CS272 - Theoretical Foundations of Learning, Decisions, and Games

Lecture 25: Omniprediction

December 2, 2025

Lecturer: Brian Lee *Readings: TBA*
Scribe: Nathaniel Macasaet

1 Multi-distribution Learning vs. Multi-objective Learning

1.1 Intuition and Definition

As a reminder, we define multi-distribution learning as follows.

Definition 1.1. A hypothesis h^* is a $(\mathcal{D}, \mathcal{H}, \epsilon)$ multi-distribution learning solution if

$$\max_{i\in[k]} \mathbb{E}_{\mathcal{D}_i}[l(y, h^*(x)] \leq \min_{h\in H} \max_{i\in[k]} \mathbb{E}_{\mathcal{D}_i}[l(y, h(x)] + \epsilon$$

for some set of distributions $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$.

In other words, we seek to find a single model h^* that performs well across multiple distributions to obtain robustness. To set up the intuition behind multi-objective learning, imagine that each distribution $\mathcal{D}_i$ has support $\mathcal{S}_i \times \mathcal{Y} \subset \mathcal{X} \times \mathcal{Y}$, with each $\mathcal{S}_i$ disjoint. Then, we can "break up" the loss function l into a set of loss functions $\mathcal{L} = \{l_1, \ldots, l_k\}$, where each $l_i = l(y, h(x)) \cdot \mathbb{1}\{x \in \mathcal{S}_i\}$.

You can imagine this example as "partitioning" the feature space $\mathcal{X}$, where we want to ensure that the hypothesis performs well on each partition. Therefore, while multi-distribution learning seeks performance across varying *distributions*, multi-objective learning seeks performance across varying *loss functions*.

Example 1.2. *We can have different loss functions.*

$$l_1(y, h(x)) = \mathbb{1}\{y \neq h(x)\}$$

$$l_2(y, h(x)) = (y - h(x))^2$$

$$l_3(y, h(s)) = \begin{cases} \alpha(y - h(x)) & y \geq h(x) \\ (1-\alpha)(h(x) - y) & y < h(x) \end{cases}$$

With this intuition set up, we define multi-objective learning as follows.

Definition 1.3. A hypothesis h^* is a $(\mathcal{L}, \mathcal{H}, \epsilon)$ multi-objective learning solution if

$$\max_{j\in[r]} \mathbb{E}_D[l_j(y, h^*(x))] \leq \min_{h\in H} \max_{j\in[r]} \mathbb{E}[l_j(y, h(x))] + \epsilon$$

The essential goal is to find a hypothesis h^* that performs well on every single loss function in our set $\mathcal{L}$. In this sense, both MDL and MOL attempt to establish some notion of universality.

1.2 Motivation

Machine learning problems tend to follow one common objective.

$$\min_{h \in \mathcal{H}} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} [l(y, h(x))]$$

When solving this problem, we as the learner have to choose both a loss function l and a distribution $\mathcal{D}$, which raises some things to worry about. In particular,

1. *Am I choosing the correct optimization problem?* This primarily comes from our choice of the loss function l. By fixing our choice of l, we could accidentally be molding our learned hypothesis h^* to perform very well with respect to l, but poorly with respect to any other loss function.

2. *Am I modeling the distribution of the data appropriately?* This primarily comes from our choice of the distribution $\mathcal{D}$. Choosing the incorrect distribution $\mathcal{D}$ could cause the expectation of the loss to change, therefore changing our learned hypothesis h^*.

Both MDL and MOL attempt to play around with these. They essentially want a hypothesis that is robust to multiple different optimization objectives or distributions.

1.3 Advantages of MOL

Imagine that you are handed a problem that is very hard, but in particular, it is very hard because it establishes universality in some way. If you can massage this problem into something that looks like MOL, there are two benefits:

1. We can view the problem as solving a two-player zero-sum game, just as we saw with MDL.

2. Because two-player zero-sum games can be solved using no-regret dynamics, you automatically get an algorithm for solving the original problem as well.

While this is very strong, there are some differences between the two-player zero-sum game analogies of MDL and MOL. In MOL, if you are able to sample from $\mathcal{D}$ with any fixed loss function l, then you not only obtain information on how h^* performs on l, but also how h^* performs against all other losses. Therefore, the adversary gets full feedback in this model.

