble risk will be bounded by the randomized ensemble RAI risk.

4 Algorithms

For simplicity, assume H is a finite set, though the results extend to uncountable sets.

The first class of algorithms are game play based algorithms, where both the min and the max players are engaged in a repeated game against each other. Both players rely on low-regret algorithms to decide their next action. As we have seen, such a procedure converges to a mixed NE of the game Cesa-Bianchi and Lugosi [2006]. In the t^{th} round, the following distribution $w^t \in W$ is computed over the training data points (which is an Follow-The-Regularized-Leader (FTRL) update):

$$w^t \leftarrow \operatorname*{argmax}_{w\in W_n} \sum_{s=1}^{t-1} \mathbb{E}_w \ell(h^s(x), y) + \eta^{t-1} \text{Reg}(w) \tag{4}$$

Here, $\text{Reg}(\cdot)$ is a strongly concave regularizer and η^{t-1} is the regularization strength. One popular choice for $\text{Reg}(\cdot)$ is the negative entropy which is given by $-\sum_i w_i \log w_i$. This regularizer is also used by AdaBoost, which is a popular boosting algorithm. Below, we provide analytical expressions for w^t for various choices of $W_n, \text{Reg}(\cdot)$. Recall that the regularizer in the FTRL update ensures the stability of the updates; *i.e.,* it ensures consecutive iterates do not vary too much. This stability is naturally guaranteed when W_n is a strongly convex set (an example of a strongly convex set is the level set of a strongly convex function. Consequently, the regularization strength η^{t-1} could be set to 0 in this case, and the algorithm still converges to a NE [Huang et al., 2017].

- $W_n = \{\widehat{P}_{data}\}$ **(Empirical Risk Minimization)**

$$w^t \leftarrow \widehat{P}_{data}$$

- $W_n = \Delta_n$ **(Worst Case Margin)**

$$w^t \leftarrow \frac{u^t}{\|u^t\|_1} \quad \text{where} \quad u_i^t \leftarrow \exp\left(-\frac{\sum_{s=1}^{t-1} l(h^s(x_i), y_i)}{\eta^{t-1}}\right)$$

- $W_n = \{w : w \in \Delta_n, w \preceq \frac{1}{\alpha n}\}$ **(α-CVaR)**

$$w_i^t \leftarrow \min\left(\frac{1}{\alpha n}, \exp\left(-\frac{\sum_{s=1}^{t-1} l(h^s(x_i), y_i)}{\eta^{t-1}} - \lambda\right)\right) \quad \text{for } \lambda \quad \text{S.T. } \sum_i w_i^t = 1$$

- $W_n = \{w : D(w||\widehat{P}_{\mathbf{data}}) \leq \rho_n\}$ **(DRO)** For general f-divergences, there do not exist closed form updates for w^t. However, they can still be empirically solved using FW-like updates.

- $W_n = \{\widehat{P}_{\mathbf{data}}(G_1), \widehat{P}_{\mathbf{data}}(G_2), \ldots \widehat{P}_{\mathbf{data}}(G_K)\}$ **(Group DRO)**

$$w^t \leftarrow \frac{u^t}{\|u^t\|_1} \quad \text{where} \quad u_i^t \leftarrow \exp\left(-\frac{\sum_{s=1}^{t-1} \sum_{i \in G_k} l(h^s(x_i), y_i)}{\eta^{t-1} s_k}\right) \text{for } i \in G_k, s_k = |G_k|$$

Once we have w^t, a new classifier h^t is computed to minimize the weighted loss relative to w^t, and added to the ensemble. This update is called the Best Response (BR) update. Learning h^t in this way helps us fix past classifiers' mistakes, eventually leading to an ensemble with good performance.

Algorithm 2 Game play algorithm for solving mixed RAI game

Input: Training data $\{(x_i, y_i)\}_{i=1}^n$, loss function ℓ, constraint set W_n, hypothesis set H, strongly concave regularizer R over W_n, learning rates $\{\eta^t\}_{t=1}^T$

1: **for** $t \leftarrow 1$ to T **do**
2: **FTRL**: $w^t \leftarrow \text{argmax}_{w \in W_n} \sum_{s=1}^{t-1} \mathbb{E}_w \ell(h^s(x), y) + \eta^{t-1} \text{Reg}(w)$
3: **BR**: $h^t \leftarrow \text{argmin}_{h \in H} \mathbb{E}_{w^t} \ell(h(x), y)$
4: **end for**
5: **return** $P^T = \frac{1}{T} \sum_{t=1}^T w^t$, $Q^T = \text{Unif}\{h^1, \dots h^T\}$

Just as we saw in the boosting lecture notes, we can also take a purely optimization theoretic viewpoint to design algorithms for solving the mixed RAI game. Let $L(Q)$ denote the inner maximization problem of (3): $L(Q) \coloneqq \max_{w \in W_n} \mathbb{E}_{h \sim Q} \mathbb{E}_w \ell(h(x), y)$. When $L(Q)$ is smooth (this is the case when W_n is a strongly convex set), one could use Frank-Wolfe (FW) to minimize it. The updates of this algorithm are given by

$$Q^t \leftarrow (1 - \alpha^t) Q^{t-1} + \alpha^t G, \quad \text{where } G = \operatorname*{argmin}_Q \left\langle Q, \nabla_Q L(Q^{t-1}) \right\rangle.$$

Here, $\nabla_Q L(Q^{t-1}) = \text{argmax}_{w \in W_n} \mathbb{E}_{h \sim Q^{t-1}} \mathbb{E}_w \ell(h(x), y)$. This algorithm is known to converge to a minimizer of $L(Q)$ at $O(1/t)$ rate [Jaggi, 2013]. When $L(Q)$ is non-smooth, we first need to smooth the objective before performing FW. In this work we perform Moreau smoothing [Parikh et al., 2014], which is given by

$$L_\eta(Q) = \max_{w \in W_n} \mathbb{E}_{h \sim Q} \mathbb{E}_w \ell(h(x), y) + \eta \text{Reg}(w). \tag{5}$$

Here $\text{Reg}(\cdot)$ is a strongly concave regularizer. If $\text{Reg}(\cdot)$ is 1-strongly concave, it is well known that $L_\eta(Q)$ is $O(1/\eta)$ smooth. Once we have the smoothed objective, we perform FW to find its optimizer (see Algorithm 3 for pseudocode).

