CPSC 532D — 8. STRUCTURAL RISK MINIMIZATION

Danica J. Sutherland

University of British Columbia, Vancouver

Fall 2023

Recall the decomposition of error we made back in lecture 2:

$$\underbrace{L_{\mathcal{D}}(\hat{h}_S) - L_{bayes}}_{\text{excess error}} = \underbrace{L_{\mathcal{D}}(\hat{h}_S) - \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h)}_{\text{estimation error}} + \underbrace{\inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) - L_{bayes}}_{\text{approximation error}}.$$

We've talked a lot about the estimation error of ERM, bounding it in terms of Rademacher complexity or (when applicable) VC dimension. What we haven't really talked about yet is the approximation error: we drew some examples with polynomials, but if we don't know what the optimal predictor looks like... what should we do?

There are some particular cases where we can analyze this approximation error gap mathematically, if we assume things about the form of $\mathcal{D}$. But those assumptions usually rely on constants that are hard to know for any specific problem, and there's not usually a clear way to estimate them (or the Bayes error) from data, either.

The practical solution is generally to just try a bunch of different $\mathcal{H}$ and/or a bunch of different learning algorithms, then pick the best based on a validation set V. This is a good idea in practice, and we can make some theoretical guarantees on its generalization based on L_V being close to $L_{\mathcal{D}}$. But it's still hard to use that approach to handle the approximation error.

1 STRUCTURAL RISK MINIMIZATION

SRM says: let's use a *huge* $\mathcal{H}$, one where the approximation error is going to be small, maybe even zero. This will probably mean $\mathcal{H}$ has infinite VC dimension, large Rademacher complexity, etc. But let's decompose

$$\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2 \cup \cdots = \bigcup_{k\in\mathbb{N}} \mathcal{H}_k.$$

For instance, we might have $\mathcal{H}_k$ the set of decision trees of depth k, the set of degree-k polynomials, or the set of linear classifiers with $\|w\| \le 2^k$. We're going to assume that *each* $\mathcal{H}_k$ has uniform convergence:

$$\forall k \in \mathbb{N}. \quad \Pr_{S\sim\mathcal{D}^m}\left(\sup_{h\in\mathcal{H}_k} L_{\mathcal{D}}(h) - L_S(h) \le \varepsilon_k(m,\delta) \right) \ge 1-\delta \tag{1}$$

for functions ε_k satisfying that for all k and all $\delta \in (0,1)$, $\lim_{m\to\infty} \varepsilon_k(m,\delta) = 0$.

We'll also need a set of weights $w_k \ge 0$ such that $\sum_{k=1}^{\infty} w_k \le 1$; a typical choice is $6/(\pi^2 k^2) \approx 0.61/k^2$, since $\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$.

This is the problem that made Euler famous.

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

PROPOSITION 1. *Let $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2 \cup \dots$ satisfy (1), and let $w_k \geq 0$ have $\sum_{k=1}^{\infty} w_k \leq 1$. Then for any $\mathcal{D}$, with probability at least $1 - \delta$ over the choice of $\mathrm{S} \sim \mathcal{D}^m$, we have*

$$\forall h \in \mathcal{H}. \quad \mathrm{L}_{\mathcal{D}}(h) \leq \mathrm{L}_{\mathrm{S}}(h) + \min_{k : h \in \mathcal{H}_k} \varepsilon_k(m, \delta w_k).$$

Proof. We do a union bound over the $\mathcal{H}_k$, allocating δw_1 probability that anything in $\mathcal{H}_1$ violates the bound, δw_2 that anything in $\mathcal{H}_2$ does, and so on. Thus the total probability anything in $\mathcal{H}$ violates it is at most $\sum_k \delta w_k \leq \delta$. □

SRM is then the algorithm that minimizes this upper bound on $\mathrm{L}_{\mathcal{D}}(h)$:

DEFINITION 2. Given bounds on a decomposition of $\mathcal{H}$ as in (1), and weights $w_k \geq 0$ with $\sum w_k \leq 1$ and $\bigcup_{k : w_k > 0} \mathcal{H}_k = \mathcal{H}$, *structural risk minimization* is given by

$$\mathrm{SRM}_{\mathcal{H},\delta}(\mathrm{S}) \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \left[\mathrm{L}_{\mathrm{S}}(h) + \varepsilon_{k_h}(m, \delta w_{k_h}) \right] \qquad \text{where } k_h \in \operatorname*{arg\,min}_{k : h \in \mathcal{H}_k} \varepsilon_k(m, w_k \delta).$$

Typically, $k_h = \min\{k : h \in \mathcal{H}_k\}$.

