CPSC 532D — 9. MARGINS AND SVMS

Danica J. Sutherland
University of British Columbia, Vancouver
Fall 2023

1 MOTIVATION

Remember that a linear classifier is given by $h(x) = \operatorname{sgn}(w \cdot x + b \geq 0)$; a *homogeneous* linear classifier is $h(x) = \operatorname{sgn}(w \cdot x)$. You can reduce from a general linear classifier to a homogeneous one by changing the data: use $\tilde{x} = \begin{bmatrix} 1 & x \end{bmatrix} \in \mathbb{R}^{d+1}$ and $\tilde{w} = \begin{bmatrix} b & w \end{bmatrix}$. So, for now, we're only going to worry about homogeneous classifiers. (Sometimes adding an intercept back in ends up being nontrivial, though – pay attention to that step!)

Recall $\operatorname{sgn}(t)$ *is* 1 *if* $t \geq 0$, -1 *otherwise.*

Letting $\mathcal{H} = \{x \mapsto \operatorname{sgn}(w \cdot x) : w \in \mathbb{R}^d\}$, we know from our study of VC theory that $\operatorname{VCdim}(\mathcal{H}) = d$ and each of the following hold for 0-1 loss with probability at least $1 - \delta$ over the choice of $S \sim \mathcal{D}^m$:

$$\sup_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) \leq \sqrt{\frac{2d}{m}[\log m + 1 - \log d]} + \sqrt{\frac{1}{2m} \log \frac{1}{\delta}}$$

$$L_{\mathcal{D}}(\hat{h}_S) - \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \leq \sqrt{\frac{2d}{m}[\log m + 1 - \log d]} + \sqrt{\frac{2}{m} \log \frac{2}{\delta}}$$

where $\hat{h}_S$ is any ERM.

So, for any fixed d, this means that ERM will work once m is big enough. But sometimes we have a really big d, and this doesn't tell us anything until $m/\log m > 2d$. Sometimes we even have an *infinite* d, and then this doesn't tell us anything at all; this is often the case with kernel methods.

Often, though, when d is big we end up with a hypothesis h that has *small norm*. This might be because we explicitly *try* to find a small-norm solution, and/or because our optimization algorithm implicitly prefers small-norm solutions; more on both situations later in the course.

To analyze that, let's again use $\mathcal{H}_B = \{x \mapsto w \cdot x : \|w\| \leq B\}$ – note this is a class that outputs continuous real numbers, not "hard" classifications, but we can get a class of binary classifiers out with $\operatorname{sgn} \circ \mathcal{H}_B$.

But note that $\operatorname{VCdim}(\operatorname{sgn} \circ \mathcal{H}_B) = d$ for any B: since VC dimension is worst-case over all possible input distributions, we can take any set that the full $\mathcal{H}$ can shatter and just scale it up so that we can still shatter it with a small-norm predictor. So, we'll need a distribution-dependent notion of complexity to do better than this; something like Rademacher complexity.

Now, recall from the Rademacher notes that $\mathbb{E}_S \operatorname{Rad}\left(\mathcal{H}_B\big|_{S_x}\right) \leq \frac{B}{\sqrt{m}} \sqrt{\mathbb{E}\|x\|^2}$. To use this in a generalization bound for the 0-1 loss, though, we'd need to bound $\mathbb{E}_S \operatorname{Rad}\left((\ell_{0\text{-}1} \circ \operatorname{sgn} \circ \mathcal{H}_B)\big|_S\right)$. The only way we really know how to deal with "peeling" off functions like that is Lipschitz functions, with Talagrand's lemma. But $\ell_{0\text{-}1} \circ \operatorname{sgn}$

For more, visit https://cs.ubc.ca/~dsuth/532D/23w1/.

isn't Lipschitz: it changes suddenly from 0 to 1 when the prediction shifts from zero to anything negative.

(In the VC derivation we pretended ℓ_{0-1} was Lipschitz, but we could only do that because our $\mathcal{H}$ mapped to $\{-1, 1\}$; we can't play any similar trick with sgn for $\mathcal{H}_B$ mapping to continuous values in $\mathbb{R}$.)

Another problem is that computing the ERM with respect to 0-1 loss, in the case where $L_{\mathcal{D}}(h^*) > 0$, is actually NP-hard [BS00]! (You can reduce a SAT variant to it.)

2 Surrogate Losses

We can work around both problems with *surrogate losses*.

One version we've already talked about is by using the logistic loss, which is 1-Lipschitz, so we can apply Talagrand. But this bounds things only in terms of the logistic loss. It turns out, though, that we'll be able to use this to say something about accuracy.

In particular, if $yh(x) > 0$, then $\ell_{0-1}(h, z) = 0 < \ell_{logistic}(h, z)$. Otherwise, if $yh(x) \le 0$, then $l_{logistic} \ge \log(1 + \exp(0)) = \log 2$, and so $\frac{1}{\log 2} l_{logistic}(h, z) \ge 1 \ge l_{0-1}(\operatorname{sgn} \circ h, z)$.

