CS272 - Theoretical Foundations of Learning, Decisions, and Games

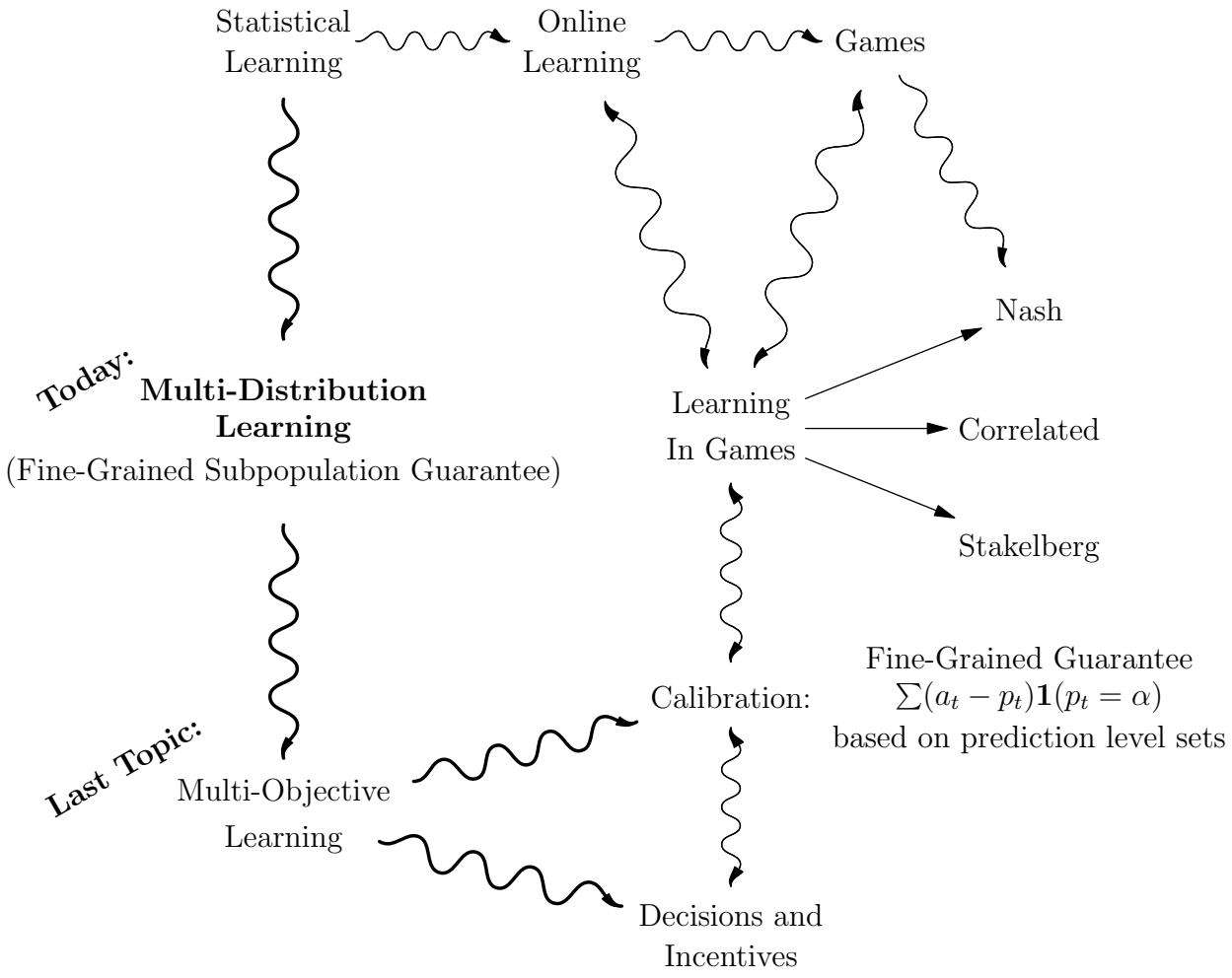
Lecture 23: Multi-distribution Learning I

November 20th, 2025

Lecturer: Nika Haghtalab *Readings: Haghtalab et al. [2022]*
Scribe: Andrew Huang and Anqi Lu

1 Semester Review and Last Module Preview

Today, we will begin the last module of the semester. Before we begin, we look back on the semester and discuss the connections between various modules:



To start off the semester, we discussed statistical learning, during which we looked at probability and performance (uniform convergence, sample complexity, algorithm complexity). Next,

we studied then online learning and how it's connected to statistical learning with methods such as multiplicative weights.

