CS395T: Continuous Algorithms, Part XIII
Generalized linear models

Kevin Tian

1 Proper loss functions

In this lecture, we explore the use of optimization techniques for *supervised learning*, a basic problem setting in machine learning applications. In supervised learning, there is a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}$. The marginal of the first d coordinates, denoted $\mathcal{D}_{\mathbf{x}}$, represents *features* $\mathbf{x}$ associated with an example (e.g., height, length, temperature, and so on). Moreover, the conditional distribution of the last coordinate, denoted $\mathcal{D}_{y|\mathbf{x}}$, represents the *label* y of the example. In the common case of *binary classification*, where $\mathcal{Y} = \{0, 1\}$, the label is usually used to indicate some quality that we want to predict (e.g., whether an image is a dog or a cat). Implicit in this setup is the belief that y is a dependent variable predicted by the features $\mathbf{x}$.

We are primarily interested in learning models that can be use for predicting future unlabeled examples from their features. In a standard supervised learning setting, we first draw examples $\{(\mathbf{x}_i, y_i)\}_{i \in [n]} \sim_{\text{i.i.d.}} \mathcal{D}$, and we want to learn a predictor $f : \mathcal{X} \to \mathbb{R}$ that is predictive of $\mathbb{E}_{y \sim \mathcal{D}_{y|\mathbf{x}}}[y]$. This lecture primarily focuses on the binary classification setting, although in principle, many of the techniques we develop can be used for more general supervised learning setups. In particular, we discuss extensions to the multiclass prediction setting when they are relevant.

Realizable and agnostic learning. Fix some choice of distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{0, 1\}$, as well as a family of classifiers $\mathcal{C}$. We assume each $c \in \mathcal{C}$ is a function $c : \mathcal{X} \to \mathbb{R}$. For example, $\mathcal{C}$ could index a set of linear functions $c_{\mathbf{w}}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$. Intuitively, our goal is to use some $c \in \mathcal{C}$ to predict the mean label function $\mathbb{E}_{\mathcal{D}_{y|\mathbf{x}}}[y]$. When there exists $c^\star \in \mathcal{C}$ with

$$c^\star(\mathbf{x}) = \mathbb{E}_{\mathcal{D}_{y|\mathbf{x}}}[y] \text{ for all } \mathbf{x} \in \mathcal{X}, \tag{1}$$

we say that we are in the *realizable setting*; when no such $c^\star$ exists, we are in the *agnostic setting*, where we want to compete with the best classifier in $\mathcal{C}$ at predicting $\mathbb{E}_{\mathcal{D}_{y|\mathbf{x}}}[y]$.

We evaluate the quality of predictors via a loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$. Our goal is usually to minimize the classification loss, i.e., output a classifier $c \in \mathcal{C}$ such that

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell\left(c(\mathbf{x}), y\right)\right] \approx \min_{c^\star \in \mathcal{C}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell\left(c^\star(\mathbf{x}), y\right)\right]. \tag{2}$$

A few examples of popular loss functions $\ell(p, y)$ include the 0-1 loss, squared loss, and cross entropy loss, which we define below respectively:

$$\mathbb{I}_{p \neq y},\ (p - y)^2,\ (1 - y)\log\left(\frac{1}{1-p}\right) + y \log\left(\frac{1}{p}\right). \tag{3}$$

Proper losses. A basic desiderata of loss functions is that in the realizable setting, it should be optimal to output the ground truth $c^\star$. Losses satisfying this property are called *proper losses*. Formally, a proper loss should have the property that for all $p^\star \in [0, 1]$,

$$p^\star \in \text{argmin}_{p \in [0,1]} \mathbb{E}_{y \sim \text{Bern}(p^\star)}\left[\ell(p, y)\right]. \tag{4}$$

In (4), we use $\text{Bern}(p)$ to denote the Bernoulli distribution over $\mathcal{Y}$ with mean p. Somewhat remarkably, there is a complete characterization of proper losses for binary classification in terms

of convex functions. To describe it, we introduce some notation. For $p \in [0,1]$, let

$$\begin{aligned} H(p) &:= \mathbb{E}_{y\sim\text{Bern}(p)}\left[\ell(p,y)\right] \\ &= \min_{q\in[0,1]} \left\{\mathbb{E}_{y\sim\text{Bern}(p)}\left[\ell(q,y)\right]\right\} \\ &= \min_{q\in[0,1]} \left\{(1-p)\ell(q,0) + p\ell(q,1)\right\}. \end{aligned} \tag{5}$$

The second equality above holds because ℓ is proper, i.e., by (4). The consequence of applying the transformation (5), H, is called the *entropy* of ℓ. For example, if ℓ is the cross entropy loss,

$$\ell\left(p,y\right) = (1-y)\log\left(\frac{1}{1-p}\right) + y\log\left(\frac{1}{p}\right) \implies H(p) = (1-p)\log\left(\frac{1}{1-p}\right) + p\log\left(\frac{1}{p}\right).$$

Similarly if ℓ is the squared loss, then $H(p) = p(1-p)$. In both cases, $p = \frac{1}{2}$ maximizes entropy.

Next, $H : [0,1] \to \mathbb{R}$ is concave, as the minimum of linear functions of p (Lemma 1, Part III). We define $\omega := -H$ to be its (convex) negation. By Lemma 1, Part III, and the definition (5),

$$\omega'(p) = -H'(p) = \ell(p,0) - \ell(p,1). \tag{6}$$

We are now ready to give our proper loss characterization.

Lemma 1. *Let $\ell : [0,1]\times\{0,1\} \to \mathbb{R}$ be a proper loss (i.e., ℓ satisfies (4)), and define $H : [0,1] \to \mathbb{R}$ as in (5). Then for all $(p,y) \in [0,1] \times \{0,1\}$,*

$$\ell\left(p,y\right) = -\omega\left(p\right) - \omega'(p)(y-p). \tag{7}$$

Conversely, for any convex $\omega : [0,1] \to \mathbb{R}$, the loss $\ell : [0,1] \times \{0,1\}$ defined in (7) is proper.

