FALL 2026 BAIS 7900

Lecture 1: Introduction to Online Learning: Decisions, Feedback, and Regret

Lecturer: *Mengxiao Zhang* Date: Aug 24th

1 Overview of online learning

The usual statistical-learning story separates learning from deployment: collect a data set, estimate a model, and then use the fitted model to make decisions. That separation is often impossible in operations. A retailer must order today's inventory even while learning today's demand, a platform must choose which product to display while learning customer preferences, and a seller must post a price before learning whether the customer will buy. Waiting for a complete data set is itself a decision, and usually an expensive one.

Three features make these problems genuinely sequential. First, decisions are made before the relevant uncertainty is resolved, so a mistake produces a real cost rather than merely a poor fit on a training sample. Second, the decision often determines the data that become available. A seller observes demand only at the posted price; a retailer that stocks out does not observe the lost demand; and a recommendation system sees a response only to the item it displayed. Third, the environment may change. Demand can drift, competitors can react, and yesterday's best policy need not remain best tomorrow.

Online learning treats deployment itself as part of the learning process. The word "online" refers to the order in which information arrives, not to the internet. At round t, the learner has access only to the past, chooses a decision w_t, incurs a loss, and then receives feedback. The feedback is used to improve the next decision:

$$\text{past observations} \longrightarrow w_t \longrightarrow \text{outcome and loss} \longrightarrow \text{feedback}.$$

This viewpoint does not require the learner to predict *every outcome* correctly. In an adversarial or rapidly changing environment, accurate prediction may be impossible. A more robust question is whether the learner can make decisions whose cumulative loss is close to that of a meaningful benchmark chosen with hindsight. Regret formalizes this comparison. It separates what the learner could reasonably have controlled from uncertainty that no algorithm could have predicted.

We begin with several examples that will recur throughout the course.

Email junk filtering. When message t arrives, it is represented by features $x_t \in \mathbb{R}^d$, such as word frequencies, sender history, and link information. Before receiving a definitive label, the filter uses a weight vector w_t to decide whether the message should enter the inbox or the junk folder. If $y_t \in \{-1, +1\}$ is the eventual label, a standard logistic loss is

$$\ell_t(w) = \log\Big(1 + e^{-y_t \langle w, x_t \rangle}\Big).$$

The filter updates after the user marks a message as junk, rescues a legitimate message, or supplies another feedback signal. Thus the examples arrive sequentially, the classifier must act before the label is known, and feedback can be delayed or incomplete. This is online classification in a familiar operational form.

Online advertising. When a visitor with context x_t arrives, a platform chooses one of K advertisements. Let $r_{t,k} \in [0,1]$ denote the normalized revenue or click reward that ad k would generate for this visitor. Define $[K] \triangleq \{1, 2, \ldots, K\}$. If the platform randomizes its recommendation according to $w_t \in \Delta(K)$ where $\Delta(K)$ denotes the $K-1$ dimensional simplex defined as follows:

$$\Delta(K) := \left\{ w \in \mathbb{R}_K : \sum_{i=1}^{K} w_i = 1, w_i \geq 0 \ \forall i \in [K] \right\},$$

its conditional expected loss is

$$\ell_t(w) = -\sum_{k=1}^{K} w_k r_{t,k}.$$

Only the response to the displayed ad is observed; the rewards of the other ads remain counterfactual. Usually, the reward vector r_t is dependent on the context x_t in a certain way (but is unknown to the platform). The platform must therefore balance exploiting an ad that currently looks good with displaying other ads to learn whether they are better. With visitor features included, this becomes a contextual bandit problem.

Portfolio choice. Suppose there are N assets. At the beginning of day t, an investor chooses $w_t \in \Delta(N)$ where $w_{t,i}$ is the fraction of wealth placed in asset i. Let $r_{t,i} > 0$ be the gross return of asset i (e.g. $r_{t,i} = 1.1$ means you gain 10% by investing asset i at round t). If wealth before trading is W_t, then we have the wealth after round t (or at the beginning of $t+1$) to be

$$W_{t+1} = W_t \langle w_t, r_t \rangle .$$

The negative log return

$$\ell_t(w) = -\log \langle w, r_t \rangle$$

is a convenient loss because it turns compounded wealth into a sum:

$$\sum_{t=1}^{T} \ell_t(w_t) = -\log \frac{W_{T+1}}{W_1}.$$

Thus minimizing cumulative loss is exactly the same as maximizing final wealth.