2 Omniprediction

We begin discussing *omniprediction*. To start, we need to establish objects and assumptions.

2.1 Objects and Assumptions

- We have features $\mathcal{X}$, which have no structural assumptions.
- We have labels $\mathcal{Y} = \{0, 1\}$, which we take to be binary.
- We have a joint distribution $\mathcal{D}$.
- We have a competitor hypothesis class $\mathcal{H} \subset \{h : \mathcal{X} \to [0, 1]\}$, which we assume to be finite. Note that these hypotheses are a little different than what we are used to, as we are mapping into the unit interval, but this will be important for probability prediction.
- We have the set of loss functions $\mathcal{L} = \{l : \{0, 1\} \times [0, 1] \to [-1, 1]\}$, which we take to be ***ALL*** possible functions that fit the description (even ones that don't really "make sense" as loss functions, like non-convex functions).

2.2 The Problematic Definition of Omniprediction

We can define omniprediction as follows.

Definition 2.1. We want to find an h^* such that

$$\max_{l \in \mathcal{L}} \mathbb{E}_{\mathcal{D}}[l(y, h^*(x)] \leq \min_{h \in \mathcal{H}} \max_{l \in \mathcal{L}} \mathbb{E}_{\mathcal{D}}[l(y, h(x))] + \epsilon$$

Essentially, we have the set of all loss functions, and we want to find a hypothesis h^* that performs well on all of them. However, this predictor cannot exist! The reason behind this is that there are losses that negate each other, i.e. they are completely opposite. If h^* performs well on one loss function, it will perform exactly opposite on its negation, and therefore will never perform well on both.

Despite this, this definition is philosophically okay in the following sense: while a perfect predictor may not exist, there still exists some object that we can learn that is still useful for doing well across all losses.

2.3 The Proper Definition of Omniprediction

Assume that we have the Bayes predictor $p^*(x) = \Pr(y = 1|x)$. The best response for any fixed loss function $l \in \mathcal{L}$ is $BR_l(p) = \text{argmin}_{v \in [0,1]} \mathbb{E}_{\tilde{y} \in Ber(p)}[l(\tilde{y}, v)]$. This is the action you take, if you truly believe that the labels are generated by the predictor, then choose the action that minimizes the corresponding loss. Our "h^*" would be this best response $BR_l(p^*(x))$ for each and every $l \in \mathcal{L}$. Therefore, as long as we can learn the Bayes predictor somehow, then we can also minimize the losses.

Definition 2.2. p is an $(\mathcal{L}, \mathcal{H}, \epsilon)$ omnipredictor if for all $l \in \mathcal{L}$,

$$\mathbb{E}_D[l(y, BR_l(p(x)))] \leq \min_{h \in \mathcal{H}} \mathbb{E}_D[l(y, h(x))] + \epsilon$$

In other words, find an object p such that the best response to this p performs not too much worse than the best hypothesis trained, no matter what loss function you choose.

3 Step Calibration $\implies$ Omniprediction

3.1 Review

Recall that we defined step-calibration as a useful relaxation of calibration that simultaneously achieves two desirable properties: it admits a $\sqrt{T}$ rate, and it is sufficient to ensure that any downstream agent who best-responds to the predictions attains good utility (that is, step-calibration upper-bounds U-calibration).

Omni-prediction and the best-response functions used here are very similar to U-Calibration, except that they require a contextual formulation in which the predictor also conditions on x. We next define step-calibration in this contextual language.

Definition 3.1. A predictor $p : \mathcal{X} \to [0, 1]$ is $(\mathcal{H}, \epsilon)$ step calibrated if for all $h \in \mathcal{H}$, for all $v, w \in I_\epsilon = \{0, \epsilon, \ldots, 1 - \epsilon, 1\}$, the following inequality holds:

$$\left|\mathbb{E}_{\mathcal{D}}[(y - p(x)) \cdot \mathbb{1}\{p(x) \leq v, h(x) \leq w\}]\right| \leq \epsilon$$

The indicator in this definition is taken over two events:

1. $\{p(x) \leq v\}$, i.e. we predict the probability being at most v.
2. $\{h(x) \leq w\}$, i.e. the competitor predicts the probability being at most w.

In this event that both of these are true, we want very little bias. With this in mind, we make another definition.