Proof. First suppose that ℓ is a proper loss. We directly compute that

$$\begin{aligned} -\omega\left(p\right) - \omega'(p)(y-p) &= H(p) + \left(\ell\left(p,1\right) - \ell\left(p,0\right)\right)(y-p) \\ &= p\ell\left(p,1\right) + (1-p)\,\ell\left(p,0\right) + \left(\ell\left(p,1\right) - \ell\left(p,0\right)\right)(y-p) \\ &= (1-y)\,\ell\left(p,0\right) + y\ell\left(p,1\right) = \ell(p,y). \end{aligned}$$

In the first line, we used (6). This proves the first statement; we move onto the second. Suppose that $\omega : [0,1] \to \mathbb{R}$ is convex. Then we should have that if $y \sim \text{Bern}(p)$, then $p \in [0,1]$ minimizes $\mathbb{E}[\ell(p,y)]$. Indeed, defining ℓ as in (7),

$$\begin{aligned} \mathbb{E}_{y\sim\text{Bern}(p)}\left[\ell(q,y) - \ell\left(p,y\right)\right] &= -\omega(q) + \omega'(q)(q-p) + \omega(p) + \omega'(p)(p-p) \\ &= \omega(p) - \omega(q) - \omega'(q)(p-q) = D_\omega(p\|q) \geq 0, \end{aligned}$$

where we used that $\mathbb{E}_{\text{Bern}(p)}[y] = p$, and applied convexity in the last line. □

Our derivation gives a way to convert any convex $\omega : [0,1] \to \mathbb{R}$ into a proper loss function via (7). In fact, it is often nicer to describe this transformation starting from a conjugate convex function $\omega^* : \mathbb{R} \to \mathbb{R}$, such that its derivative $\sigma := (\omega^*)'$ maps $\mathbb{R} \to [0,1]$. By the theory of convex duality (i.e., Corollary 1, Part III), we have that if $\sigma(t) = p$ and σ is one-to-one, then

$$pt = \omega(p) + \omega^*(t).$$

Thus, we could alternatively parameterize (7), starting from $\sigma : \mathbb{R} \to [0,1]$, as

$$-\omega(p) + p\omega'(p) - y\omega'(p) = \omega^*(t) - yt = \int_0^t \left(\sigma\left(\tau\right) - y\right)\mathrm{d}\tau, \tag{8}$$

where $t := \omega'(p)$. The transformation (8) from σ into a proper loss is actually the more conventional viewpoint from the perspective of optimizing generalized linear models. In this context, $\sigma : \mathbb{R} \to [0,1]$ is called the *link function*, and the loss in (8) is called the associated *matching loss*.

Generalized linear models. We can now develop the theory of generalized linear models (GLMs) for binary classification. A GLM makes predictions from features $\mathbf{x} \in \mathbb{R}^d$ using a linear predictor $\mathbf{w} \in \mathbb{R}^d$, combined with a link function $\sigma : \mathbb{R} \to [0, 1]$. It posits that

$$\sigma(\mathbf{w} \cdot \mathbf{x}) = \mathbb{E}_{y \sim \mathcal{D}_{y|\mathbf{x}}}[y] \tag{9}$$

is the mean function governing the labels. We require that the link function is monotone nondecreasing: this arises from σ being derived in (8) as the gradient of a convex function ω^*.

The model (9) is quite flexible; it only asks that the data have a "preferred direction" $\mathbf{w}$, and that the more correlated $\mathbf{x}$ is with $\mathbf{w}$, the more likely the label equals 1. When σ is known and fixed, (9) is the GLM with link σ; otherwise, if σ is treated as a flexible part of the model, (9) is called a *single-index model* (SIM). SIMs are an example of a semiparametric model family: the linear function $\mathbf{w}$ is the parametric component, and the arbitrary monotone function $\sigma : \mathbb{R} \to [0, 1]$ is the nonparametric component which must be jointly estimated from data, alongside $\mathbf{w}$.

In the remainder of the lecture, we focus on learning a realizable SIM in the binary classification setting, i.e., where (9) holds for some unknown $(\sigma, \mathbf{w})$. We discuss various extensions, especially to the agnostic setting, in Section 3. Briefly, we mention that there is work on generalizing this lecture's techniques to more complex settings involving multiple classes or linear predictors. For example, the theory developed this section generalizes to proper losses on multiclass predictions, where the label $\mathbf{y}$ of each example belongs to the set $\{\mathbf{e}_i\}_{i \in [k]}$ (see, e.g., Chapter 14 of [Duc23]). Moreover, a topic of significant interest in the theory of neural networks (among other applications) is *multi-index models*, where predictions given features $\mathbf{x} \in \mathbb{R}^d$ depend on a designated subspace represented by $\mathbf{W} \in \mathbb{R}^{d \times k}$; for a survey of recent work in the area, see [DKK+24].

We now set up some specific notation for the rest of this lecture. Let $\sigma : \mathbb{R} \to [0, 1]$ be a monotone nondecreasing (henceforth in this lecture, monotone) link function. Following (8), we denote the *matching loss* associated with σ, introduced by [AHW95], via

$$\ell_{\mathsf{m},\sigma}(t, y) := \int_0^t (\sigma(\tau) - y)\, \mathrm{d}\tau. \tag{10}$$

For example, if $\sigma(t) = t$, then $\ell_{\mathsf{m},\sigma}$ is the squared loss (up to a constant); if $\sigma(t) = \frac{\exp(t)}{1+\exp(t)}$ is the logit, then $\ell_{\mathsf{m},\sigma}(t, y) = \log(1 + \exp(t)) - yt$. We summarize some important properties of $\ell_{\mathsf{m},\sigma}$ here.

Lemma 2. *Let $\sigma : \mathbb{R} \to [0, 1]$ be a monotone link function, and define $\ell_{\mathsf{m},\sigma}$ as in (10). Then*

$$\frac{\partial}{\partial t}\ell_{\mathsf{m},\sigma}(t, y) = \sigma(t) - y, \ \frac{\partial^2}{\partial t^2}\ell_{\mathsf{m},\sigma}(t, y) = \sigma'(t). \tag{11}$$

Consequently, $\ell_{\mathsf{m},\sigma}(t, y)$ is convex in t, and if σ is β-Lipschitz, then $\ell_{\mathsf{m},\sigma}(t, y)$ is β-smooth in t. Finally, for any $p \in [0, 1]$ and $t \in \mathbb{R}$ with $\sigma(t) = p$, we have

$$t \in \operatorname{argmin}_{t \in \mathbb{R}} \mathbb{E}_{y \sim \mathrm{Bern}(p)}[\ell_{\mathsf{m},\sigma}(t, y)]. \tag{12}$$

Proof. The first conclusion (10) follows from a direct calculation. Because $\sigma'(t) \geq 0$ for all t, convexity is immediate (if σ is not differentiable, convexity follows because the first derivative $\sigma(t) - y$ is monotone in t). Moreover if σ is β-Lipschitz, then for any $t, t' \in \mathbb{R}$, we have

$$\left|\frac{\partial}{\partial t}\ell_{\mathsf{m},\sigma}(t, y) - \frac{\partial}{\partial t}\ell_{\mathsf{m},\sigma}(t', y)\right| = |\sigma(t) - \sigma(t')| \leq \beta|t - t'|,$$

i.e., $\ell_{\mathsf{m},\sigma}(t, y)$ is smooth in t (Definition 3, Part II). Finally, if $\sigma(t) = p$, then (12) follows from

$$\frac{\partial}{\partial t}\mathbb{E}_{y \sim \mathrm{Bern}(p)}[\ell_{\mathsf{m},\sigma}(t, y)] = \mathbb{E}_{y \sim \mathrm{Bern}(p)}[\sigma(t) - y] = p - p = 0.$$

□

Lemma 2 gives us the regularity properties needed to use the tools developed in Part II and III to efficiently optimize $\ell_{\mathsf{m},\sigma}$. We note that in the realizable SIM setting (9), the predictor $t(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ optimizes $\ell_{\mathsf{m},\sigma}$, because (9) implies the precondition for (12) to hold. The key algorithmic challenge

in learning SIMs is that we do not know σ, which must be jointly estimated. We explore this observation in Section 3 when we design an algorithm for learning realizable SIMs.