Inventory control. For a single product, let D_t denote demand in period t. The retailer selects an order quantity $w_t \in [0, D_{\max}]$ before D_t is known. With holding cost h and shortage cost b, the period loss is

$$\ell_t(w) = h(w - D_t)_+ + b(D_t - w)_+,$$

where $(a)_+ = \max\{a, 0\}$. This loss is convex in the order quantity. The feedback, however, may be incomplete. Usually, the retailer sees only sales $S_t = \min\{w_t, D_t\}$ instead of the demand D_t. Therefore, a stockout reveals only that $D_t \geq w_t$. Larger orders can reveal more about demand, but they also expose the firm to more holding cost. The statistical and operational decisions are therefore intertwined.

Personalized pricing. At time t, a customer with context x_t arrives and the seller posts a price p_t. If the customer's value is V_t, the purchase indicator Y_t and revenue Rev_t are defined as

$$Y_t = \mathbf{1}\{V_t \geq p_t\}, \qquad \mathsf{Rev}_t = p_t Y_t.$$

For a fixed context, negative expected revenue can be written as

$$\ell_t(p) = -p\,\mathbb{P}(V_t \geq p \mid x_t).$$

Only the outcome at the posted price is observed. A purchase shows that $V_t \geq p_t$ and a non-purchase shows that $V_t < p_t$, but neither reveals what would have happened at another price.

Assortment choice. Suppose product i has revenue r_i and attraction parameter $v_i > 0$, while the no-purchase option has attraction $v_0 = 1$. If an assortment S_t is offered, the multinomial-logit model assigns probability

$$\mathbb{P}(i \mid S_t) = \frac{v_i}{v_0 + \sum_{j \in S_t} v_j}, \qquad i \in S_t.$$

The negative expected revenue of the assortment is

$$\ell(S_t) = -\frac{\sum_{i \in S_t} r_i v_i}{v_0 + \sum_{j \in S_t} v_j}.$$

The action is a set rather than a vector in an ordinary Euclidean space, and the observation is only the purchased item or no purchase. This example will require both combinatorial optimization and statistical learning.

Although these problems look different, each is specified by the same four ingredients: a feasible decision set, a sequence of losses, a feedback rule, and a benchmark. Keeping these ingredients separate is important. Two problems may have the same loss but different feedback and therefore require different algorithms.

2 A unified model

Let Ω be the set of feasible decisions. An online-learning problem runs for T rounds. At round t,

1. the learner chooses $w_t \in \Omega$ using only the information available before round t;
2. the environment determines a loss function $\ell_t : \Omega \to \mathbb{R}$;
3. the learner incurs $\ell_t(w_t)$ and observes feedback specified by the model.

This order is essential: the decision must be made *before* the new outcome is available.

2.1 Examples in the common notation

The table below returns to the examples from the opening and adds several standard learning problems. It records the two objects that define the optimization problem: the feasible set Ω and the round-t loss. The feedback is discussed separately because the same loss can be paired with very different observations.

Problem	**Decision set Ω**	**Loss $\ell_t(w)$**
Email filtering (classification)	$\{w \in \mathbb{R}^d : \|w\|_2 \le B\}$	$\log\big(1 + e^{-y_t\langle w, x_t\rangle}\big)$
Portfolio choice	$\Delta(N)$	$-\log\langle w, r_t\rangle$
Online ad selection	$\Delta(K)$	$-\sum_{k=1}^{K} w_k r_{t,k}$
Inventory control	$[0, D_{\max}]$	$h(w - D_t)_+ + b(D_t - w)_+$
Pricing	$[0,p_{\max}]$	$-w\mathbb{P}(V_t \ge w \mid x_t)$
Assortment	$\Big\{w \in \{0,1\}^N : \sum_{i=1}^{N} w_i \le K\Big\}$	$-\dfrac{\sum_{i=1}^{N} w_i r_i v_i}{v_0 + \sum_{j=1}^{N} w_j v_j}$

2.2 Feedback and environments

The feedback rule describes what is revealed after the decision. Under *full information*, the learner observes the entire loss function, or enough information to reconstruct it. First-order feedback reveals a gradient or subgradient at the chosen point. Under *bandit feedback*, the learner observes only the loss of the selected action. Inventory censoring and customer-choice data are examples of structured partial feedback that do not fit neatly into either extreme.