3.2 Omniprediction Error

Definition 3.2. We define the omniprediction error, or OmniErr as

$$\text{OmniErr}(p, \mathcal{L}, \mathcal{H}) = \max_{l \in \mathcal{L}}(\mathop{\mathbb{E}}_{\mathcal{D}}[l(y, BR_l(p(x)))] - \min_{h \in \mathcal{H}} \mathop{\mathbb{E}}_{\mathcal{D}}[l(y, h(x))])$$

Intuitively, this omniprediction error is the worst case scenario difference between the best response to the predictor p versus the best performing hypothesis in the competitor class. We present a theorem with this new definition, relating it to step calibration error.

Theorem 3.3. *OmniErr*$(p, \mathcal{L}, \mathcal{H}) \leq 10$*StepCalErr*$(p, \mathcal{H}) + 10\epsilon$.

As a thought experiment, fix some p, and suppose that it is exactly the Bayes predictor $p^*(x)$. This means that

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}, \tilde{y} \sim Ber(p(x))}[l(y, BR_l(p(x)))] \leq \min_{h \in \mathcal{H}} \mathop{\mathbb{E}}_{x \sim \mathcal{D}, \tilde{y} \sim Ber(p)}[l(y, h(x))]$$

This inequality holds because best responding to the actual Bayes predictor is the best action you can possibly take. Next, say that these next inequalities holds true:

$$\left|\mathop{\mathbb{E}}_{(x,y) \sim \mathcal{D}}[l(y, BR_l(p(x)))] - \mathop{\mathbb{E}}_{x \sim \mathcal{D}, \tilde{y} \sim Ber(p)}[l(\tilde{y}, BR_l(p(x)))]\right| \leq \epsilon \tag{1}$$

$$\max_{h \in \mathcal{H}} \left| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[l(y, h(x))] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}, \tilde{y}\sim Ber(p)}[l(\tilde{y}, h(x))] \right| \leq \epsilon \tag{2}$$

By swapping out terms, we get a total regret of at most 2ϵ. Therefore, we seek to find a p that satisfies both of the above inequalities simultaneously. All that we need to ask of p, then, is that it is indistinguishable from the real Bayes predictor in the above sense. If we are able to accomplish this, then we will achieve omniprediction.

Then, the problem boils down to: *Can I learn a p that is not quite p^* but close enough?*

3.3 Outcome Indistinguishability Error

As an exercise, take any loss that is of the form $l(y, v)$ for $y \in \{0, 1\}$. We can write this loss function as

$$\begin{aligned} l(y, v) &= yl(1, v) + (1 - y)l(0, v) \\ &= y(l(1, v) - l(0, v)) + l(0, v) \\ &= \Delta l(v) + l(0, v) \end{aligned}$$

where $\Delta l(v)$ is known as the discrete derivative of l, which is just the difference between the loss when $y = 1$ and when $y = 0$. If $\Delta l(v)$ is small, then the loss views v as "good" no matter what the true label is. If $\Delta l(v)$ is large, then v is only good if the label is 1 or 0, but not the other.

We can plug this new massaged expression into the inequalities, and note that $l(0, v)$ is just a constant. Therefore, when we plug these into the expectations in both equalities, one will be positive and the other negative, so they will cancel completely. What remains is:

$$\left| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[y\Delta l(BR_l(p(x)))] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \left[\mathop{\mathbb{E}}_{\tilde{y}\sim Ber(p(x))}[\tilde{y}|x] \cdot \Delta l(BR_l(p(x))) \right] \right|$$

But note that $\mathbb{E}_{\tilde{y}\sim Ber(p)}[\tilde{y}|x] = p(x)$, as p is our Bayes predictor, so then the above becomes

$$\left| \mathbb{E}[(y - p(x)) \cdot \Delta l(BR_l(p(x)))] \right| \leq \epsilon \tag{3}$$

Which then implies that (2) becomes

$$\max_{h \in \mathcal{H}} \left| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[(y - p(x)) \cdot \Delta l(h(x))] \right| \leq \epsilon \tag{4}$$

The left hand side of (4) is typically called Hypothesis Outcome Indistinguishability Error, or HypOIErr$(p, \mathcal{L}, \mathcal{H})$, and the left hand side of (3) is typically called Decision Outcome Indistinguishability Error, or DecOIErr$(p, \mathcal{L})$. These inequalities look much more like step calibration constraints!