Finally, to draw a connection to our earlier discussion on proper losses, observe that for any monotone and one-to-one link function $\sigma : \mathbb{R} \to [0,1]$, we can define an associated loss

$$\ell_{\mathsf{p},\sigma}(p,y) := \ell_{\mathsf{m},\sigma}\left(\sigma^{-1}(p), y\right). \tag{13}$$

We call $\ell_{\mathsf{p},\sigma}$ the *proper loss* associated with σ. Note that $\ell_{\mathsf{p},\sigma}$ takes as inputs values in $[0,1] \times [0,1]$. We can view σ as a function that takes $t \in \mathbb{R}$ (the "unlinked space") into $[0,1]$ (the "linked space" of predictions). Under this one-to-one mapping, Lemma 2 implies that

$$p \in \operatorname{argmin}_{q \in [0,1]} \mathbb{E}_{y \sim \mathrm{Bern}(p)} \left[\ell_{\mathsf{p},\sigma}(q,y)\right],$$

justifying the name proper loss. Indeed, tracing back through our earlier derivation shows that $\ell_{\mathsf{p},\sigma}$ is exactly the function defined in (7), under the mapping $\omega' = \sigma^{-1}$.

Interestingly, properties of the matching loss (8) allow us to optimize nonconvex losses for measuring the prediction error of a SIM. For example, consider the *squared loss* of a predictor $\mathbf{x} \to \hat{\sigma}(\hat{\mathbf{w}} \cdot \mathbf{x})$, where $(\mathbf{x}, y) \sim \mathcal{D}$ follows the realizable SIM (9):

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\left(\hat{\sigma}\left(\hat{\mathbf{w}} \cdot \mathbf{x}\right) - y\right)^2\right]. \tag{14}$$

The squared loss (14) is one of the most popular loss functions in statistical learning settings, due to its various beneficial properties, e.g., the fact that it is proper. Unfortunately, even in the GLM setting where $\hat{\sigma} = \sigma$ is known and we are only trying to optimize over the linear predictor $\hat{\mathbf{w}}$, the objective (14) can be quite far from convex: indeed, in the agnostic setting, even simple link functions such as the logit induce square losses with exponentially many local minima [AHW95]. Nonetheless, by using the matching loss (8) as a convex surrogate, in Section 3 we will give learning guarantees even under the squared loss objective. This broader idea of using convex surrogates to drive progress has been very influential in the design of machine learning algorithms.

2 Isotonic regression

Before giving our full algorithm for learning SIMs, in this section, we first develop a simple nonparametric estimation method: the *pool-adjacent-violators (PAV)* algorithm [ABE+55, GW84].

The PAV algorithm solves a one-dimensional curve fitting problem called *isotonic regression*. In this problem, we are given inputs $\{y_i\}_{i\in[n]} \subset \mathbb{R} \times \{0,1\}$. Our goal is to output $\{p_i\}_{i\in[n]}$ solving

$$\min_{\{p_i\}_{i\in[n]} \subset [0,1]} \sum_{i\in[n]} (p_i - y_i)^2 \text{ subject to } p_1 \le p_2 \le \ldots \le p_n. \tag{15}$$

Intuitively, this problem arises in our SIM-learning algorithm because given a candidate linear predictor $\hat{\mathbf{w}}$, we wish to find the link $\hat{\sigma} : \mathbb{R} \to [0,1]$ that best fits our observations (in the squared loss), subject to being monotone. However, (15) is also of independent interest in one-dimensional nonparametric estimation, if we aim to compete with the family of monotone classifiers.

PAV. We now describe the PAV algorithm for solving (15). The algorithm maintains a partition $\mathcal{P}$ of $[n]$ into consecutive *pools* $P_1, \ldots, P_s$, so that the indices in each P_j are all smaller than the indices in P_{j+1}. Initially, $s = n$ and each $P_j = \{j\}$. For any pool P, we define

$$\bar{y}_P := \frac{1}{|P|} \sum_{i \in P} y_i$$

to be the average label in the pool. The algorithm terminates as soon as $\bar{y}_{P_j} \le \bar{y}_{P_{j+1}}$ for all $i \in [s-1]$, i.e., the pools have average labels in monotone order. At termination, it returns the label $p_i = \bar{y}_P$ for all indices $i \in P$, looping over all pools $P \in \mathcal{P}$, which is feasible for (15).

Before termination, the PAV algorithm repeatedly finds a pair of adjacent pools P_i and P_{i+1} such that $\bar{y}_{P_j} > \bar{y}_{P_{j+1}}$. It then merges these pools into a single pool, updating $s \leftarrow s-1$. As each merge decreases the pool count by 1, PAV can straightforwardly be implemented in $O(n^2)$ time.

By being more clever, we can obtain a faster runtime of $O(n)$. Roughly speaking, the algorithm maintains a list of pool sizes and averages, so that merging adjacent pools takes $O(1)$ time. Further, the algorithm maintains a pointer to the j^{th} block, with the invariant that the first j pools have monotonic averages. We can amortize the cost of fixing this invariant to be $O(1)$ per pool merge, and as long as the invariant is maintained, detecting a new violation or advancing the pointer j also takes $O(1)$ time. Overall, this costs $O(1)$ time each time the pool count decreases, for an overall $O(n)$-time algorithm. For more details, we refer the reader to Section 4 of [GW84].

Properties of PAV. We next show that PAV optimally solves (15). To begin, we describe two important properties of its output. The first is that PAV always yields a *calibrated* predictor.

Definition 1 (Calibration)**.** *Let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y) \in \mathcal{X} \times \{0, 1\}$, and let $p : \mathcal{X} \to [0, 1]$ be a predictor. We say p is* calibrated *with respect to $\mathcal{D}$ if for all $v \in [0, 1]$,*

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[y \mid p(\mathbf{x}) = v\right] = v.$$

Asking p to be calibrated is asking that, conditioned on $p(\mathbf{x}) = v$, the observed outcome $y \mid \mathbf{x}$ should average out to v. In other words, a calibrated p gives predictions that can be understood as probabilities. Calibration is a well-studied, basic interpretability measure, and is closely-related to our discussion in Section 1; for instance, postprocessing any predictor to be calibrated decreases any proper loss, essentially by applying (4). Understanding the calibration of modern neural networks is an ongoing research direction, see e.g., [GPSW17]. Here, we only need the following simple fact.