Example 2.1 (Feedback structures in applications)**.** The distinction between full and partial feedback is easiest to see by asking a simple question: *after making today's decision, which counterfactual outcomes become observable?*

- **Full information.** In portfolio choice, a small investor may place wealth in only one portfolio, but at the end of the day the returns of all publicly traded assets are observed. Likewise, when forecasting tomorrow's temperature across a fixed collection of locations, the realized temperatures can be observed even at locations for which a particular forecast was not used. In such examples, the learner can evaluate the loss of many decisions that were not actually chosen.
- **Bandit feedback.** In online advertising or recommendation, the platform observes the click, purchase, or revenue generated by the displayed item, but it does not observe what the same customer would have done had a different item been shown. Only the realized action is evaluated.
- **Structured partial feedback.** Dynamic pricing lies between these two extremes. If the seller posts price p_t and observes whether the customer buys, then a purchase reveals that $V_t \ge p_t$, while a nonpurchase reveals that $V_t < p_t$. Thus the seller learns more than the revenue of a single arm, but still does not observe the customer's exact value or the revenue that every other price would have produced. Inventory censoring has a similar structure: observing sales $\min\{w_t, D_t\}$ reveals exact demand only when there is no stockout.

The amount and structure of feedback determine what can be learned, even when the underlying loss function is unchanged.

The environment describes how losses are generated. In a stochastic model, the losses are independent draws from a fixed distribution. An oblivious adversary may choose an arbitrary sequence, but must fix it before the interaction begins. An adaptive adversary may choose the next loss using the past behavior of the learner.

Example 2.2 (Different models of the environment)**.** These models should be viewed as different assumptions on how much regularity the learner is willing to impose on the sequence of losses.

- **Stochastic environment.** A basic inventory model may assume that daily demands D_t are i.i.d. draws from a fixed distribution. Similarly, repeated customers in an advertising or pricing problem may be modeled as i.i.d. draws from a fixed population. The fixed distribution creates statistical regularity that the learner can exploit.
- **Oblivious adversary.** Sometimes the sequence may be highly irregular, yet is not affected by the learner's actions. For example, for a sufficiently small investor, a sequence of market returns can be treated as externally generated: the returns may be arbitrary, but the investor's portfolio choice does not change tomorrow's market return. Modeling such a sequence as fixed in advance gives the oblivious-adversary model.
- **Adaptive adversary.** Adaptivity matters when other agents can react to the learner. A competitor may change prices after observing a firm's past prices, an auction participant may adjust bids in response to previous outcomes, and an opponent in a repeated game may deliberately react to the learner's past play. Here the next loss can depend on the learner's history.
- **Non-stationary environment.** Many applications fall between i.i.d. sampling and a fully adversarial sequence. Demand may change across seasons, customer preferences may drift, or a system may switch between a few regimes. One can model such problems by allowing the data-generating distribution to evolve over time while controlling its total variation, number of changes, or another measure of non-stationarity.

Stronger regularity assumptions usually permit sharper guarantees, while weaker assumptions provide more robustness. Part of online learning is understanding which assumptions are justified by the application.

For randomized decisions, one further timing condition is needed. The environment may know the learner's distribution, but it cannot wait to see the fresh random action and then punish exactly that action. This non-anticipation condition prevents the benchmark from becoming meaningless.

Question 2.3. Why can the adversary depend on the learner's past behavior, but not observe the learner's fresh random action before choosing the current loss?

2.3 Regret

The standard performance measure is regret against the best fixed decision in hindsight:

$$R_T := \sum_{t=1}^{T} \ell_t(w_t) - \min_{w \in \Omega} \sum_{t=1}^{T} \ell_t(w). \tag{1}$$

The first term is the learner's actual cumulative loss. The second is the loss of the single best decision after all T losses are known. The comparator is allowed to see the future, but it must use the same decision on every round.

For example, suppose two pricing rules have loss vectors $\ell_1 = (0, 1)$, $\ell_2 = (1, 0)$, and $\ell_3 = (0, 1)$. A learner that selects rules $(2, 1, 2)$ suffers loss 3. The two fixed rules suffer 1 and 2, so the learner's regret is

$$R_3 = 3 - \min\{1, 2\} = 2.$$

It would be unfair to compare with a benchmark that selects rule 1 on rounds 1 and 3 but rule 2 on round 2: that benchmark changes after seeing every outcome, whereas the regret comparator is fixed.