We can implement this minimization by a finite number of calls to an "ERM oracle", as long as our loss is lower-bounded by $a \leq \ell(h, z)$ (typically $a = 0$):

```
function SRM$_{\mathcal{H},\delta}$(S)
    best $\leftarrow \infty$
    for $k = 1, 2, \dots$ do
        $h_k \leftarrow$ ERM$_{\mathcal{H}_k}$(S)
        cand_loss $\leftarrow \mathrm{L}_\mathrm{S}(h_k) + \varepsilon_k(m, w_k \delta)$
        if cand < best then
            $\hat{h} \leftarrow h_k$
            best $\leftarrow$ cand
        if $\min_{k' > k} a + \varepsilon_{k'}(m, w_{k'} \delta) >$ best then
            break
    return $\hat{h}$
```

Note that if we "decompose" as $\mathcal{H}_1 = \mathcal{H}$, then SRM becomes just $\mathrm{ERM}_{\mathcal{H}}$.

THEOREM 3. *Let $h^* \in \mathcal{H}$ be any fixed hypothesis in the setup of Definition 2, and let $a \leq \ell(h, z) \leq b$ for all $h \in \mathcal{H}$, $z \in \mathcal{Z}$. Then, with probability at least $1 - 2\delta$ over the choice of random samples $\mathrm{S} \sim \mathcal{D}^m$, SRM satisfies*

$$\mathrm{L}_{\mathcal{D}}(\mathrm{SRM}_{\mathcal{H},\delta}(\mathrm{S})) \leq \mathrm{L}_{\mathcal{D}}(h^*) + \varepsilon_{k_{h^*}}\left(m, w_{k_{h^*}} \delta\right) + (b - a)\sqrt{\tfrac{1}{2m} \log \tfrac{1}{\delta}}.$$

Proof. Let $\hat{h}_\mathrm{S} = \mathrm{SRM}_{\mathcal{H}}(\mathrm{S})$. We have that

$$\begin{aligned} \mathrm{L}_{\mathcal{D}}(\hat{h}_\mathrm{S}) &\leq \mathrm{L}_\mathrm{S}(\hat{h}_\mathrm{S}) + \varepsilon_{k_{\hat{h}_\mathrm{S}}}(m, w_{k_{\hat{h}_\mathrm{S}}} \delta/2) && \text{by Proposition 1, prob} \geq 1 - \delta \\ &\leq \mathrm{L}_\mathrm{S}(h^*) + \varepsilon_{k_{h^*}}(m, w_{k_{h^*}} \delta/2) && \text{by def of SRM;} \end{aligned}$$

the conclusion follows by applying Hoeffding's inequality with probability δ to upper-bound $\mathrm{L}_\mathrm{S}(h^*)$. □

Note that the number of samples m required to achieve a particular error ε depends on the choice of h^*, unlike in PAC learning!

1.1 *Nonuniform learnability*

DEFINITION 4. An algorithm $\mathcal{A}(S)$ (ε, δ)-*competes with* a hypothesis h if it satisfies $\Pr_{S\sim\mathcal{D}^m}(L_{\mathcal{D}}(\mathcal{A}(S)) \le L_{\mathcal{D}}(h) + \varepsilon) \ge 1 - \delta$.

DEFINITION 5. An algorithm $\mathcal{A}$ *nonuniformly learns* $\mathcal{H}$ there is a finite sample complexity function $m(\varepsilon, \delta, h)$ such that for all $\varepsilon, \delta \in (0, 1)$ and $h \in \mathcal{H}$, given $m \ge m(\varepsilon, \delta, h)$ iid samples from any $\mathcal{D}$, $\mathcal{A}(S)$ (ε, δ)-competes with h.

DEFINITION 6. A hypothesis class $\mathcal{H}$ is *nonuniformly learnable* if there exists an algorithm $\mathcal{A}$ which nonuniformly learns $\mathcal{H}$.

Theorem 3 establishes that SRM nonuniformly learns any $\mathcal{H}$ which we can decompose into a countable union of $\mathcal{H}_k$ which each allow for uniform convergence.

In fact, for binary classifiers with 0-1 loss, SRM nonuniformly learns any $\mathcal{H}$ which is nonuniformly learnable:

PROPOSITION 7. *If $\mathcal{H}$ of binary classifiers is nonuniformly learnable under the 0-1 loss, it can be written as a countable union of $\mathcal{H}_k$ with finite VC dimension.*

Proof. Define

$$\mathcal{H}_k = \left\{h \in \mathcal{H} : m\left(\tfrac{1}{8}, \tfrac{1}{7}, h\right) \le k\right\},$$

where $m(\varepsilon, \delta, h)$ is the sample complexity function of an algorithm A that nonuniformly learns $\mathcal{H}$. Then $\mathcal{H} = \bigcup_{k\ge1} \mathcal{H}_k$.