So, suppose that we have some loss ℓ_{surr} such that $\ell_{surr}(h, z) \ge \ell_{0-1}(\operatorname{sgn} \circ h, z)$ for all h, z. Then $L_{\mathcal{D}}^{surr}(h) = \mathbb{E}_z \ell_{surr}(h, z) \ge \mathbb{E}_z \ell_{0-1}(\operatorname{sgn} \circ h, z) = L_{\mathcal{D}}^{0-1}(\operatorname{sgn} \circ h)$. Thus, if we pick such a surrogate loss that's also ρ-Lipschitz and bounded in $[a, b]$, we get by Talagrand and our Rademacher concentration results that with probability at least $1 - \delta$,

$$L_{\mathcal{D}}^{0-1}(\operatorname{sgn} \circ h) \le L_{\mathcal{D}}^{surr}(h) \le L_S^{surr}(h) + 2\rho \mathop{\mathbb{E}}_S \operatorname{Rad}(\mathcal{H}|_{S_x}) + (b - a)\sqrt{\frac{1}{2m} \log \frac{1}{\delta}}.$$

Ideally, we'd have a surrogate loss that also makes ERM easy to solve with respect to that loss; if $L_S^{surr}(h)$ is small, this would give small 0-1 loss as well. Logistic regression is one, when $\|x\|$ is bounded: we can just multiply our previous bound by $\log 2$ and get a bound on the accuracy.

Logistic regression is a pretty loose upper bound, though: if h is really wrong, this ℓ_{surr} grows without bound, while $\ell_{0-1} \circ \operatorname{sgn}$ stays just 1. So let's look at a tighter analysis and not worry about solving it first.

3 Analysis with Ramp Loss

One natural way to get a bounded, 1-Lipschitz upper bound on the 0-1 loss is with the *ramp loss*

$$\ell_{ramp}(h, (x, y)) = l_y^{ramp}(h(x)) = \begin{cases} 1 & yh(x) \le 0 \\ 1 - yh(x) & 0 \le yh(x) \le 1 \\ 0 & 1 \le yh(x) \end{cases}.$$

That is, if we make an incorrect prediction, $\operatorname{sgn}(h(x)) \ne y$, we get 1 loss. If we make a correct prediction and are confident enough in it, $|h(x)| \ge 1$, we get 0 loss. But in between, when we're right but not very confident, we incur some partial loss. This is indeed an upper bound on the 0-1 loss, l_y^{ramp} is 1-Lipschitz, and it's bounded in

$[0,1]$, so we have with probability at least $1-\delta$ for all h in a real-valued $\mathcal{H}$ that

$$L_{\mathcal{D}}^{0-1}(\operatorname{sgn}\circ h) \le L_{\mathcal{D}}^{ramp}(h) \le L_S^{ramp}(h) + 2\mathop{\mathbb{E}}_S \operatorname{Rad}(\mathcal{H}|_{S_x}) + \sqrt{\frac{1}{2m}\log\frac{1}{\delta}}. \tag{1}$$

Now let's look at linear classifiers, and assume $\mathbb{E}\|x\|^2 \le C^2$. For predictors from $\mathcal{H}_B = \{x \mapsto w\cdot x : \|w\| \le B\}$, we have

$$L_{\mathcal{D}}^{0-1}(h) \le L_S^{ramp}(h) + \frac{2BC}{\sqrt{m}} + \sqrt{\frac{1}{2m}\log\frac{1}{\delta}}. \tag{2}$$

What about that ramp loss term?

One nice special case when the distribution is *separable with margin 1*, meaning that there's a w^* such that $\Pr_{(x,y)\sim\mathcal{D}}(yx\cdot w^* \ge 1) = 1$. Then we know that $\inf_{h\in\mathcal{H}_{\|w^*\|}} L_{\mathcal{D}}^{ramp}(h) = 0$. Plugging into (2) tells us that any predictor $\hat{h} = h_{\hat{w}} = \hat{w}\cdot x$ with $L_S^{ramp}(\hat{h}) = 0$ and $\|\hat{w}\| \le \|w^*\|$ has

$$L_{\mathcal{D}}^{0-1}(\operatorname{sgn}\circ\hat{h}) \le \frac{2C\|w^*\|}{\sqrt{m}} + \sqrt{\frac{1}{2m}\log\frac{1}{\delta}}. \tag{3}$$

Notice that if the distribution is separable with margin γ, scaling w by $1/\gamma$ makes it separable with margin 1. So, if the distribution is separable with any finite margin, then there is some $h^* = h_{w^*}$ that separates with margin 1, i.e. achieves $L_S^{ramp}(h^*) = 0$. Ramp-loss ERM on $\mathcal{H}_{\|w^*\|}$ would then necessarily get zero sample ramp loss, and achieve generalization 0-1 error of $\frac{2C\|w^*\|}{\sqrt{m}} + \sqrt{\frac{2}{m}\log\frac{2}{\delta}}$.

The problem is, though, we presumably don't know $\|w^*\|$ in advance, and so it's not obvious what $\mathcal{H}$ to use for ERM. We could do some version of SRM, but if the distribution is separable with a margin, we can actually still do ERM without explicitly knowing the hypothesis class beforehand by finding the *minimum-norm interpolator*:

$$\hat{w} = \arg\min \|w\| \quad \text{s.t. } L_S^{ramp}(h_w) = 0. \tag{4}$$

"Interpolator" in this context means "something that achieves zero sample error," generalizing the notion of, say, polynomial interpolation.

This will have $\|\hat{w}\| \le \|w^*\|$, so it is an ERM in for separable data, meaning it's a decent learning algorithm on separable data. We'll think about this algorithm more in a moment.