Relaxing the simplex constraint. We can obtain a slightly different algorithm by relaxing the simplex constraint on Q. Using Lagrangian duality we can rewrite $\min_{Q \in \Delta_H} L_\eta(Q)$ as the following problem for some $\lambda \in \mathbb{R}$

$$\min_{Q \succeq 0} L_\eta(Q) + \lambda \sum_{h \in H} Q(h).$$

One interesting observation is that when W_n is the entire simplex and when $\lambda = -1/2$, we recover the AdaBoost algorithm. Given the practical success of AdaBoost, we extend it to general W_n. In particular, we set $\lambda = -1/2$ and solve the resulting objective using greedy coordinate-descent. The updates of this algorithm are given in Algorithm 3.

Remark 7 *Algorithm 3 takes the step sizes $\{\alpha^t\}_{t=1}^T$ as input. In practice, one could use line search to figure out the optimal step-sizes, for better performance.*

Algorithm 3 Greedy algorithms for solving Equation (3)

Input: Training data $\{(x_i, y_i)\}_{i=1}^n$, loss function ℓ, constraint set W_n, hypothesis set H, strongly concave regularizer R over W_n, regularization strength η, step sizes $\{\alpha^t\}_{t=1}^T$

1: **for** $t \leftarrow 1$ to T **do**
2: $\quad G^t = \text{argmin}_Q \langle Q, \nabla_Q L_\eta(Q^{t-1}) \rangle$
3: $\quad$ **FW:** $Q^t \leftarrow (1-\alpha^t)Q^{t-1} + \alpha^t G^t$ / **Gen-AdaBoost:** $Q^t \leftarrow Q^{t-1} + \alpha^t G^t$
4: **end for**
5: **return** Q^T

References

Microsoft. Microsoft. https://www.microsoft.com/en-us/ai/responsible-ai, 2021. Accessed: Date Accessed.

Google. Google. https://ai.google/responsibility/responsible-ai-practices/, 2020. Accessed: Date Accessed.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems*, volume 24, pages 2178–2186. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/b571ecea16a9824023ee1af16897a582-Paper.pdf.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society*, 2016.

Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018.

John Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses for latent covariate mixtures. *arXiv preprint arXiv:2007.13982*, 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Tatsunori B. Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. *International Conference On Machine Learning*, 2018.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. Doro: Distributional and outlier robust optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12345–12355. PMLR, 18-24 Jul 2021. URL `https://proceedings.mlr.press/v139/zhai21a.html`.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

John Rawls. *A theory of justice: Revised edition*. Harvard university press, 2020.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness in machine learning. *Nips tutorial*, 1:2017, 2017.

Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54 (6):1–35, 2021.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in Neural Information Processing Systems*, 32:1567–1578, 2019.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

Ganesh Ramachandra Kini, Orestis Paraskevas, Samet Oymak, and Christos Thrampoulidis. Label-imbalanced and group-sensitive classification under overparameterization. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Hongseok Namkoong and John C Duchi. Variance-based regularization with convex objectives. *Advances in neural information processing systems*, 30, 2017.

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

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Rui Gao, Xi Chen, and Anton J Kleywegt. Wasserstein distributionally robust optimization and variation regularization. *Operations Research*, 2022.

Matthew Staib and Stefanie Jegelka. Distributionally robust optimization and generalization in kernel methods. *Advances in Neural Information Processing Systems*, 32, 2019.

Jia-Jie Zhu, Wittawat Jitkrittum, Moritz Diehl, and Bernhard Schölkopf. Kernel distributionally robust optimization. *arXiv preprint arXiv:2006.06981*, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Andrew Cotter, Maya Gupta, and Harikrishna Narasimhan. On making stochastic classifiers deterministic. *Advances in Neural Information Processing Systems*, 32, 2019.

Jimmy Wu, Yatong Chen, and Yang Liu. Metric-fair classifier derandomization. In *International Conference on Machine Learning*, pages 23999–24016. PMLR, 2022.

Alexandre Lacasse, François Laviolette, Mario Marchand, Pascal Germain, and Nicolas Usunier. Pac-bayes bounds for the risk of the majority vote and the variance of the gibbs classifier. *Advances in Neural information processing systems*, 19, 2006.

Pascal Germain, Alexandre Lacasse, Francois Laviolette, Mario Marchand, and Jean-Francis Roy. Risk bounds for the majority vote: From a pac-bayesian analysis to a learning algorithm. *arXiv preprint arXiv:1503.08329*, 2015.

Andrés Masegosa, Stephan Lorenzen, Christian Igel, and Yevgeny Seldin. Second order pac-bayesian bounds for the weighted majority vote. *Advances in Neural Information Processing Systems*, 33:5263–5273, 2020.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvári. Following the leader and fast rates in online linear prediction: Curved constraint sets and other regularities. *The Journal of Machine Learning Research*, 18(1):5325–5355, 2017.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pages 427–435. PMLR, 2013.

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

Yash Gupta, Runtian Zhai, Arun Suggala, and Pradeep Ravikumar. Responsible ai (rai) games and ensembles. *NeurIPS*, 2023.