The next thing we studied was games, where natural equilibria connected to online learning; indeed we found that the bridge between games and online learning is very versatile. During this combined study of learning with games, we looked at multiple equilibria, including Nash, Correlated, and Stackelberg.

Next, we used Hart's minmax theorem to get many results in calibration. In fact, Hart's minmax theorem is useful in so many regards. Furthermore, this actually means something for decisions and incentives.

The thing is that calibration is a fine-grained guarantee. Generally we think about, on average, how much things disagree with each other. For calibration, we condition and care about performance on each of the level sets. Fine-grained guarantees are fundamental to machine learning, and this is what calibration does. Today, with our study of multi-distribution learning, we again study fine-grained guarantees, but now they depend not on predictor but rather the population. Given a risk predictor (i.e. cancer) for a population, if there are some meaningful minority subpopulations that have different socioeconomic needs, we want the predictor to each have a guaranteed rate of success for all of these subpopulations.

We'll spend 2 lectures on multi-distribution learning. Then, after Thanksgiving in the last week, we will see how this can allow us to hit multi-objective guarantees.

2 Multi-Distribution Learning: Definitions and Basics

2.1 Recall Statistical Learning Overview

Setup: Let $\mathcal{H}$ be a hypothesis class and $\mathcal{D}$ be the data distribution over $\mathcal{X} \times \mathcal{Y}$. We collect m samples $S \sim \mathcal{D}^m$ and learn a hypothesis h_S. The generalization error is defined as:

$$\mathrm{err}_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[h(x) \neq y].$$

Alternatively, this is often formulated as the expected loss:

$$\mathcal{L}_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[h(x) \neq y].$$

Realizable PAC Learning

- **Assumption:** There exists a perfect predictor $h^\star \in \mathcal{H}$ such that $\mathrm{err}_{\mathcal{D}}(h^\star) = 0$.
- **Goal:** Learn h such that $\mathrm{err}_{\mathcal{D}}(h) \leq \varepsilon$ with probability $1 - \delta$.
- **Sample Complexity:** $|S| = \tilde{O}\left(\frac{d+\log(1/\delta)}{\varepsilon}\right)$.

Agnostic Learning

- **Goal:** Learn h such that $\mathrm{err}_{\mathcal{D}}(h) \leq \min_{h^* \in \mathcal{H}} \mathrm{err}_{\mathcal{D}}(h^*) + \varepsilon$ with probability $1 - \delta$.
- **Sample Complexity:** $|S| = \tilde{O}\left(\frac{d+\log(1/\delta)}{\varepsilon^2}\right)$.

Recall our approach for tackling these problems:

Algorithm (ERM): We typically use Empirical Risk Minimization, which selects the hypothesis that minimizes error on the sample S:

$$h_S = \operatorname*{argmin}_{h \in \mathcal{H}} \mathrm{err}_S(h).$$

2.2 The Trouble with "Average" Guarantees

Let's say that we found a hypothesis $h \in \mathcal{H}$ such that the expected loss is $\mathcal{L}_{\mathcal{D}}(h) = 5\%$. Ideally, this would imply that for every $x \in \mathcal{X}$, there is a 5% chance of error. However, verifying such a pointwise guarantee is impossible with a finite number of samples.

Consequently, this average metric allows for a scenario where the error is 0 on 90% of the population, but 50% on the remaining 10%. This disparity is unsatisfying for many applications.

2.3 Multi-distribution Learning

Multi-distribution learning serves as a middle ground to reach a compromise between global averages and individual guarantees. It provides finer-grained guarantees on specific, pre-determined populations.

This was introduced by the work of Haghtalab et al. [2022], Blum et al. [2017] as a general framework encompassing several different considerations in ML. Formally, given sample access to distributions $\mathcal{D}_1, \ldots, \mathcal{D}_k$ (which are fixed before learning), we want to learn $h \in \mathcal{H}$ satisfying one of the following criteria:

1. **Realizable PAC setting:** We seek an h such that (assuming a consistent hypothesis exists):

$$\max_{i \in [k]} \mathcal{L}_{\mathcal{D}_i}(h) \leq \varepsilon.$$

2. **Agnostic setting:** We seek an h such that:

$$\max_{i \in [k]} \mathcal{L}_{\mathcal{D}_i}(h) \leq \min_{h^\star \in \mathcal{H}} \left(\max_{i \in [k]} \mathcal{L}_{\mathcal{D}_i}(h^\star) \right) + \varepsilon.$$

Note that if we define each $\mathcal{D}_i$ to be a single individual, this approach converges to the idealized case of individual guarantees. As a result, the parameter k plays a central role in the complexity of multi-distribution learning.