A learner is called *no-regret* if $R_T \le B(T) = o(T)$. Dividing (1) by T gives

$$\frac{1}{T}\sum_{t=1}^{T}\ell_t(w_t) - \min_{w\in\Omega}\frac{1}{T}\sum_{t=1}^{T}\ell_t(w) \le \frac{B(T)}{T} = o(1).$$

Thus sublinear regret says that the learner's average loss is asymptotically no worse than that of the best fixed decision.

There are two natural objections to this benchmark. First, the learner may change decisions while the comparator cannot. This asymmetry is deliberate: the comparator has perfect hindsight, and competing with it is already nontrivial. Regret can even be negative when a changing learner outperforms every fixed action. Stronger notions, including switching and dynamic regret, allow the comparator to change over time.

Second, an adaptive environment might have reacted differently had the learner followed another policy. Static regret evaluates every decision on the realized loss sequence and therefore ignores this counterfactual change. Policy regret was introduced for such settings. Static regret remains the starting point because it is mathematically tractable, supports stronger regret notions, and has powerful consequences even when the environment changes over time. We will see later how we can handle stronger benchmark with a *changing decision sequence*.

3 Connection to statistical learning

Online learning makes no distributional assumption in its basic formulation. Classical statistical learning begins instead with an i.i.d. sample $z_1, \dots, z_T \sim D$ and asks for a single predictor that performs well on a new draw from the same distribution. For a loss $\ell : \Omega \times \mathcal{Z} \to [0,1]$, define the population risk

$$L(w) := \mathbb{E}_{z\sim D}[\ell(w,z)]$$

and let $w^\star \in \arg\min_{w\in\Omega} L(w)$. The goal is to output $\bar{w}$ with small excess risk $L(\bar{w}) - L(w^\star)$.

At first sight, this seems easier than online learning because the data are stochastic rather than adversarial. Indeed, that is the case! The useful fact is that a no-regret online algorithm can be turned into a statistical learner by making one pass through the sample.

Online-to-batch procedure. Run an online algorithm on the examples in their given order. Before observing z_t, let the algorithm choose w_t; after observing z_t, feed it the loss

$$\ell_t(w) := \ell(w, z_t).$$

After T rounds, either select $\bar{w} = w_\tau$ for a uniformly random $\tau \in \{1, \dots, T\}$, or, when L is convex, return

$$\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t.$$

The first option does not require an operation for averaging predictors. The second uses convexity to obtain a deterministic average. The following theorem formally shows this online-to-batch conversion.

Theorem 3.1 (Online-to-batch conversion)**.** *Suppose the online algorithm satisfies* $\mathbb{E}[R_T] \le B(T)$*. Either output above satisfies*

$$\mathbb{E}[L(\bar{w})] - L(w^\star) \le \frac{B(T)}{T},$$

with convexity of L *required for the uniformly averaged output.*

Proof. Consider first the uniform-random-iterate output. The important observation is that w_t is chosen before the fresh example z_t is revealed. Conditional on the past, w_t is fixed and z_t still has distribution D. Hence, we know that

$$\mathbb{E}[L(w_t)] = \mathbb{E}[\ell(w_t, z_t)].$$

Averaging over the random index and then applying the regret guarantee gives

$$\begin{aligned}
\mathbb{E}[L(\bar{w})] = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} L(w_t)\right] &= \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_{z_t}[\ell(w_t, z_t)]\right] \\
&\le \mathbb{E}\left[\min_{w\in\Omega} \frac{1}{T}\sum_{t=1}^{T} \ell(w, z_t)\right] + \frac{B(T)}{T} \\
&\le \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\ell(w^\star, z_t)] + \frac{B(T)}{T} && \text{(since } w^\star \in \Omega\text{)} \\
&= L(w^\star) + \frac{B(T)}{T}.
\end{aligned}$$

For the averaged output, Jensen's inequality gives

$$L\left(\frac{1}{T}\sum_{t=1}^{T} w_t\right) \le \frac{1}{T}\sum_{t=1}^{T} L(w_t),$$

after which the same calculation applies. □

The one-pass order is not a technical convenience but is what makes $\mathbb{E}[L(w_t)] = \mathbb{E}[\ell(w_t, z_t)]$ valid. If the same training example is reused, a later iterate can depend on that example, and this equality may fail. This explains why regret analysis can prove generalization for one-pass algorithms even when a traditional empirical-risk argument does not.