For any k, consider $\mathcal{H}_k$. Let $\mathcal{D}$ be any distribution realizable by $\mathcal{H}_k$, i.e. there is some $h^* \in \mathcal{H}_k$ with $L_{\mathcal{D}}(h^*) = 0$. Since $\mathcal{A}(S)$ competes with that h^*, $\Pr_{S\sim\mathcal{D}^m}(L_{\mathcal{D}}(A(S)) \le \frac{1}{8}) \ge \frac{6}{7}$. This means that we can (roughly) learn *any* realizable distribution. But the No Free Lunch theorem from last time (specifically Corollary 2) implied that, if $\mathrm{VCdim}(\mathcal{H}_k) = \infty$, then there would be some realizable $\mathcal{D}$ that we can't learn to this (ε, δ). Thus $\mathrm{VCdim}(\mathcal{H}_k)$ can't be infinite. □

1.2 *Problems with bound minimization*

Concentration inequalities are usually pretty conservative, since they hold for *all* distributions subject to some mild constraints (e.g. sub-Gaussianity). Symmetrization is also often a bit loose; it introduces a factor of 2 that might not be needed, e.g. in equation (11) / Appendix E.4 of [Zho+22] we established that this 2 can (basically) be a 1 for Gaussian-data ℓ_1-loss regression.

So, if we minimize a potentially loose bound, then we might get bad results: because our bound is too conservative, we'll have too much bias towards a simple solution. (If the problem turns out to be realizable, but we didn't assume that from the outset, then we can't adapt to the fast $1/m$ rate; we'll operate assuming the slow $1/\sqrt{m}$ rate.) Fundamentally, this means the performance of our algorithm is based on how good at theoretical analysis we are; we'd usually rather have an algorithm that works well whether we're smart or not.

It's also kind of weird for us to have to pre-commit to a certain failure probability δ; that's not usually how we think about things. That in particular, though, we'll be able to avoid.

1.3 *Avoiding dependence on* δ

Let $\mathcal{R}_k = \mathbb{E}_{S\sim\mathcal{D}^m} \operatorname{Rad}((\ell \circ \mathcal{H}_k)|_S)$, and assume $\ell \in [a, b]$. Then Proposition 1 becomes that, with probability at least $1-\delta$ it holds uniformly for all $h \in \mathcal{H}$ that

$$L_{\mathcal{D}}(h) \le L_S(h) + 2\mathcal{R}_{k_h} + (b-a)\sqrt{\frac{1}{2m}\log\frac{1}{w_{k_h}\delta}}.$$

Let's choose the concrete set of weights $w_k = 6/(\pi^2 k^2)$. We can make things look a little nicer by noticing that

$$\log\frac{1}{w_{k_h}\delta} = \log\frac{\pi^2 k_h^2}{6} + \log\frac{1}{\delta} = \underbrace{\log\frac{\pi^2}{6}}_{0.4977\cdots} + 2\log k_h + \log\frac{1}{\delta}$$

$$\sqrt{\log\frac{1}{w_{k_h}\delta}} < \sqrt{\frac{1}{2} + 2\log k_h + \log\frac{1}{\delta}} \le \sqrt{2\log k_h} + \sqrt{\log\frac{\sqrt{e}}{\delta}}.$$

Thus, with probability at least $1-\delta$,

$$\forall h \in \mathcal{H}.\quad L_{\mathcal{D}}(h) \le L_S(h) + 2\mathcal{R}_{k_h} + (b-a)\sqrt{\frac{1}{m}\log k_h} + (b-a)\sqrt{\frac{1}{2m}\log\frac{\sqrt{e}}{\delta}}.$$

Using this slightly looser upper bound in SRM, we can get a version where our optimization doesn't depend on δ (a nice thing to have in practice):

$$\hat{h}_S \in \operatorname*{arg\,min}_{h\in\mathcal{H}} L_S(h) + 2\mathop{\mathbb{E}}_{S'} \operatorname{Rad}\left(\mathcal{H}_{k_h}|_{S'_x}\right) + (b-a)\sqrt{\frac{1}{m}\log k_h}. \tag{2}$$