We now aim to ask the same fundamental questions about multi-distribution learning that we asked for standard statistical learning: what are the sample complexities and algorithmic paradigms for multi-distribution learning?

3 Sample Complexity and Algorithmic Paradigm

Setting	**Single Dist.**	**Multi-Distribution Attempts**		
$(m_{\delta,\varepsilon})$	$\tilde{O}(\cdot)$	**Naive (Attempt 1)**	**Attempt 2**	**Attempt 3**
PAC	$\frac{d+\log(1/\delta)}{\varepsilon}$	$\frac{kd}{\varepsilon}$	$\frac{(d+k)\log k}{\varepsilon^4}$	$\frac{(d+k)\log(k/\delta)}{\varepsilon}$
Agnostic	$\frac{d+\log(1/\delta)}{\varepsilon^2}$	$\frac{kd}{\varepsilon^2}$	—	$\frac{(d+k)\log(k/\delta)}{\varepsilon^2}$

A naive approach would be to collect a sufficient number of samples for every sub-distribution independently. Since standard sample complexity scales with roughly d/ε, this yields a total complexity of:

$$km_{\varepsilon,\delta} \approx \frac{kd}{\varepsilon}.$$

However, given that both the dimension d and the number of groups k can be large, this multiplicative bound leads to prohibitively high sample complexity.

Remarkably, we can improve upon this significantly. We will show that, up to logarithmic factors, the dependence on d and k can be **additive** rather than **multiplicative**—scaling with roughly $d + k$.

To understand why this is efficient, consider the cost of verification: if we were given a candidate hypothesis and asked to check if it satisfies our requirements (i.e., "are we happy?"), we would need approximately k/ε^2 samples to validate it across all groups. The fact that we can learn a hypothesis in the multi-distribution setting with an overhead that is effectively the cost of verification implies that we are doing extremely well; effectively, this suggests that as long as we can efficiently assess performance on each subpopulation, we can learn efficiently.

Today, we will focus on the **PAC** setting; on Tuesday, we will examine the **Agnostic** setting.

3.1 Attempt 1: The Naive Approach

Algorithm:

1. For each distribution $i \in [k]$, collect a dataset $S_i \sim \mathcal{D}_i^m$ with sample size $m = m_{\varepsilon,\delta/k}$.
2. Construct the union of all datasets $\overline{S} = \bigcup_{i\in[k]} S_i$.
3. Return the Empirical Risk Minimizer on the combined dataset:

$$h = \text{ERM}(\overline{S}).$$

Sample Complexity: The total sample complexity is the sum of samples across all k distributions:

$$|S| = k \times m_{\varepsilon,\delta/k} \approx \frac{k\left(d + \log\left(\frac{k}{\delta}\right)\right)}{\varepsilon}$$

Analysis: Why does this meet the multi-distribution PAC bound?

We start by assuming the realizability condition, meaning there exists a perfect hypothesis $h^\star$ such that $\mathcal{L}_{\mathcal{D}_i}(h^\star) = 0$ for all $i \in [k]$. Since the learner h minimizes error on the union $\overline{S}$ (and $h^\star$ achieves 0 error), h must achieve 0 error on $\overline{S}$. This implies zero error on every sub-sample, i.e.,

$$\mathcal{L}_{\overline{S}}(h) = 0 \implies \mathcal{L}_{S_i}(h) = 0, \quad \forall i \in [k].$$

Fixing a single distribution i, standard PAC analysis with confidence parameter $\delta' = \delta/k$ bounds the failure probability:

$$\mathbb{P}\left[\exists h \in \mathcal{H} \text{ s.t. } \mathcal{L}_{S_i}(h) = 0 \text{ but } \mathcal{L}_{\mathcal{D}_i}(h) \geq \varepsilon\right] \leq \frac{\delta}{k}.$$

Applying the Union Bound over all k distributions, we bound the probability that any distribution fails:

$$\mathbb{P}\left[\exists i \in [k] \text{ s.t. a bad event occurs}\right] \leq \sum_{i=1}^{k} \frac{\delta}{k} = \delta.$$

Thus, we conclude that with probability $\geq 1 - \delta$, the output hypothesis h satisfies $\mathcal{L}_{\mathcal{D}_i}(h) \leq \varepsilon$ for all $i \in [k]$ simultaneously.