3.1 A high-probability version

In fact, we can achieve stronger results on the excess risk if we have R_T bounded (instead of its expectation). We first record the two concentration inequalities needed below. They look similar, but their assumptions are different. Hoeffding's inequality applies to independent variables. Azuma–Hoeffding applies to a sequential process whose next increment has conditional mean zero given the past.

Theorem 3.2 (Hoeffding's inequality)**.** *Let* $U_1, \dots, U_T$ *be independent random variables with* $U_t \in [a_t, b_t]$ *almost surely. Then, for every* $s > 0$*,*

$$\mathbb{P}\left(\sum_{t=1}^{T}(U_t - \mathbb{E}\left[U_t\right]) \ge s\right) \le \exp\left(-\frac{2s^2}{\sum_{t=1}^{T}(b_t - a_t)^2}\right).$$

In particular, if $U_t \in [0,1]$, then with probability at least $1-\delta$,

$$\frac{1}{T}\sum_{t=1}^{T} U_t \le \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[U_t\right] + \sqrt{\frac{\log(1/\delta)}{2T}}.$$

For the sequential version, let $\mathcal{F}_t$ denote the information available through round t. A sequence X_t is a *martingale-difference sequence* if X_t is measurable with respect to $\mathcal{F}_t$ and $\mathbb{E}[X_t \mid \mathcal{F}_{t-1}] = 0$.

Theorem 3.3 (Azuma–Hoeffding inequality)**.** *Let $X_1, \dots, X_T$ be a martingale-difference sequence satisfying $|X_t| \le c_t$ almost surely. Then, for every $s > 0$,*

$$\mathbb{P}\left(\sum_{t=1}^{T} X_t \ge s\right) \le \exp\left(-\frac{s^2}{2\sum_{t=1}^{T} c_t^2}\right).$$

If $|X_t| \le 1$, then with probability at least $1-\delta$,

$$\frac{1}{T}\sum_{t=1}^{T} X_t \le \sqrt{\frac{2\log(1/\delta)}{T}}.$$

The expected online-to-batch guarantee can now be strengthened by controlling two sampling errors. The first involves the adaptive decisions w_t. Define

$$X_t = L(w_t) - \ell(w_t, z_t).$$

Because w_t is selected before z_t, we have $\mathbb{E}_{z_t}[\ell(w_t, z_t) \mid \mathcal{F}_{t-1}] = L(w_t)$ and the sequence satisfies $\mathbb{E}[X_t \mid \mathcal{F}_{t-1}] = 0$ and $|X_t| \le 1$. Azuma–Hoeffding therefore controls $\sum_t X_t$. The second error compares the empirical loss of the fixed decision $w^\star$ with $L(w^\star)$ and is handled by the ordinary Hoeffding inequality.

Theorem 3.4 (High-probability online-to-batch conversion)**.** *Assume Ω and $\ell(\cdot, z)$ are convex and $\ell(w, z) \in [0,1]$. For $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$, with probability at least $1-\delta$,*

$$L(\bar{w}) \le L(w^\star) + \frac{R_T}{T} + 2\sqrt{\frac{2\log(2/\delta)}{T}}.$$

Proof. Jensen's inequality and Azuma–Hoeffding imply, except on an event of probability $\delta/2$,

$$L(\bar{w}) \le \frac{1}{T}\sum_{t=1}^{T} L(w_t) \le \frac{1}{T}\sum_{t=1}^{T} \ell(w_t, z_t) + \sqrt{\frac{2\log(2/\delta)}{T}}.$$

By the definition of regret,

$$\sum_{t=1}^{T} \ell(w_t, z_t) \le \sum_{t=1}^{T} \ell(w^\star, z_t) + R_T.$$

Hoeffding's inequality gives, with another failure probability $\delta/2$,

$$\frac{1}{T}\sum_{t=1}^{T} \ell(w^\star, z_t) \le L(w^\star) + \sqrt{\frac{\log(2/\delta)}{2T}}.$$

A union bound combines the two events. Replacing the two different square-root terms by the displayed common upper bound gives the claim. □

Consequently, a regret bound of $\mathcal{O}(\sqrt{T})$ produces the usual statistical rate $T^{-1/2}$. An expected regret guarantee, on the other hand, belongs in the expected theorem and cannot simply be transferred into the high-probability statement.