Instead of 3, *we could use* $1+\sqrt{e}\approx 2.65$ *if we cared.*

We can now do an version of Theorem 3; for $\hat{h}_S$ being this version of SRM, we have that

$$\begin{aligned}
L_{\mathcal{D}}(\hat{h}_S) &\le L_S(\hat{h}_S) + 2\mathcal{R}_{k_{\hat{h}_S}} + (b-a)\sqrt{\frac{1}{m}\log k_{\hat{h}_S}} + (b-a)\sqrt{\frac{1}{2m}\log\frac{3}{\delta}} && \text{prob} \ge 1 - \frac{\sqrt{e}}{3}\delta \\
&\le L_S(h^*) + 2\mathcal{R}_{k_{h^*}} + (b-a)\sqrt{\frac{1}{m}\log k_{h^*}} + (b-a)\sqrt{\frac{1}{2m}\log\frac{3}{\delta}} && \text{def of SRM} \\
&\le L_{\mathcal{D}}(h^*) + 2\mathcal{R}_{k_{h^*}} + (b-a)\sqrt{\frac{1}{m}\log k_{h^*}} + (b-a)\sqrt{\frac{2}{m}\log\frac{3}{\delta}} && \text{prob} \ge 1 - \frac{1}{3}\delta,
\end{aligned}$$

where the last step applied Hoeffding to $L_S(h^*)$. The total failure probability is thus at most $\frac{\sqrt{e}+1}{3}\delta < 0.883\,\delta < \delta$.

By comparison, if we'd known the "correct" $\mathcal{H}_{k_{h^*}}$ from the start and run ERM on that, we'd get

$$L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}_{k_{h^*}}}(S)) \le L_{\mathcal{D}}(h^*) + 2\mathcal{R}_{k_{h^*}} + (b-a)\sqrt{\frac{2}{m}\log\frac{2}{\delta}};$$

the only differences are changing $\log\frac{2}{\delta}$ to $\log\frac{3}{\delta}$ and adding the $\sqrt{\frac{1}{m}\log k_h}$ term, which is typically not a big deal if we've picked our $\mathcal{H}_k$ appropriately.

1.4 *Varyingly-bounded losses*

If we use losses where $[a, b]$ varies for $\mathcal{H}_k$ – for instance, logistic regression with $\|x\| \leq C$ and $\mathcal{H}_k = \{x \mapsto w \cdot x : \|w\| \leq B_k\}$ – then this variant of SRM becomes a little awkward: the bound above would use $(b-a)_{k_{h^*}}$ for the $\log k_{h^*}$ term, but the confidence term would be $\left((b-a)_{k_{\hat{h}_S}} + (b-a)_{k_{h^*}}\right)\sqrt{\frac{1}{2m}\log\frac{3}{\delta}}$. If we instead use the version from Definition 2 and commit to a δ during learning, the bound of Theorem 3 would apply and could look like the above with only $(b-a)_{k_{h^*}}$.

If we choose $B_k = 2^k$ for logistic regression, we'd have $k_h < \log_2(2\|w\|)$, so that $\sqrt{\log k_{h^*}} < \sqrt{\log\log_2(2\|w^*\|)}$, while $(b-a)_{k_h} < (2C\|w\| + 1)$. Thus the variant that doesn't commit to a δ beforehand looks like

$$\hat{h}_S = h_{\hat{w}_S};\ \hat{w}_S \in \operatorname*{arg\,min}_{w \in \mathbb{R}^d} L_S(h_w) + \frac{4C\|w\|}{\sqrt{m}} + (2C\|w\| + 1)\sqrt{\frac{1}{m}\log\log_2(2\|w\|)} \tag{3}$$

$$L_{\mathcal{D}}(\hat{h}_S) \leq L_{\mathcal{D}}(h_{w^*}) + \frac{4C\|w^*\|}{\sqrt{m}} + (2C\|w^*\| + 1)\sqrt{\frac{1}{m}\log\log_2(2\|w^*\|)}$$
$$+ (2C\|w^*\| + 2C\|\hat{w}_S\| + 2)\sqrt{\frac{1}{2m}\log\frac{3}{\delta}}.$$

1.5 *Relationship to regularization*

It's worth highlighting that, if we squint a bit at (3) and assume that w with $\|w\|$ so big that $\log\log_2\|w\|$ is meaningfully more than "constant" aren't relevant to the optimization, it looks a lot like

$$\hat{w}_S \in \operatorname*{arg\,min}_{w \in \mathbb{R}^d} L_S(h_w) + \frac{\lambda}{\sqrt{m}}\|w\|$$

for some $\lambda > 0$. This is pretty close to the "default" regularized logistic regression, which would use $\|w\|^2$. (It also probably wouldn't have an explicit m in the equation, but if you're tuning λ for a fixed particular problem, that doesn't matter, and indeed the total amount of regularization should often scale with m according to $\sqrt{m}$, as we'll see a little later in the course.)