The actual value of the bound in (3), though, depends on $\|w^*\|$, which we don't know – the disadvantage of using an implicit $\mathcal{H}$! But we can use an argument like the one we used for SRM to get a bound that only depends on $\|\hat{w}\|$:

PROPOSITION 1. *Let $\mathbb{E}_{(x,y)\sim\mathcal{D}}\|x\|^2 \le C^2$, and $h_w(x) = \operatorname{sgn}(\hat{w}\cdot x)$. Then for any $\delta\in(0,1)$ and $r > 0$ fixed independent of the data, we have with probability at least $1-\delta$ over the choice of sample $S\sim\mathcal{D}^m$ that for all $w\in\mathbb{R}^d$,*

$$L_{\mathcal{D}}^{0-1}(\operatorname{sgn}\circ h_w) \le L_S^{ramp}(h_w) + \frac{1}{\sqrt{m}}\left[\sqrt{\tfrac{1}{2}\log\tfrac{2}{\delta}} + \begin{cases} 4Cr & \text{if } \|w\| \le r \\ 4C\|w\| + \sqrt{\log\log_2\frac{2\|w\|}{r}} & \text{if } \|w\| \ge r \end{cases}\right].$$

*You can think of either $r = 1$ or r small; it's an annoying technicality. (The best choice is $r = \|\hat{w}\|$, but we can't choose it based on data.) Theorem 26.14 of [SSBD] doesn't have it, but that's because **that theorem is wrong**.*

Proof. Define $B_k = r2^k$ and $\delta_k = \frac{6\delta}{\pi^2k^2}$ for all $k \ge 1$, noting $\sum_{k=1}^{\infty}\delta_k = \delta$. For each k, it

holds with probability at least $1 - \delta_k$ that

$$\forall h \in \mathcal{H}_{B_k}. \quad L_{\mathcal{D}}^{0-1}(\operatorname{sgn} \circ h) \le L_{\mathcal{D}}^{ramp}(h) \le L_S^{ramp}(h) + \frac{2B_k C}{\sqrt{m}} + \sqrt{\frac{1}{2m} \log \frac{1}{\delta_k}}. \tag{5}$$

This bit of the analysis would work for $\|w\| \le 2r$, but the final bound will be continuous (though slightly looser) if we switch at norm r instead.

For any $h = h_w$ with $\|w\| \le r = \frac{1}{2}B_1$, we have $h \in \mathcal{H}_{B_1}$, which has $\delta_1 = 6\delta/\pi^2$. Upper-bounding $\pi^2/6 \approx 1.64$ by 2 for simplicity, loosening (5) for $\mathcal{H}_{B_1}$ slightly gives

$$L_{\mathcal{D}}^{0-1}(\operatorname{sgn} \circ h) \le L_S^{ramp}(h) + \frac{4Cr}{\sqrt{m}} + \sqrt{\frac{1}{2m} \log \frac{2}{\delta}}.$$

Otherwise, we have $\|w\| > r$. Let $k_w = \left\lceil \log_2 \frac{\|w\|}{r} \right\rceil \ge 1$; then

$$B_{k_w} = r2^{k_w} = r2^{\left\lceil \log_2 \frac{\|w\|}{r} \right\rceil} < r2^{1+\log_2 \frac{\|w\|}{r}} = 2\|w\|,$$

so that $h \in \mathcal{H}_{B_{k_w}}$. Also,

$$\frac{1}{\delta_{k_w}} = \frac{\pi^2 k_w^2}{6\delta} = \frac{\pi^2/6}{\delta} \left\lceil \log_2 \frac{\|w\|}{r} \right\rceil^2.$$

Using that $\pi^2/6 < 2$ and $\lceil \log_2 a \rceil < \log_2(a) + 1 = \log_2(2a)$,

$$\log \frac{1}{\delta_{i_w}} \le \log \frac{2}{\delta} + 2 \log \log_2 \frac{2\|w\|}{r}.$$

Thus (5) shows, with a slight loosening for simplicity,

$$L_{\mathcal{D}}^{0-1}(\operatorname{sgn} \circ h) \le L_S^{ramp}(h) + \frac{4C\|w\|}{\sqrt{m}} + \sqrt{\frac{1}{2m} \log \frac{2}{\delta}} + \sqrt{\frac{1}{m} \log \log_2 \frac{2\|w\|}{r}}.$$

Unioning these bounds over all $k \ge 1$ gives the desired result. □

If we pick an r that's much smaller than any reasonable $\|\hat{w}\|$ but not so small that $\log \log_2 \frac{1}{r}$ is significant, we get for $\hat{w}$ that separate the sample S with margin 1 that, roughly, $L_{\mathcal{D}}^{0-1}(\hat{w}) \approx \mathcal{O}_p(\|\hat{w}\|/\sqrt{m})$. This reinforces that the minimum-norm interpolator seems like a good idea, when we think the distribution is separable with a margin.

4 RAMP INTERPOLATION = HARD SVM = MAX MARGIN

How can we find this min-norm interpolator, and what does it mean?

Expanding out the definition of L_S^{ramp}, (4) is equivalent to

$$\hat{h} = h_{\hat{w}}; \qquad \hat{w} \in \arg\min_w \|w\|^2 \quad \text{s.t. } \forall i \in [m], \quad y_i w \cdot x_i \ge 1. \tag{HardSVM}$$

This form is a *convex quadratic program*, a well-studied class of optimization problems. This is known as a *(hard) support vector machine* (SVM).

The usual motivation for SVMs is in terms of margin maximization. We can see this

by noting that (HardSVM) is equivalent to

$$\begin{aligned}
\hat{w} &= \arg\max_w \frac{1}{\|w\|} && \text{s.t. } \forall i \in [m],\ \ y_i w \cdot x_i \geq 1 \\
&= \arg\max_w \frac{1}{\|w\|} \min_{i\in[m]} w \cdot x_i && \text{s.t. } \forall i \in [m],\ \ y_i w \cdot x_i \geq 1 \\
&\subseteq \arg\max_w \min_{i\in[m]} \frac{w \cdot x_i}{\|w\|} && \text{s.t. } \forall i \in [m],\ \ y_i w \cdot x_i > 0.
\end{aligned}$$

In the second line, $\min_{i\in[m]} w \cdot x_i$ will equal 1 for any optimal w: it must be at least 1 for the constraint to hold, and if it were bigger we could just scale down w to also scale down all the predictions, which would improve the objective while keeping the constraints valid.