3.2 An Additive Attempt

Recall the naive approach where we treated every distribution $\mathcal{D}_1, \ldots, \mathcal{D}_k$ independently. This ignores the fact that distributions often share structure. If two distributions are correlated, a hypothesis learned on one might effectively transfer to the other. Consequently, we should not need to pay the full sample complexity for both. But this structure is also unknown a priori so it must be learned as we sample. This suggests an adaptive strategy: rather than deciding sample sizes ahead of time, we should take a few samples to gauge current performance, identify which distributions are correlated, and adaptively shift our sampling budget toward the "hard" distributions that are not yet well-explained by our current hypothesis.

Before designing an adaptive algorithm, we explain to ourselves that a static approach is fundamentally limited. If we determine the sample sizes $m_1, \ldots, m_k$ before seeing the data (non-adaptive), we cannot exploit these correlations and are forced into the worst-case sample complexity. There is a theorem stating that any non-adaptive (non-interactive) sampling strategy incurs a sample complexity of $\tilde{\Omega}(dk/\varepsilon)$. This lower bound confirms that to improve upon the naive approach, the algorithm must be adaptive.

3.2.1 Multi-distribution as a Minmax Equilibrium:

We can frame the multi-distribution learning problem as finding an ε-approximate MinMax equilibrium. Formally, we seek a hypothesis h that satisfies:

$$\max_{i\in[k]} \mathcal{L}_{\mathcal{D}_i}(h) \leq \min_{h^*\in\mathcal{H}} \max_{i\in[k]} \mathcal{L}_{\mathcal{D}_i}(h^*) + \varepsilon.$$

We can visualize this interaction as a game played on a matrix. The rows represent the hypothesis class $\mathcal{H}$ (the minimizing player), and the columns represent the distribution indices $i \in [k]$ (the maximizing player). Each entry in this conceptual matrix corresponds to the loss $\mathcal{L}_{\mathcal{D}_i}(h)$.

Idea: Use No-Regret Dynamics. Drawing from the proof of the MinMax Equilibrium, we know that if one player uses a no-regret algorithm and the other plays either a no-regret algorithm or a Best Response strategy, the average history of play converges to the equilibrium.

Sample Complexity. A major challenge in this statistical setting is that the game utilities—the actual values of $\mathcal{L}_{\mathcal{D}_i}(h)$—are unknown. This introduces two distinct sources of sample complexity:

1. **Estimation Cost:** We must estimate the unknown matrix entries (utilities) using samples at every step of the game.

2. **Convergence Cost:** We must run the no-regret dynamics for enough timesteps to reach convergence, which scales according to the convergence rate ($\approx \text{Regret}/T$), as well as δ and ε.

3.2.2 Using No-Regret vs. Best Response Dynamics (Attempt 2)

We now aim to achieve a sample complexity of $\tilde{O}\left(\frac{d+k}{\varepsilon^4}\right)$. To do this, we set up a game where the **Maximizing Player** (the adversary choosing distributions) runs a **No-Regret Algorithm** (specifically Randomized Weighted Majority or RWM), and the **Minimizing Player** (the learner choosing hypotheses) plays a **Best Response Algorithm**.

Overview of the Algorithm: Let us give an overview of an algorithmic approach and why it works. In this section, we are intentionally high-level and will not give every detail of the algorithm or sample complexity. We will be working towards better algorithms in the next lecture which we will prove more rigorously.

For $t = 1, \ldots, T$ time steps:

1. **Maximizing Player (RWM/Hedge):** Maintains a distribution over the k sub-populations. Let w_i^t be the weight associated with distribution i at time t as does a no-regret algorithm. These weights are initially set to 1 and then updated.

   The mixture distribution is $p^t = \frac{1}{W_t}\sum_{i=1}^k w_i^t \mathcal{D}_i$ where $W^t = \sum_i w_i^t$. The weights are updated to focus on distributions where the current hypothesis performs poorly (high error). Using the exponential update rule with learning rate η:

   $$w_i^{t+1} \leftarrow w_i^t \exp(\mathcal{L}_{S_i}(h^t) \cdot \eta).$$

Since the true loss $\mathcal{L}_{\mathcal{D}_i}$ is unknown, we use the empirical loss $\mathcal{L}_{T_i}$ estimated from a test sample set T_i of size $\frac{\log(kT/\delta)}{\epsilon'^2}$ for each $\mathcal{D}_i$. For convince we assume to be able to get the same no-regret guarantees as RWM while having access only to the estimated losses.