Lemma 3. *Let $p : \mathcal{X} \to [0, 1]$ be the predictor resulting from PAV run on inputs $\{y_i\}_{i\in[n]}$ where $\mathcal{X} := [n]$, and let $\mathcal{D}$ be uniform over $(i, y_i) \in \mathcal{X} \times \{0, 1\}$. Then p is calibrated with respect to $\mathcal{D}$.*

Proof. For every value $v \in [0, 1]$, note that $p_i = v$ only if i belongs to a pool P with average label $\bar{y}_P = v$. However, this clearly means $\mathbb{E}_{i\in[n]}[y_i \mid p_i = v] = v$, so p is calibrated. □

The other property of PAV we need is more subtle, and roughly translates to the requirement that the residuals $\{p_i - y_i\}_{i\in[n]}$ are anticorrelated with any monotone sequence.

Lemma 4. *Let $\{u_i\}_{i\in[n]}$ be monotone nondecreasing, and let $\{p_i\}_{i\in[n]}$ be the result of PAV run on inputs $\{y_i\}_{i\in[n]}$. Then $\sum_{i\in[n]}(p_i - y_i)u_i \geq 0$.*

Proof. We begin by proving a much more general statement. Let $\mathcal{P}_t$ be the set of pools (i.e., a partition of $[n]$ by contiguous subsets) maintained by PAV after t pool merges, so $\mathcal{P}_0$ places each index in a singleton pool. We claim that for all iterations t before termination, any pool $P = [\ell, r] \in \mathcal{P}_t$, and any index $j \in P$, the following inequality holds:

$$\sum_{i=\ell}^{j} (y_i - \bar{y}_P) \geq 0. \tag{16}$$

In other words, pool prefix averages always exceed overall pool averages, at any point in the algorithm and for any pool prefix. The base case $t = 0$ is clear, as $y_j = \bar{y}_P$ for all $P = \{j\}$. Next, consider some iteration $t \geq 1$, and suppose (16) holds for $P \in \mathcal{P}_{t-1}$. Suppose that in this iteration, adjacent pools $P = [\ell, r], P' = [\ell' = r + 1, r'] \in \mathcal{P}_{t-1}$ were merged to form pool $Q = P \cup P'$.

To complete the induction, we need to show that (16) holds for $P \leftarrow Q$, as this is the only pool that has changed. Observe that $\bar{y}_P > \bar{y}_{P'}$ implies

$$\bar{y}_Q = \frac{r - \ell + 1}{r' - \ell + 1}\bar{y}_P + \frac{r' - \ell' + 1}{r' - \ell + 1}\bar{y}_{P'} \in (\bar{y}_{P'}, \bar{y}_P)\,.$$

Thus, for all $j \in P$, we have by induction that

$$\sum_{i=\ell}^{j} (y_i - \bar{y}_Q) \geq \sum_{i=\ell}^{j} (y_j - \bar{y}_P) \geq 0. \tag{17}$$

It remains to consider $j \in P'$. If $j = r'$, (16) with $P \leftarrow Q$ is clear; otherwise, for $j \leq r' - 1$,

$$\begin{aligned}\sum_{i=\ell}^{j}(y_i - \bar{y}_Q) &= -\frac{j-\ell+1}{r'-j}\sum_{i=j+1}^{r'}(y_i - \bar{y}_Q) \\ &\geq -\frac{j-\ell+1}{r'-j}\sum_{i=j+1}^{r'}(y_i - \bar{y}_{P'}) \\ &= \frac{j-\ell+1}{r'-j}\cdot\frac{j-\ell'+1}{r'-j}\cdot\sum_{i=\ell'}^{j}(y_i - \bar{y}_{P'}) \geq 0.\end{aligned} \tag{18}$$

The first and third lines above used the definitions of $\bar{y}_Q$ and $\bar{y}_{P'}$ as pool averages; the second line used our earlier conclusion $\bar{y}_{P'} < \bar{y}_Q$. Thus by combining (17) and (18), we have shown (16).

Now the claim follows from (16) and monotonicity of $\{u_i\}_{i\in[n]}$, upon applying the Abel transformation. For any pool $P = [\ell, r]$ at termination, let $s_{\ell-1} := 0$, and let $s_j := \sum_{i=\ell}^{j}(y_i - \bar{y}_P) \geq 0$ for all $j \in P$ (where the inequality holds because of (16)). Then,

$$\begin{aligned}\sum_{j=\ell}^{r}(p_j - y_j)u_j &= \sum_{j=\ell}^{r}(\bar{y}_P - y_j)u_j \\ &= \sum_{j=\ell}^{r} s_j(u_{j+1} - u_j) + u_\ell s_{\ell-1} - u_{r+1}s_r \\ &= \sum_{j=\ell}^{r} s_j(u_{j+1} - u_j) \geq 0,\end{aligned}$$

where we used that $s_r = 0$ in the third line. We can now sum over all pools. □

PAV solves isotonic regression. We can combine Lemmas 3 and 4 to prove a strong statement about PAV's optimality for isotonic regression, first observed in [BP13]. In particular, we can show that PAV is optimal for (15) when the squared loss is replaced by *any proper loss*. The case of the squared loss is recovered by setting the link σ in Theorem 1 to be the identity function.

Theorem 1. *Let $\sigma : \mathbb{R} \to [0,1]$ be an arbitrary monotone, one-to-one link function, and define the associated proper loss $\ell_{\mathsf{p},\sigma} : [0,1] \times \{0,1\} \to \mathbb{R}$ as in (13). Let $\{p_i\}_{i\in[n]}$ be the result of PAV run on inputs $\{y_i\}_{i\in[n]} \in \{0,1\}^n$. Then for any $\{v_i\}_{i\in[n]} \subset [0,1]$ satisfying $v_1 \leq v_2 \leq \ldots \leq v_n$,*

$$\sum_{i\in[n]} \ell_{\mathsf{p},\sigma}(p_i, y_i) \leq \sum_{i\in[n]} \ell_{\mathsf{p},\sigma}(v_i, y_i).$$

Proof. Following the notation (8), we prove that for any $\{t_i\}_{i\in[n]} \subset \mathbb{R}$ with $t_1 \leq t_2 \leq \ldots \leq t_n$,

$$\sum_{i\in[n]} \ell_{\mathsf{m},\sigma}(\sigma^{-1}(p_i), y_i) \leq \sum_{i\in[n]} \ell_{\mathsf{m},\sigma}(t_i, y_i). \tag{19}$$

The desired claim follows by letting $t_i = \sigma^{-1}(v_i)$ for all $i \in [n]$.