In the third line, the objective is invariant to scaling w by a constant, so any multiple of a w that minimized the second line will minimize the third line.

Also, if we scale any minimizer for the third line by $\min_{i\in[m]} y_i w \cdot x_i$, we'll get a minimizer for the second line. Note that scaling by a positive constant doesn't change the hard classifier, $\operatorname{sgn}(w \cdot x) = \operatorname{sgn}(cw \cdot x)$ for $c > 0$; it just changes our confidence score.

The quantity $w \cdot x_i / \|w\|$ is the *geometric margin* of the point x_i: it's the distance of x_i from the hyperplane $\{z : w \cdot z = 0\}$. (For a formal proof of this fact, see Claim 15.1 of [SSBD].)

So, (HardSVM) maximizes the worst-case geometric margin on the training set, and anything maximizing the geometric margin will be a multiple of a solution to (HardSVM).

Thus, if $L_S^{ramp}(w) = 0$, then $\frac{1}{\|w\|}$ is at least the worst-case geometric margin (with equality for the max-margin solution).

For a graphical illustration of these concepts, see Figures 5.1 to 5.3 of [MRT].

I should probably add similar illustrations here.

Note that, as a convex QP, we can solve (HardSVM) in polynomial time – e.g. with a generic interior point algorithm [YT89], although there are many specialized solvers and other possibilities. Thus, if $m \geq \frac{1}{\varepsilon^2}\left[2C\|w^*\| + \sqrt{2\log\frac{2}{\delta}}\right]^2$ then we efficiently achieve 0-1 loss less than ε with probability at least $1 - \delta$.

This doesn't violate NP-hardness for 0-1 loss ERM, since it's only for separable distributions. It also doesn't contradict our VC dimension lower bounds, since we have two assumptions on $\mathcal{D}$ here: separability with a margin and the bound C on the norm of the data. (It doesn't even establish nonuniform learning, because of the dependence on C.)

5 HINGE LOSS AND SOFT SVM

When the data isn't linearly separable, (HardSVM) will just fail: the constraints aren't achievable, so it's minimizing over an empty set.

A natural idea is to try to trade off between having a small $L_S^{ramp}(h)$ and a small $\|w\|$. For example, like in SRM, we could try to minimize the upper bound in Proposition 1. We could try to literally do that, but choosing an r and whatnot is annoying, so we might prefer to avoid worrying about. Being a little fuzzy, pretend we pick an r small enough that $\max\{r, \|w\|\} = \|w\|$ for any "reasonable" w but not so

small that $\sqrt{\log \log_2 \frac{1}{r}}$ is relevant to anything. Also, the $\sqrt{\log \log_2 \|w\|}$ term is also not going to be at all relevant compared to the $\|w\|$ term. So, it seems reasonable to try to pick

$$\arg\min_w \mathrm{L}_{\mathrm{S}}^{ramp}(h_w) + \frac{4\mathrm{C}}{\sqrt{m}}\|w\|.$$

Unfortunately, solving this problem is still NP-hard [MI15, Theorem 2.3].

To dodge this, we can *again* take a surrogate loss, $\ell_{hinge} \geq \ell_{ramp} \geq \ell_{0-1}$ given by

$$\ell_{hinge}(h,(x,y)) = l^{hinge}(h(x)) = \begin{cases} 1 - yh(x) & \text{if } yh(x) \leq 1 \\ 0 & \text{if } yh(x) \geq 1. \end{cases}$$

This is like the ramp loss, except once it starts going, it never stops: you get more loss for a more-confident wrong answer. This loss is still 1-Lipschitz, but it's not bounded. More importantly, though, it's *convex*, which makes it easy to optimize. (We'll talk more about convexity shortly.)

5.1 *Hinge loss ERM with bounded weights*

It then makes sense to try to have both small $\mathrm{L}_{\mathrm{S}}^{hinge}$ and small $\|w\|$. We can see from (2) that, for example, if

$$\hat{h}_{\mathrm{B}} = \arg\min_{h \in \mathcal{H}_{\mathrm{B}}} \mathrm{L}_{\mathrm{S}}^{hinge}(h_w), \tag{6}$$

since $\ell_{hinge} \geq \ell_{ramp}$, with probability at least $1 - \delta$

$$\mathrm{L}_{\mathcal{D}}^{0-1}(\mathrm{sgn} \circ \hat{h}_{\mathrm{B}}) \leq \mathrm{L}_{\mathrm{S}}^{hinge}(\hat{h}_{\mathrm{B}}) + \frac{2\mathrm{BC}}{\sqrt{m}} + \sqrt{\frac{1}{2m}\log\frac{1}{\delta}}.$$

While ℓ_{hinge} is unbounded, we know that $\sup_{h,x}|h(x)| \leq 1 + \sup_{h,x}\|w\|\|x\|$. Thus if we strengthen our assumption on $\mathcal{D}$ to $\Pr(\|x\| \leq \mathrm{C}) = 1$, our usual route of $\mathrm{L}_{\mathrm{S}}^{hinge}(\hat{h}_{\mathrm{B}}) \leq \mathrm{L}_{\mathrm{S}}^{hinge}(h^*)$, applying Hoeffding to $\mathrm{L}_{\mathrm{S}}^{hinge}(h^*)$, and choosing the best h^* yields

$$\mathrm{L}_{\mathcal{D}}^{0-1}(\mathrm{sgn} \circ \hat{h}_{\mathrm{B}}) \leq \inf_{h \in \mathcal{H}_{\mathrm{B}}} \mathrm{L}_{\mathcal{D}}^{hinge}(h) + \frac{2\mathrm{BC}}{\sqrt{m}} + (2 + \mathrm{BC})\sqrt{\frac{1}{2m}\log\frac{2}{\delta}}.$$