Proposition 3.4. *OmniErr$(p, \mathcal{L}, \mathcal{H}) \leq$ DecOIErr$(p, \mathcal{L})$ + HypOIErr$(p, \mathcal{L}, \mathcal{H})$*

So, if there is some way to solve for DecOIErr and HypOIErr, then we will be done! There is actually a way to accomplish this, and it turns out that it is not that hard. This will be discussed in the next class.

4 A few pointers to next lecture

4.1 Horrible Runtime

To close out, we will discuss the problem with just directly applying multi-objective learning to some sets of loss functions.

Imagine that we take $\mathcal{L}$ to be the set of all 1-Lipschitz functions. You can write a MOL problem that says something like: Find p such that

$$\max_{l \in \mathcal{L}} \max_{\text{"}f\text{"}} \left| \mathbb{E}[(y - p(x)) \cdot \Delta l(f(x))] \right| \leq \epsilon$$

To solve this problem, you need to somehow search over all of the possible 1-Lipschitz functions. This is usually done by forming an ϵ-net of the functions, but the covering number to do so, is $2^{1/\epsilon}$. This is absolutely horrible runtime, and any sample complexity will carry the term $\log(2^{1/\epsilon})$, so the dependence on ϵ has gotten polynomially too weak.

A cover is not the only way that we can try to control an infinitely large objecct. Instead, we can claim that for all "reasonable" losses, we can construct a basis for this space. If we apply MOL to this basis, then MOL becomes tractable. It just so happens that this basis is going to be formulated by the step functions given in step calibration.

4.2 What is "Reasonable"?

Definition 4.1. Suppose that $f : [0, 1] \to [-1, 1]$. The total variation of f is defined as

$$v(f) = \sup_{n \in \mathbb{N}} \sup_{0 = x_0 < x_1 < \ldots < x_n < x_{n+1} = 1} \sum_{i=1}^{n+1} \left| f(x_i) - f(x_{i-1}) \right|$$

Intuitively, this is how harshly the function varies up and down as we move left to right across the x-axis.

Definition 4.2. Define the set of all losses with bounded variation as

$$\mathcal{L}_{BV} = \{l : \max_{y} v(l(y, \cdot)) \leq 1\}$$

Similarly, define the set of loss functions that have a bounded discrete derivative

$$\Delta\mathcal{L}_{BV} = \{l : v(\Delta l) \leq 2\}$$

If a function does not fall into $\mathcal{L}_{BV}$, then the function has unbounded variation, i.e. it looks something like $\sin(1/x)$, which oscillates infinitely near the origin. There is no reason to believe that this function gives any useful prediction or information about human behavior. Therefore, even though $\mathcal{L}_{BV}$ contains very bad functions for loss, it at least weeds out some more "unreasonable" options with infinite variation.

5 Historical Notes

The problem of omniprediction was introduced by Gopalan et al. [2022]. Early omniprediction constructions such as [Gopalan et al., 2022] require sample complexity on the order of $O(1/\epsilon^{10})$, which is an exceedingly large dependence. More recently, Okoroafor et al. [2025] revisited this problem and obtained nearly optimal bounds: their algorithms achieve sample complexity $\tilde{O}(\epsilon^{-2})$ for randomized (non-deterministic) omnipredictors and $\tilde{O}(\epsilon^{-4})$ for deterministic ones, via a bespoke online-to-offline reduction. In these lectures, we instead follow an alternative approach proposed by Balakrishnan, Haghtalab, Hsu, Lee, and Zhao [Balakrishnan et al., 2025]. Their framework is based on step-calibration and multi-objective learning, and when specialized to omniprediction it not only matches the $\tilde{O}(\epsilon^{-2})$ rate for randomized predictors but also improves the deterministic sample complexity to $\tilde{O}(\epsilon^{-3})$. This approach also allows us to continue using the concept of step-calibration as a valuable pedagogical tool across several different lectures.

References

Sivaraman Balakrishnan, Nika Haghtalab, Daniel Hsu, Brian Lee, and Eric Zhao. Panprediction: Optimal predictions for any downstream task and loss. *arXiv preprint arXiv:2510.27638*, 2025.

Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215, page 79, 2022.

Princewill Okoroafor, Robert Kleinberg, and Michael P Kim. Near-optimal algorithms for omniprediction. *arXiv preprint arXiv:2501.17205*, 2025.