Let $\mathcal{D}$ be the uniform distribution over (i, y_i) for $i \in [n]$. Let $\widehat{\mathcal{D}}$ be the distribution over $(i, y) \in [n] \times \{0,1\}$ which first draws $i \in [n]$ uniformly at random, and then samples $y \sim \mathrm{Bern}(p_i)$. By the definition of proper losses (i.e., by applying (12) to each pool),

$$\mathbb{E}_{(i,y)\sim\widehat{\mathcal{D}}}\left[\ell_{\mathsf{m},\sigma}\left(\sigma^{-1}(p_i), y\right)\right] \leq \mathbb{E}_{(i,y)\sim\widehat{\mathcal{D}}}\left[\ell_{\mathsf{m},\sigma}\left(t_i, y\right)\right]. \tag{20}$$

Moreover,

$$\begin{aligned}\mathbb{E}_{(i,y)\sim\mathcal{D}}\left[\ell_{\mathsf{m},\sigma}\left(\sigma^{-1}(p_i), y\right)\right] - \mathbb{E}_{(i,y)\sim\widehat{\mathcal{D}}}\left[\ell_{\mathsf{m},\sigma}\left(\sigma^{-1}(p_i), y\right)\right] &= \mathbb{E}_{i\sim_{\text{unif.}}[n]}\left[(p_i - y_i)\sigma^{-1}(p_i)\right] = 0, \\ \mathbb{E}_{(i,y)\sim\widehat{\mathcal{D}}}\left[\ell_{\mathsf{m},\sigma}\left(t_i, y\right)\right] - \mathbb{E}_{(i,y)\sim\mathcal{D}}\left[\ell_{\mathsf{m},\sigma}\left(t_i, y\right)\right] &= \mathbb{E}_{i\sim_{\text{unif.}}[n]}\left[(y_i - p_i)t_i\right] \leq 0.\end{aligned} \tag{21}$$

where the first line used the definition of $\ell_{\mathsf{m},\sigma}$ (see (10)) and Lemma 3, and the second line used the definition of $\ell_{\mathsf{m},\sigma}$ and Lemma 4. Thus, by summing (20) with (21),

$$\mathbb{E}_{(i,y)\sim\mathcal{D}}\left[\ell_{\mathsf{m},\sigma}(\sigma^{-1}(p_i), y)\right] - \mathbb{E}_{(i,y)\sim\mathcal{D}}\left[\ell_{\mathsf{m},\sigma}\left(t_i, y\right)\right] \leq 0,$$

which is the claim (19) to be proven, upon multiplying both sides above by n. □

3 Isotron

In this section, we consider the problem of learning a realizable SIM with respect to the squared loss, (14). Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$ for $\mathcal{X} \subseteq \mathbb{R}^d$, and define $\mathcal{D}_{\mathbf{x}}$, $\mathcal{D}_{y|\mathbf{x}}$ as in Section 1. For clarity, in this section we use $\sigma_\star : \mathbb{R} \to [0,1]$ to denote a designated monotone link function, and $\mathbf{w}_\star \in \mathbb{R}^d$ to denote a designated linear predictor, such that for all $\mathbf{x} \in \mathcal{X}$,

$$\sigma_\star(\mathbf{w}_\star \cdot \mathbf{x}) = \mathbb{E}_{y \sim \mathcal{D}_{y|\mathbf{x}}}[y]. \tag{22}$$

In other words, there exists a SIM $(\sigma_\star, \mathbf{w}_\star)$ that exactly predicts the mean label given features, (22). To simplify our statements, we use the following notation in this section:

$$\begin{aligned}
\ell_{\mathsf{m},\sigma}(\mathbf{w};\mathcal{D}) &:= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell_{\mathsf{m},\sigma}(\mathbf{w}\cdot\mathbf{x}, y)], \\
\ell_{\mathsf{sq},\sigma}(\mathbf{w};\mathcal{D}) &:= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma(\mathbf{w}\cdot\mathbf{x}) - y)^2\right], \\
\overline{\ell_{\mathsf{sq},\sigma}}(\mathbf{w};\mathcal{D}) &:= \ell_{\mathsf{sq},\sigma}(\mathbf{w};\mathcal{D}) - \ell_{\mathsf{sq},\sigma_\star}(\mathbf{w}_\star;\mathcal{D}).
\end{aligned} \tag{23}$$

The definition of $\overline{\ell_{\mathsf{sq},\sigma}}$ in (23) is motivated by a *bias-variance decomposition*:

$$\begin{aligned}
\overline{\ell_{\mathsf{sq},\sigma}}(\mathbf{w};\mathcal{D}) &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma(\mathbf{w}\cdot\mathbf{x}) - y)^2\right] - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma_\star(\mathbf{w}_\star\cdot\mathbf{x}) - y)^2\right] \\
&= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma(\mathbf{w}\cdot\mathbf{x}) - \sigma_\star(\mathbf{w}_\star\cdot\mathbf{x}))^2\right].
\end{aligned} \tag{24}$$

In other words, $\overline{\ell_{\mathsf{sq},\sigma}}$ is the explainable error that is caused by incorrect predictions $\sigma(\mathbf{w}\cdot\mathbf{x}) \neq \sigma_\star(\mathbf{w}_\star\cdot\mathbf{x})$, and $\ell_{\mathsf{sq},\sigma} - \overline{\ell_{\mathsf{sq},\sigma}}$ is the squared loss inherent to the problem due to label noise.

As previously described, in general the squared loss is a nonconvex function of the linear predictor $\mathbf{w}$ (even holding the link σ constant). However, the *matching loss* $\ell_{\mathsf{m},\sigma}$ of any link function σ is convex (Lemma 2), so we can hope to minimize it efficiently via gradient descent. This has an implication on minimizing the squared loss, in the realizable setting.