5.2 *Bound minimization*

Rather than picking a hard constraint B *a priori*, which might be difficult, we could do something SRM-like with Proposition 1 and let $\hat{h} = h_{\hat{w}}$ minimize

$$\mathrm{L}_{\mathrm{S}}^{hinge}(h_w) + \frac{1}{\sqrt{m}} \begin{cases} 4\mathrm{C}r & \text{if } \|w\| \leq r \\ 4\mathrm{C}\|w\| + \sqrt{\log\log_2 \frac{2\|w\|}{r}} & \text{if } \|w\| > r. \end{cases} \tag{7}$$

Then, like in SRM, we know that quantity is minimized for $\hat{w}$, and so can say that, for any arbitrary $h^* = h_{w^*}$, assigning $\frac{2}{3}\delta$ failure probability for the bound of Proposition 1 and $\frac{1}{3}\delta$ probability for a Hoeffding bound on $\mathrm{L}_{\mathrm{S}}^{hinge}(h^*)$, and again

assuming that $\|x\| \le C$ a.s.,

$$L_{\mathcal{D}}^{0-1}(\operatorname{sgn} \circ \hat{h}) \le L_S^{hinge}(\hat{h}) + \frac{1}{\sqrt{m}}\left[\sqrt{\tfrac{1}{2}\log\tfrac{3}{\delta}} + \begin{cases} 4Cr & \text{if } \|\hat{w}\| \le r \\ 4C\|\hat{w}\| + \sqrt{\log\log_2 \frac{2\|\hat{w}\|}{r}} & \text{if } \|\hat{w}\| \ge r \end{cases}\right]$$

$$\le L_S^{hinge}(h^*) + \frac{1}{\sqrt{m}}\left[\sqrt{\tfrac{1}{2}\log\tfrac{3}{\delta}} + \begin{cases} 4Cr & \text{if } \|w^*\| \le r \\ 4C\|w^*\| + \sqrt{\log\log_2 \frac{2\|w^*\|}{r}} & \text{if } \|w^*\| \ge r \end{cases}\right]$$

$$\le L_{\mathcal{D}}^{hinge}(h^*) + \frac{1}{\sqrt{m}}\left[(2 + C\|w^*\|)\sqrt{\tfrac{1}{2}\log\tfrac{3}{\delta}} + \begin{cases} 4Cr & \text{if } \|w^*\| \le r \\ 4C\|w^*\| + \sqrt{\log\log_2 \frac{2\|w^*\|}{r}} & \text{if } \|w^*\| \ge r \end{cases}\right],$$

where here h^* is any arbitrary predictor in $\mathcal{H}$. This is like a nonuniform learning bound, but only for $\mathcal{D}$ satisfying $\|x\| \le C$.

If $\mathcal{D}$ is separable, we can take h^* to be the minimum-norm predictor achieving zero distribution hinge loss, in which case we know that (7) has zero-one error decaying like $1/\sqrt{m}$. Again assuming that r is small but not so small that the $\log\log_2$ term is meaningfully non-constant, $L_{\mathcal{D}}(\operatorname{sgn} \circ \hat{h}) \approx \mathcal{O}_p\left(\frac{C\|w^*\|}{\sqrt{m}}\right) = \mathcal{O}_p\left(\frac{C}{\text{margin}\cdot\sqrt{m}}\right)$

5.3 *Soft SVM*

Unfortunately, the optimization problem (7) is kind of a huge pain. It's not even convex, because of the $\sqrt{\log\log_2 \|w\|}$ term. Again, we can reason that we can probably ignore r and the $\sqrt{\log\log_2 \|w\|}$ term, and argue for instead minimizing the nearly-equivalent

$$L_S^{hinge}(h_w) + \frac{4C}{\sqrt{m}}\|w\|.$$

It's not obvious that these bounds are especially tight, though, so maybe $\frac{4C}{\sqrt{m}}$ isn't the right constant to trade off between the loss and $\|w\|$. Also, it turns out to be more convenient to minimize with $\|w\|^2$ rather than $\|w\|$. *Soft SVM*s use the squared norm of w and replace $4C/\sqrt{m}$ with a hyperparameter λ:

$$\hat{h}_\lambda = h_{\hat{w}_\lambda}; \qquad \hat{w}_\lambda \in \arg\min_w L_S^{hinge}(h_w) + \lambda\|w\|^2. \tag{SoftSVM}$$

(In the version with an intercept b, we typically *don't* add λb^2 to the loss; this is one difference from the homogeneous reduction.)

The constrained ERM (6) and the regularized (SoftSVM) are in fact dual to each other, in the sense that for any B there is some λ such that $\hat{h}_B$'s weight vector agrees with $\hat{w}_\lambda$, and vice versa. (We can't just write down a given B for a given λ or vice-versa, though, unfortunately.)

If you know convex optimization: set up the Lagrangian of either problem; strong duality holds via Slater's condition.

Soft SVMs also have a nice motivation in terms of margin maximization. If h_w classifies a point x correctly with margin at least 1, then it doesn't contribute to the objective at all. If it's "inside" the margin or even misclassified, though, we get loss equal to the distance by which we're on the wrong side of the margin. One way to consider this is as a hard SVM on a modified problem, where we drag points around to be on the margin, and penalize how much dragging around we need to do.