2. **Minimizing Player (Best Response):** Given the adversary's mixture p^t, the learner chooses the hypothesis that minimizes the loss over this mixture. Since the true loss $\mathcal{L}_{p^t}$ is unknown, we run the empirical risk minimization algorithm on sample set S^t from p^t of size $m_{\epsilon',\delta/(kT)}$ and have
$$h^t = \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{L}_{S^t}(h)$$

Proof Sketch: We define our final hypothesis $\overline{h}$ as the uniform average over time (or approximately, a random draw from $h^1, \ldots, h^T$):

$$\overline{h} = \frac{1}{T} \sum_{t=1}^{T} h^t$$

We claim that for all $i \in [k]$, $\mathcal{L}_{\mathcal{D}_i}(\overline{h}) \leq \varepsilon$.

Note that by the ERM guarantees, with probability $1 - \delta$, at all time steps, we have that $\varepsilon' \geq \mathcal{L}_{p^t}(h^t)$. Then we have,

$$\varepsilon' \geq \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_{p^t}(h^t) \geq \max_{i \in [k]} \underbrace{\left(\frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_{\mathcal{D}_i}(h^t) \right)}_{= \mathcal{L}_{\mathcal{D}_i}(\bar{h})} - \underbrace{\sqrt{\frac{\log k}{T}}}_{\text{Regret Term}},$$

where the first inequality is due to the ERM, the second inequality is by the fact that the maximization place is no-regret, and the last term is the regret of RWM in the full-information setting.

By setting $T = \frac{\log k}{(\varepsilon')^2}$, the regret term becomes $\leq \varepsilon'$. Rearranging the inequality:

$$\max_{i \in [k]} \mathcal{L}_{\mathcal{D}_i}(\overline{h}) \leq 2\varepsilon'$$

By setting $\varepsilon' = \varepsilon/2$, we satisfy the error bound.

Sample Complexity Analysis: The total sample complexity is the sum of the samples needed for the two distinct parts of the algorithm over T timesteps.

1. **Estimation Cost (For RWM Update):** At every step t, we must estimate the loss of h^t on *each* of the k distributions to update the weights. Using Chernoff/Hoeffding bounds, and that T_i is of size $O(\ln(kT/\delta)/\varepsilon'^2)$ we can estimate $\mathcal{L}_{\mathcal{D}_i}(h^t)$ within tolerance ε' for all time steps and all distributions:
$$\text{Cost}_1 = T \times k \times \frac{\log(kT/\delta)}{\varepsilon'^2} \approx \frac{\log k}{\varepsilon'^2} \cdot \frac{k}{\varepsilon'^2} = \frac{k \log k}{\varepsilon'^4}$$

2. **Learning Cost (For Best Response):** At every step t, the learner needs samples from the mixture p^t to find the best response h^t. This is a standard PAC learning step.

$$\text{Cost}_2 = T \times m_{\varepsilon',\delta/(kT)} \approx \frac{\log k}{\varepsilon'^2} \cdot \frac{d + \log(k/\delta)}{\varepsilon'^2} = \tilde{O}\left(\frac{d \log k}{\varepsilon'^4}\right).$$

Total Complexity: Summing these up gives the final bound:

$$\tilde{O}\left(\frac{(d+k)\log k}{\varepsilon^4}\right).$$

3.3 Improvements and Next Lecture Preview

Next lecture, we will cover a third attempt at an algorithm, one that achieves a better complexity (a better exponent for ε in the denominator). There are a couple of places in our Attempt 2 where our argument has looseness. The first aspect we bring our attention to is the Hoeffding bound in assessing the test error. It turns out that we can use a Chernoff Multiplicative bound to shave off a factor of ε off: $1/\varepsilon^2 \to 1/\varepsilon$, since we are trying to estimate small losses that are close to 0.

Another major point of looseness comes from the regret term. Recall that we use RWM to get $\sqrt{T}$ regret when the best expert was not that good. But in settings where the best expert was perfect or really excellent, we could get away with no randomization and better regret bounds of the type we first encourated when studying Mistake Bound. In particular, this is quite similar to the our analysis of weighted majority algorithm for loss minimization where by setting $w_i^t \leftarrow 2w_i^{(t-1)}$ when expert i makes an error, the algorithm's loss is bounded by $2.4(\text{OPTLoss} + \log(k))$. We will see in more detail how this has to be adapted for our setting where the regret-minimization is done by the maximizing agent.

References

Avrim Blum, Nika Haghtalab, Ariel D Procaccia, and Mingda Qiao. Collaborative pac learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Nika Haghtalab, Michael Jordan, and Eric Zhao. On-demand sampling: Learning optimally from multiple distributions. *Advances in Neural Information Processing Systems*, 35:406–419, 2022.