Lemma 5. *For $\beta > 0$, let $\mathcal{S}_\beta$ be the set of monotone β-Lipschitz $\sigma : \mathbb{R} \to [0,1]$, i.e.,*

$$\mathcal{S}_\beta := \{\sigma : \mathbb{R} \to [0,1] \mid 0 \le \sigma(t) - \sigma(t') \le \beta(t - t') \text{ for all } t' \le t\}.$$

Then if $\mathcal{D}$, a distribution on $\mathcal{X} \times \{0,1\}$, satisfies (22) for $\sigma_\star \in \mathcal{S}_\beta$, we have for any $\mathbf{w} \in \mathbb{R}^d$ that

$$\ell_{\mathsf{m},\sigma_\star}(\mathbf{w};\mathcal{D}) - \ell_{\mathsf{m},\sigma_\star}(\mathbf{w}_\star;\mathcal{D}) \ge \frac{1}{2\beta}\overline{\ell_{\mathsf{sq},\sigma_\star}}(\mathbf{w};\mathcal{D}). \tag{25}$$

Moreover, for any invertible $\hat{\sigma} \in \mathcal{S}_\beta$, we have that

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma(\mathbf{w}\cdot\mathbf{x}) - y)\left(\sigma_\star^{-1}(\sigma(\mathbf{w}\cdot\mathbf{x})) - \mathbf{w}_\star\cdot\mathbf{x}\right)\right] \ge \frac{1}{\beta}\overline{\ell_{\mathsf{sq},\sigma}}(\mathbf{w};\mathcal{D}). \tag{26}$$

Proof. First, we expand:

$$\begin{aligned}
\ell_{\mathsf{m},\sigma_\star}(\mathbf{w};\mathcal{D}) - \ell_{\mathsf{m},\sigma_\star}(\mathbf{w}_\star;\mathcal{D}) &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\int_{\mathbf{w}_\star\cdot\mathbf{x}}^{\mathbf{w}\cdot\mathbf{x}} (\sigma_\star(\tau) - y)\,\mathrm{d}\tau\right] \\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\left[\int_{\mathbf{w}_\star\cdot\mathbf{x}}^{\mathbf{w}\cdot\mathbf{x}} (\sigma_\star(\tau) - \sigma_\star(\mathbf{w}_\star\cdot\mathbf{x}))\,\mathrm{d}\tau\right] \\
&\ge \frac{1}{2\beta}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\left[(\sigma_\star(\mathbf{w}\cdot\mathbf{x}) - \sigma_\star(\mathbf{w}_\star\cdot\mathbf{x}))^2\right],
\end{aligned}$$

which proves (25) upon applying (24). The only inequality above used co-coercivity of the gradient, i.e., letting $\omega_\star^*$ be the β-smooth antiderivative of $\sigma_\star$, we have $\omega_\star^*(t) - \omega_\star^*(t') - \sigma_\star(t')(t - t') \ge \frac{1}{2\beta}(\sigma_\star(t) - \sigma_\star(t'))^2$ for all t, t' by Corollary 2, Part III. Next, we similarly have

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma(\mathbf{w}\cdot\mathbf{x}) - y)\left(\sigma_\star^{-1}(\sigma(\mathbf{w}\cdot\mathbf{x})) - \mathbf{w}_\star\cdot\mathbf{x}\right)\right] \\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\left[(\sigma(\mathbf{w}\cdot\mathbf{x}) - \sigma_\star(\mathbf{w}_\star\cdot\mathbf{x}))\left(\sigma_\star^{-1}(\sigma(\mathbf{w}\cdot\mathbf{x})) - \mathbf{w}_\star\cdot\mathbf{x}\right)\right] \\
&\ge \frac{1}{\beta}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\left[(\sigma_\star(\mathbf{w}\cdot\mathbf{x}) - \sigma_\star(\mathbf{w}_\star\cdot\mathbf{x}))^2\right],
\end{aligned}$$

proving the second claim. The only inequality used $\sigma_\star \in \mathcal{S}_\beta$, which means that for all t, t',

$$(\sigma_\star(t) - \sigma_\star(t'))\,(t - t') \geq \frac{1}{\beta}\,(\sigma_\star(t) - \sigma_\star(t'))^2\,.$$

□

Under the realizability assumption (22), the claim (25) shows that the minimizer of $\ell_{\mathsf{m},\sigma_\star}$ also induces the global minimizer of the squared loss. Thus, in the GLM setting where the link function $\sigma_\star \in \mathcal{S}_\beta$ is treated as known and fixed, we can optimize the squared loss in $\mathbf{w}$ simply by optimizing $\ell_{\mathsf{m},\sigma_\star}(\cdot;\mathcal{D})$, a smooth convex function. Indeed, achieving $2\beta\epsilon$ loss in $\ell_{\mathsf{m},\sigma_\star}$ implies loss ϵ in $\ell_{\mathsf{sq},\sigma_\star}(\cdot;\mathcal{D})$, via (25). To obtain a near-minimizer for $\ell_{\mathsf{m},\sigma_\star}(\cdot;\mathcal{D})$, any smooth convex optimization algorithm will do, e.g., Theorem 3, Part II or Theorem 2, Part V.

We now discuss the SIM case, where $\sigma_\star$ is unknown. First, it is worthwhile to compute the form of $\nabla\ell_{\mathsf{m},\sigma}(\cdot;\mathcal{D})$, the gradient used in these optimization methods. By the chain rule,

$$\nabla\ell_{\mathsf{m},\sigma}\,(\mathbf{w};\mathcal{D}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma\,(\mathbf{w}\cdot\mathbf{x}) - y)\,\mathbf{x}\right] = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}\left[(\sigma\,(\mathbf{w}\cdot\mathbf{x}) - \sigma_\star\,(\mathbf{w}_\star\cdot\mathbf{x}))\,\mathbf{x}\right], \tag{27}$$

where the last line used (22). When σ is the identity link and $\mathcal{D}$ is uniform over the rows of a regression matrix $\mathbf{X} = \{\mathbf{x}_i\}_{i\in[n]} \in \mathbb{R}^{n\times d}$ modeling $\mathbf{X}\mathbf{w} \approx \mathbf{y}$, the gradient of the (least squares) matching loss is $\nabla(\frac{1}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2) \propto \mathbb{E}_{i\sim_{\text{unif.}}[n]}[(\mathbf{w}\cdot\mathbf{x}_i - \mathbf{y}_i)\mathbf{x}_i]$, in agreement with (27).

We now present an elegant SIM learning algorithm called the *Isotron* due to [KS09], later explored in more depth by [KKKS11]. Briefly, the main idea is to alternate between learning $\mathbf{w}$ via gradient descent on $\ell_{\mathsf{m},\sigma}$ (with respect to the current guess of σ), and learning σ via the PAV algorithm in Section 2 (with respect to the current unlinked predictions $\mathbf{w}\cdot\mathbf{x}$).