The classic framing is $C\,L_S^{hinge}(h_w) + \|w\|^2$; there the penalty for moving points around is C. You can think of $C = \frac{1}{\lambda}$.

In the limit as $\lambda \to 0$, on separable data, (SoftSVM) becomes (HardSVM). Soft

SVMs with a nonzero λ might give different results from hard SVMs, though, even on separable data: they might allow a few points to violate a bigger "theoretical" margin.

To analyze $\hat{w}_\lambda$ directly, we can still use Proposition 1 and (if we like) bound the ramp loss by the hinge loss: the result holds for all linear predictors. This gives us an upper bound on $L_{\mathcal{D}}^{0-1}(\hat{h}_\lambda)$ in terms of $L_S^{hinge}(\hat{h}_\lambda)$ and $\|\hat{w}_\lambda\|$. It's more difficult to relate this to the loss of a comparison hypothesis h^*, though we can maybe take some solace in (SoftSVM) being similar to (7), which does have an actual bound.

Or, instead, we can use stability bounds (discussed soon).

6 SVM DUALITY

The following stuff is historically very important, serves as a nice segue into our next topic, explains the name "support vector machine," and introduces an area of math that's profoundly important to optimization / often useful in theory / beautiful in its own right. It's not, however, as practically important as it once was.

6.1 *Hard SVM*

Starting from (HardSVM), we can rewrite these hard constraints by introducing *dual variables* α_i for $i \in [m]$:

$$\min_w \frac{1}{2}\|w\|^2 \text{ s.t. } \forall i.\ y_i w \cdot x_i \geq 1 \ = \ \min_w \max_{\alpha_i \geq 0} \frac{1}{2}\|w\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i w \cdot x_i).$$

If any of the $y_i w \cdot x_i < 1$, then the inner maximizer can drive $\alpha_i \to \infty$ and make the objective arbitrarily big; the outer minimizer, then, can't allow that to happen. For any $y_i w \cdot x_i > 1$, the inner maximizer will prefer to pick $\alpha_i = 0$. If any are exactly $y_i w \cdot x = 1$, then it doesn't matter what α_i it picks.

Reminder: start from $\max_y f(x,y) \geq f(x,y')$.

We've already used in class that $\min_x \max_y f(x,y) \geq \max_y \min_x f(x,y)$. In this setting, this is called *weak (Lagrangian) duality*. In this case, though, we actually have *strong* duality via something called *Slater's condition*: swapping the min and the max doesn't change the value.

Slater's condition holds when the objective is convex, any equality constraints are affine, and inequality constraints are convex and "strictly feasible." Here, if any w is feasible, $2w$ will be strictly feasible; if nothing is feasible, the convention is that $\min\{\} = \infty$ *and the RHS will also always be* ∞.

$$\min_w \frac{1}{2}\|w\|^2 \text{ s.t. } \forall i.\ y_i w^\mathsf{T} x_i \geq 1 \ = \ \max_{\alpha_i \geq 0} \min_w \frac{1}{2}\|w\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i w \cdot x_i).$$

The inner minimization in w is differentiable and unconstrained, so we can find its value by setting the gradient to zero:

$$w + \sum_{i=1}^m (-\alpha_i y_i x_i) = 0 \quad \Rightarrow \quad w = \sum_{i=1}^m \alpha_i y_i x_i \quad \Rightarrow \quad \sum_{i=1}^m \alpha_i y_i x_i \cdot w = \|w\|^2,$$

which also implies that

$$\|w\|^2 = \sum_{i=1}^m \sum_{j=1}^m \alpha_i y_i x_i^\mathsf{T} x_j y_j \alpha_j = \alpha^\mathsf{T} \operatorname{diag}(y) \mathbf{X}\mathbf{X}^\mathsf{T} \operatorname{diag}(y) \alpha.$$

Here $\alpha \in \mathbb{R}^m$ is the vector of α_is, $\operatorname{diag}(y) \in \mathbb{R}^{m \times m}$ is a matrix with y_i as its (i,i)th entry and zero off-diagonal, and $\mathbf{X} \in \mathbb{R}^{m \times d}$ has ith row x_i. Thus we've shown that

(HardSVM) is equivalent to

$$\max_{\alpha_i \geq 0} \mathbf{1}^\mathsf{T}\alpha - \tfrac{1}{2}\alpha^\mathsf{T} \operatorname{diag}(y)\mathbf{X}\mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha. \tag{HardSVM'}$$

Once we find α, we can recover w as $\mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha$, meaning that

$$h_w(x) = \alpha^\mathsf{T} \operatorname{diag}(y)\mathbf{X}x = \sum_{i=1}^{m} \alpha_i y_i x_i \cdot x.$$

This is called the *dual form* of (HardSVM). We've transformed the *primal* form, a constrained optimization over $w \in \mathbb{R}^b$, to an unconstrained optimization over $\alpha \in \mathbb{R}^m_{\geq 0}$. We can solve this with any of several algorithms: it's also a convex quadratic program, and there are many specialized algorithms for (HardSVM') in particular, but since the constraints are simple we can also think about easy things like projected gradient descent.

Support vectors (HardSVM') also motivates the name *support vector machine*. As we mentioned when we first introduced the dual variables, if $y_i w^\mathsf{T} x_i > 1$ for some i, then we necessarily have $\alpha_i = 0$ at optimum. We can only have $\alpha_i \neq 0$ if $y_i w^\mathsf{T} x_i = 0$, i.e. the point (x_i, y_i) is exactly on the margin of the hard SVM. These points are called support vectors, because they "support" the position of the margin. This sparsity in the solution has some other nice consequences as well, e.g. computationally.