Theorem 2. *Let $\mathcal{D}$, a distribution over $\mathcal{X}\times\{0,1\}$ for $\mathcal{X}\subseteq\mathbb{R}^d$, satisfy (22) for $\sigma_\star\in\mathcal{S}_\beta$, and let*

$$\left\|\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}\left[\mathbf{x}\mathbf{x}^\top\right]\right\|_{\text{op}} \leq L^2,\ \|\mathbf{w}_0 - \mathbf{w}_\star\|_2^2 \leq R^2,$$

for some $\mathbf{w}_0\in\mathbb{R}^d$. Consider the following iteration: for all $0\leq t<T$, let

$$\begin{aligned}\sigma_t &\leftarrow \operatorname{argmin}_{\sigma\in\mathcal{S}_\beta}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma\,(\mathbf{w}_t\cdot\mathbf{x}) - y)^2\right]\right],\\ \mathbf{w}_{t+1} &\leftarrow \mathbf{w}_t - \eta\nabla\ell_{\mathsf{m},\sigma_t}\,(\mathbf{w}_t;\mathcal{D})\,.\end{aligned}$$

Then for $\eta = \frac{1}{\beta L^2}$ and $T \geq \frac{\beta^2L^2R^2}{\epsilon}$, we have

$$\ell_{\mathsf{sq},\sigma_t}\,(\mathbf{w}_t;\mathcal{D}) \leq \ell_{\mathsf{sq},\sigma_\star}\,(\mathbf{w}_\star;\mathcal{D}) + \epsilon \text{ for some } 0\leq t<T.$$

Proof. First observe that we are playing standard gradient descent on the $\{\mathbf{w}_t\}_{0\leq t<T}$ iterates, so that Theorem 2, Part II implies the regret bound

$$\frac{1}{T}\sum_{0\leq t<T}\langle\nabla\ell_{\mathsf{m},\sigma_t}(\mathbf{w}_t;\mathcal{D}),\mathbf{w}_t - \mathbf{w}_\star\rangle \leq \frac{R^2}{2\eta T} + \frac{\eta}{2T}\sum_{0\leq t<T}\|\nabla\ell_{\mathsf{m},\sigma_t}(\mathbf{w}_t;\mathcal{D})\|_2^2\,. \tag{28}$$

We can bound the right-hand side above, recalling the formula (27), via

$$\begin{aligned}\|\nabla\ell_{\mathsf{m},\sigma}(\mathbf{w};\mathcal{D})\|_2^2 &= \sup_{\substack{\mathbf{v}\in\mathbb{R}^d\\ \|\mathbf{v}\|_2=1}}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}\left[(\sigma\,(\mathbf{w}\cdot\mathbf{x}) - \sigma_\star\,(\mathbf{w}_\star\cdot\mathbf{x}))(\mathbf{x}\cdot\mathbf{x})\right]^2\\ &\leq \sup_{\substack{\mathbf{v}\in\mathbb{R}^d\\ \|\mathbf{v}\|_2=1}}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}\left[(\sigma\,(\mathbf{w}\cdot\mathbf{x}) - \sigma_\star\,(\mathbf{w}_\star\cdot\mathbf{x}))^2\right]\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}\left[(\mathbf{x}\cdot\mathbf{v})^2\right]\\ &\leq \overline{\ell_{\mathsf{sq},\sigma}}\,(\mathbf{w};\mathcal{D})\left\|\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}\left[\mathbf{x}\mathbf{x}^\top\right]\right\|_{\text{op}} \leq L^2\overline{\ell_{\mathsf{sq},\sigma}}\,(\mathbf{w};\mathcal{D})\,.\end{aligned} \tag{29}$$

To bound the left-hand side of (28), we further have for all $0\leq t<T$ that

$$\begin{aligned}\langle\nabla\ell_{\mathsf{m},\sigma_t}(\mathbf{w}_t;\mathcal{D}),\mathbf{w}_t - \mathbf{w}_\star\rangle &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma_t(\mathbf{w}_t\cdot\mathbf{x}) - y)\,(\mathbf{w}_t\cdot\mathbf{x} - \mathbf{w}_\star\cdot\mathbf{x})\right]\\ &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma_t(\mathbf{w}_t\cdot\mathbf{x}) - y)\left(\mathbf{w}_t\cdot\mathbf{x} - \sigma_\star^{-1}\left(\sigma_t\,(\mathbf{w}_t\cdot\mathbf{x})\right)\right)\right]\\ &\quad + \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma_t(\mathbf{w}_t\cdot\mathbf{x}) - y)\left(\sigma_\star^{-1}\left(\sigma_t\,(\mathbf{w}_t\cdot\mathbf{x})\right) - \mathbf{w}_\star\cdot\mathbf{x}\right)\right]\\ &\geq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma_t(\mathbf{w}_t\cdot\mathbf{x}) - y)\left(\sigma_\star^{-1}\left(\sigma_t\,(\mathbf{w}_t\cdot\mathbf{x})\right) - \mathbf{w}_\star\cdot\mathbf{x}\right)\right]\\ &\geq \frac{1}{\beta}\overline{\ell_{\mathsf{sq},\sigma_t}}\,(\mathbf{w}_t;\mathcal{D})\,.\end{aligned} \tag{30}$$

In the second-to-last-line above, we used that σ_t is calibrated[1] and anticorrelated with monotone sequences (Lemmas 3 and 4), so

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma_t(\mathbf{w}_t\cdot\mathbf{x})-y)\,\sigma_\star^{-1}(\sigma_t(\mathbf{w}_t\cdot\mathbf{x}))\right]=0,\ \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\sigma_t(\mathbf{w}_t\cdot\mathbf{x})-y)\,\mathbf{w}_t\cdot\mathbf{x}\right]\geq 0.$$

We also used (26) in the last line of (30). Plugging (29) and (30) into (28), we obtain

$$\frac{1}{2\beta T}\sum_{0\leq t<T}\overline{\ell_{\mathsf{sq},\sigma_t}}\left(\mathbf{w}_t;\mathcal{D}\right)\leq\frac{\beta L^2R^2}{2T},$$

which gives the claim upon rearranging and choosing the iterate with smallest $\ell_{\mathsf{sq},\sigma_t}(\mathbf{w}_t;\mathcal{D})$. □

Theorem 2 proves that the semiparametric problem of learning realizable SIMs in the squared loss is polynomial time tractable. We mention a few extensions of Theorem 2 to more general settings.

Finite-sample bounds. Perhaps the biggest shortcoming of Theorem 2 is that, as stated, it does not yield an implementable algorithm due to its requirement of evaluating population average quantities such as $\nabla\ell_{\mathsf{m},\sigma}(\mathbf{w};\mathcal{D})$. To give guarantees on learning SIMs from a finite dataset $\{(\mathbf{x}_i,y_i)\}\sim_{\text{i.i.d.}}\mathcal{D}$, we must appeal to generalization bounds. The topic of generalization for infinite function families (e.g., the family of predictors $\mathbf{x}\to\sigma(\mathbf{w}\cdot\mathbf{x})$) is outside the scope of this course. However, we got a taste for this concept when proving net-based bounds on the conditioning of random Gaussian matrices in Proposition 1, Part IX and Lemma 2, Part XII.

To prove a finite-sample analog of Theorem 2, the main challenge is to argue that the link function fitting step used to compute σ_t is well-approximated via samples. The standard approach for doing so (e.g., as in [KKKS11]) goes through the *empirical Rademacher complexity*, which roughly measures the worst case deviation from a function evaluated on samples to its expectation. This in turn can be bounded via *chaining* appropriately-sized nets over the function class. We refer the reader to the excellent resources [vH16, Wu20] for accessible introductions to this topic.

Agnostically learning SIMs. A challenging extension to Theorem 2 asks for a learning result (e.g., in squared loss) with respect to the best-fitting SIM, under an arbitrary distribution $\mathcal{D}$ over $\mathcal{X}\times\{0,1\}$ where (22) need not hold. That is, we want to return $(\sigma,\mathbf{w})\in\mathcal{S}_\beta\times\mathbb{R}^d$ such that

$$\ell_{\mathsf{sq},\sigma}\left(\mathbf{w};\mathcal{D}\right)\approx\min_{(\sigma^\star,\mathbf{w}^\star)\in\mathcal{S}_\beta\times\mathbb{R}^d}\ell_{\mathsf{sq},\sigma_\star}\left(\mathbf{w}_\star;\mathcal{D}\right).\tag{31}$$

If our notion of approximation is multiplicative error, standard cryptographic assumptions imply that this agnostic learning goal is unachievable in polynomial time, even when the link function $\sigma=\sigma_\star$ is known and fixed [DKMR22]. Interestingly, a recent line of work shows that under certain distributional assumptions, e.g., concentration and anticoncentration of $\mathcal{D}_\mathbf{x}$ and regularity of the link function, multiplicative approximations of the form (31) are possible, even in the SIM setting [DGK+20, WZDD23, ZWDD24]. These results roughly proceed by characterizing the landscape of squared loss under distributional assumptions as having certain proxies for strong convexity, outside a neighborhood of the optimal choice of $\mathbf{w}_\star$. This landscape structure then implies that any highly-suboptimal candidate $\mathbf{w}$ can be improved via gradient methods.

Omniprediction. The aforementioned results show that we can understand the performance of gradient methods up to a constant multiplicative factor, for agnostically learning SIMs. Can we give a tight result on agnostically learning SIMs, perhaps of a different nature than square loss minimization? Modern work on supervised learning defined the notion of *omniprediction* [GKR+22], where a single predictor is simultaneously near-optimal for a family of loss functions against a set of candidates. In fact, Theorem 1 is an example of such an omniprediction guarantee, which says that PAV optimizes all proper losses against all monotone predictors.

Recently, [GHK+23, HTY25] showed that omniprediction for SIMs is achievable, even agnostically. Roughly, the omniprediction guarantee they show is that a single learned predictor simultaneously minimizes all matching losses against linear predictors. Perhaps surprisingly, the omnipredictor in [HTY25] simply postprocesses the iterates of Isotron (Theorem 2) itself, highlighting a deeper connection between nonparametric curve-fitting problems and supervised loss minimization.

[1] A limiting argument shows that under mild regularity assumptions, σ_t is well-approximated by the PAV solution to a finite sample of $\mathbf{w}\cdot\mathbf{x}$ for $\mathbf{x}\sim\mathcal{D}_\mathbf{x}$. The PAV solution for finite samples is calibrated (Lemma 3) so σ_t is as well. A similar argument holds for Lemma 4 at a population level.

Source material

Portions of this lecture are based on reference material in [Duc23, Kan23], as well as the author's own experience working in the field. We would like to thank Lunjia Hu and Chutong Yang for many helpful conversations related to this lecture.

References

[ABE+55] Miriam Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and Edward Silverman. An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, 26(4):641–647, 1955.

[AHW95] Peter Auer, Mark Herbster, and Manfred K. Warmuth. Exponentially many local minima for single neurons. In *Advances in Neural Information Processing Systems 8, NIPS*, pages 316–322. MIT Press, 1995.

[BP13] Niko Brummer and Johan du Preez. The PAV algorithm optimizes binary proper scoring rules. *arXiv preprint arXiv:1304.2331*, 2013.

[DGK+20] Ilias Diakonikolas, Surbhi Goel, Sushrut Karmalkar, Adam R. Klivans, and Mahdi Soltanolkotabi. Approximation schemes for relu regression. In *Conference on Learning Theory, COLT 2020*, volume 125 of *Proceedings of Machine Learning Research*, pages 1452–1485. PMLR, 2020.

[DKK+24] Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Agnostically learning multi-index models with queries. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024*, pages 1931–1952. IEEE, 2024.

[DKMR22] Ilias Diakonikolas, Daniel Kane, Pasin Manurangsi, and Lisheng Ren. Hardness of learning a single neuron with adversarial label noise. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022*, volume 151 of *Proceedings of Machine Learning Research*, pages 8199–8213. PMLR, 2022.

[Duc23] John Duchi. *Lecture Notes on Statistics and Information Theory*. 2023.

[GHK+23] Parikshit Gopalan, Lunjia Hu, Michael P. Kim, Omer Reingold, and Udi Wieder. Loss minimization through the lens of outcome indistinguishability. In *14th Innovations in Theoretical Computer Science Conference, ITCS 2023*, volume 251 of *LIPIcs*, pages 60:1–60:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[GKR+22] 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 of *LIPIcs*, pages 79:1–79:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017.

[GW84] Stephen J Grotzinger and Christoph Witzgall. Projections onto order simplexes. *Applied mathematics and Optimization*, 12(1):247–270, 1984.

[HTY25] Lunjia Hu, Kevin Tian, and Chutong Yang. Omnipredicting single-index models with multi-index models. In *STOC '25, 57th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2025.

[Kan23] Varun Kanade. *Computational Learning Theory*. 2023.

[KKKS11] Sham M. Kakade, Adam Kalai, Varun Kanade, and Ohad Shamir. Efficient learning of generalized linear and single index models with isotonic regression. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011*, pages 927–935, 2011.

[KS09] Adam Tauman Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT 2009 - The 22nd Conference on Learning Theory*, 2009.

[vH16] Ramon van Handel. *Probability in High Dimension*. 2016.

[Wu20] Yihong Wu. *Lecture Notes on: Information-Theoretic Methods for High-Dimensional Statistics*. 2020.

[WZDD23] Puqian Wang, Nikos Zarifis, Ilias Diakonikolas, and Jelena Diakonikolas. Robustly learning a single neuron via sharpness. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 36541–36577. PMLR, 2023.

[ZWDD24] Nikos Zarifis, Puqian Wang, Ilias Diakonikolas, and Jelena Diakonikolas. Robustly learning single-index models via alignment sharpness. In *Forty-first International Conference on Machine Learning, ICML 2024*. OpenReview.net, 2024.