This is called complementary slackness *in the KKT conditions.*

6.2 *Soft SVM duality*

Not covered in class, but you might want to look at.

Start by introducing auxiliary variables ξ_i accounting for the hinge loss in (SoftSVM), then go through the same kind of argument, where now we'll additionally have dual variables β for the nonnegativity constraints on ξ. We're also going to use our dual variables for the margin constraints as $2\lambda\alpha_i$ instead of just α_i, because it just makes stuff work out nicer in the end.

$$\begin{aligned}
\min_w \mathrm{L}_\mathrm{S}^{hinge} + \lambda \|w\|^2 &= \min_{w,\xi} \lambda \|w\|^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i \quad \text{s.t. } \forall i,\ y_i w \cdot x_i \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \\
&= \min_{\xi \geq 0, w} \max_{\alpha \geq 0, \beta \geq 0} \lambda \|w\|^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} 2\lambda\alpha_i(1 - y_i w \cdot x_i - \xi_i) - \sum_{i=1}^{m} \beta_i \xi_i \\
&= \max_{\alpha \geq 0, \beta \geq 0} \min_{\xi \geq 0, w} \lambda \|w\|^2 + \frac{1}{m}\mathbf{1}^\mathsf{T}\xi + 2\lambda\alpha^\mathsf{T}[\mathbf{1} - \operatorname{diag}(y)\mathbf{X}w - \xi] - \beta^\mathsf{T}\xi.
\end{aligned}$$

Setting the w gradient to zero, $2\lambda w - 2\lambda\mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha = 0$ and again $w = \mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha$. Plugging this in, and rearranging, we get

$$\min_w \mathrm{L}_\mathrm{S}^{hinge} + \lambda \|w\|^2 = \max_{\alpha \geq 0, \beta \geq 0} \min_{\xi \geq 0} 2\lambda\mathbf{1}^\mathsf{T}\alpha - \lambda\alpha^\mathsf{T} \operatorname{diag}(y)\mathbf{X}\mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha + \left(\tfrac{1}{m}\mathbf{1} - 2\lambda\alpha - \beta\right)^\mathsf{T} \xi.$$

Now, if $\beta_i = \frac{1}{m} - 2\lambda\alpha_i$, then the corresponding part of that final term is zero regardless of the value of ξ_i. If any β_i are smaller than that, then the inner minimizer will pick $\xi_i = 0$, and that component will still be zero. If any β_i are bigger than that, though, choosing $\xi_i \to \infty$ will give the value $-\infty$ for the inner minimizer, so the outer maximizer can't allow that. Since the only thing β_i needs to do is be

nonnegative and satisfy that constraint, the diverging situation can be avoided if $2\lambda\alpha_i \le 1/m$, and then the value of β doesn't actually matter anymore. Thus, the problem finally simplifies to

$$\min_w \mathrm{L}_S^{hinge} + \lambda\|w\|^2 = 2\lambda \max_{0\le\alpha_i\le\frac{1}{2\lambda m}} \mathbf{1}^\top\alpha - \tfrac{1}{2}\lambda\alpha^\top \operatorname{diag}(y)\mathbf{X}\mathbf{X}^\top \operatorname{diag}(y)\alpha. \quad \text{(SoftSVM')}$$

Remarkably, this is *exactly* (HardSVM') with an extra upper bound on α.

Using the same kind of argument as we made for support vectors earlier, we can see that indeed $\xi_i = 0$ unless $y_i w\cdot x_i < 1$: we only "move the input points" if we need to. For these points, $\beta_i = 0$, meaning that $\alpha_i = \frac{1}{2\lambda m}$, and we can immediately tell which points are misclassified or classified correctly with too small a margin. Any points with $0 < \alpha_i < \frac{1}{2\lambda m}$ have $\xi_i = 0$ but $y_i w\cdot x_i = 1$, and so lie exactly on the margin as before.

6.3 *Including an intercept*

Not covered in class, but you might want to look at.

So far, we've been assuming that intercept terms, $\operatorname{sgn}(w\cdot x + b)$ rather than $\operatorname{sgn}(w\cdot x)$, are handled via $\tilde{w} = [b, w]$, $\tilde{x} = [1, x]$. But then note that $\|\tilde{w}\|^2 = b^2 + \|w\|^2$: we're regularizing the intercept as well, which isn't motivated in terms of the geometric margin and is also counter to usual statistical practice. So, it's maybe worth figuring out what happens if we explicitly include b and don't regularize it.

Compared to the derivation of (SoftSVM'), the constraint is $y_i(w\cdot x_i + b) \ge 1 - \xi_i$, which only adds a term $2\lambda\alpha_i y_i b$. This doesn't affect the optimization for w or ξ; if $\alpha\cdot y = 0$, it doesn't affect the optimization, but otherwise the inner minimizer could drive things to $-\infty$. Thus the dual becomes slightly modified, to

$$\max_{0\le\alpha_i\le\frac{1}{2\lambda m} \text{ and } \alpha\cdot y=0} \mathbf{1}^\top\alpha - \frac{1}{2}\alpha^\top \operatorname{diag}(y)\mathbf{X}\mathbf{X}^\top \operatorname{diag}(y)\alpha.$$

Our final predictor is $w\cdot x + b = \alpha^\top \operatorname{diag}(y)\mathbf{X}x + b$, so we still need to figure out the value of b. But note that, for points with $0 < \alpha_i < \frac{1}{2\lambda m}$, we know that $y_i(w\cdot x_i + b) = 1$: so, once we've found α, we can just pick any such i and set $b = y_i - w\cdot x_i = y_i - \alpha^\top \operatorname{diag}(y)\mathbf{X}x_i$.

7 Aside: Margin Analysis

Not covered in class, but you might want to look at.

The following is a slightly different way to frame ramp loss analysis that can sometimes be easier to think about. It's also more natural to look at for general hypothesis classes. It's based on the ρ-margin loss, which gives us full credit if our confidence is at least ρ:

$$\ell_{\rho\text{-}margin}(h,(x,y)) = l_y^{\rho-margin}(h(x)) = \begin{cases} 1 & \text{if } yh(x) \le 0 \\ 1 - \frac{yh(x)}{\rho} & \text{if } 0 \le yh(x) \le \rho \\ 0 & \text{if } yh(x) \ge \rho. \end{cases}$$

This upper-bound to 0-1 loss ramps at ρ instead of 1, and is $\frac{1}{\rho}$-Lipschitz. So, the analogue of (1) is that for any fixed ρ, with probability at least $1-\delta$, we have for any $\mathcal{H}$ of real-valued hypotheses that

$$\forall h\in\mathcal{H},\quad \mathrm{L}_{\mathcal{D}}^{0-1}(\operatorname{sgn}\circ h) \le \mathrm{L}_S^{\rho-margin}(h) + \frac{2}{\rho}\mathop{\mathbb{E}}_{S'\sim\mathcal{D}^m}\operatorname{Rad}(\mathcal{H}|_{S'_x}) + \sqrt{\frac{1}{2m}\log\frac{1}{\delta}}. \quad (8)$$

We can avoid committing to a particular margin, similarly to in Proposition 1. Another difference is that now we don't have to assume $\mathcal{H}_B$, but allow general real-valued, but *fixed*, $\mathcal{H}$.

PROPOSITION 2. *Let $\mathcal{H}$ contain functions mapping to $\mathbb{R}$, and fix some $r > 0$. Then for any $\delta \in (0,1)$, we have with probability at least $1-\delta$ over the choice of $S \sim \mathcal{D}^m$ that it holds for all $h \in \mathcal{H}$ and $\rho \in (0, r]$ that*

$$L_{\mathcal{D}}^{0-1}(\operatorname{sgn} \circ h) \le L_S^{\rho-margin}(h) + \frac{4}{\rho} \mathop{\mathbb{E}}_{S' \sim \mathcal{D}^m} \operatorname{Rad}(\mathcal{H}|_{S'_x}) + \sqrt{\frac{1}{m} \log \log_2 \frac{2r}{\rho}} + \sqrt{\frac{1}{2m} \log \frac{2}{\delta}}.$$

Proof. Let $\rho_k = r2^{-k}$ for all $k \ge 0$, and $\delta_k = \frac{6\delta}{\pi^2 k^2}$ for $k \ge 1$; note that $\sum_{k=1}^{\infty} \delta_k = \delta$. By (8), it holds with probability at least $1 - \delta_k$ for each ρ_k that

$$\forall h \in \mathcal{H}, \quad L_{\mathcal{D}}^{0-1}(\operatorname{sgn} \circ h) \le L_S^{\rho_k-margin}(h) + \frac{2}{\rho_k} \mathop{\mathbb{E}}_{S' \sim \mathcal{D}^m} \operatorname{Rad}(\mathcal{H}|_{S'_x}) + \sqrt{\frac{1}{2m} \log \frac{1}{\delta_k}}.$$

For any $\rho \in (0, r]$, the smallest k such that $\rho_k \le \rho$ is given by $k = \left\lceil \log_2 \frac{r}{\rho} \right\rceil$.

We have $\ell_{\rho'-margin} \le \ell_{\rho-margin}$ for any $\rho' \le \rho$, so $L_S^{\rho_k-margin}(h) \le L_S^{\rho-margin}(h)$.

We also know that $\rho \le \rho_{k-1} = 2\rho_k$, so $\frac{1}{\rho_k} \le \frac{2}{\rho}$.

Finally, from $\log \frac{1}{\delta_k} = \log \frac{\pi^2}{6\delta} + 2 \log \log_2 \left\lceil \log_2 \frac{r}{\rho} \right\rceil$ we use that $\pi^2/6 < 2$ and $\lceil \log_2 a \rceil < \log_2(a) + 1 = \log_2(2a)$. □

We do have to commit to some predefined upper bound on the margin r, but the resulting bound only depends on it through $\sqrt{\log \log_2 r}$ so we can pick something big. (This r corresponds to $\frac{1}{\gamma}$ from Proposition 1.)

In this bound, unlike Proposition 1, we always consider a fixed $\mathcal{H}$ (hence why it can be generic) – but the trade-off in our analysis is between that $L_S^{\rho\text{-margin}}$, which decreases as ρ shrinks (we only try to get a smaller margin), versus the $\frac{1}{\rho}$ terms, which increase as ρ shrinks.

Because $\mathcal{H}$ is fixed, we can't easily use this to analyze the minimum-norm interpolator (HardSVM). But the bound holds for any $\mathcal{H}$, not just linear predictors.

REFERENCES

[BS00] Shai Ben-David and Hans Ulrich Simon. "Efficient learning of linear perceptrons." *Advances in Neural Information Processing Systems*. 2000.

[MI15] Søren Frejstrup Maibing and Christian Igel. "Computational Complexity of Linear Large Margin Classification With Ramp Loss." *AISTATS*. 2015.

[MRT] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talkwalkar. *Foundations of Machine Learning*. 2nd edition. MIT Press, 2018.

[SSBD] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[YT89] Yinyu Ye and Edison Tse. "An extension of Karmarkar's projective algorithm for convex quadratic programming." *Mathematical Programming* 44 (1989), pages 157–159.