In fact, the problem with $\|w\|$ and with $\|w\|^2$ are themselves equivalent: if you consider the curve of possible solutions as you vary λ (the "regularization path"), you would get the exact same set of solutions. So, SRM can be seen as motivation for standard regularization techniques.

2 MINIMUM DESCRIPTION LENGTH

2.1 *Singleton Classes*

Suppose we have a countable $\mathcal{H} = \{h_1, h_2, \ldots\}$. Then we could partition it into *singleton* sub-classes, $\mathcal{H}_k = \{h_k\}$. Denoting the weight for the class $\{h\}$ by w_h, each of these $\mathcal{H}_k$ have "uniform convergence" via a simple Hoeffding bound with

$$\varepsilon_k(m, w_h\delta) \leq (b-a)\sqrt{\frac{1}{2m}\log\frac{1}{w_h\delta}} \leq (b-a)\sqrt{\frac{1}{2m}\log\frac{1}{w_h}} + (b-a)\sqrt{\frac{1}{2m}\log\frac{1}{\delta}},$$

splitting out the dependence on δ for simplicity as before. SRM then becomes

$$\mathrm{SRM}_{\mathcal{H}}(\mathrm{S}) \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \mathrm{L}_{\mathrm{S}}(h) + \sqrt{\frac{1}{2m} \log \frac{1}{w_h}},$$

and this has the guarantee by Theorem 3 that

$$\mathrm{L}_{\mathcal{D}}(\mathrm{SRM}_{\mathcal{H}}(\mathrm{S})) \le \mathrm{L}_{\mathcal{D}}(h^*) + (b-a)\sqrt{\frac{1}{2m} \log \frac{1}{w_{h^*}}} + (b-a)\sqrt{\frac{2}{m} \log \frac{2}{\delta}}.$$

But…how should we set w_h? What order should we use?

2.2 *Minimum Description Length*

One popular way to decide on weights is based on choosing some *prefix-free binary language* to determine the hypotheses: for example, the binary representation of a `gzip`ed Python program implementing that hypothesis. Then we can choose a weight according to the following result:

proposition 8 (Kraft's inequality). *If $\mathcal{S} \subseteq \{0,1\}^*$ is prefix-free (there are no $s \ne s' \in \mathcal{S}$ such that s is a prefix of s'), then*

$$\sum_{s \in \mathcal{S}} 2^{-|s|} \le 1.$$

Proof. Define the following random process: starting with the empty string, add either a 0 or a 1 with equal probability. If the current string is in $\mathcal{S}$, terminate; if no element of $\mathcal{S}$ begins with the current string, also terminate; otherwise, repeat. Since $\mathcal{S}$ is prefix-free, this process hits any string $s \in \mathcal{S}$ with probability $2^{-|s|}$; these probabilities must sum to at most one. □

Thus, we can choose a representation for $\mathcal{H}$ and assign $w_h = 2^{-|h|}$. This gives

$$\mathrm{MDL}_{\mathcal{H}}(\mathrm{S}) \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \mathrm{L}_{\mathrm{S}}(h) + \sqrt{\frac{\log 2}{2m} |h|}$$

$$\mathrm{L}_{\mathcal{D}}(\mathrm{MDL}_{\mathcal{H}}(\mathrm{S})) \le \mathrm{L}_{\mathcal{D}}(h^*) + (b-a)\sqrt{\frac{\log 2}{2m} |h^*|} + (b-a)\sqrt{\frac{2}{m} \log \frac{2}{\delta}}.$$

This is one formalization of Occam's razor: if there are multiple explanations of the data ($\mathrm{L}_{\mathrm{S}}(h_1) = 0 = \mathrm{L}_{\mathrm{S}}(h_2)$), prefer the simplest one (the one with shortest explanation).

But we need to *pre-commit* to a notion of description length before seeing the data. A nice analogy: `codegolf.stackexchange.com`, a site where people compete to find the shortest implementation of a program doing some task, prohibits by default any language written after the contest was started.

It's not quite *the same; MAP wouldn't have the square root.*

If we choose $|h|$ to be the length of shortest possible implementation of h in some programming language, this is known as the *Kolmogorov complexity*. This version of the MDL principle is then to regularize by the Kolmogorov complexity. If you're familiar with Bayesian learning, this would be something like *maximum a posteriori* (MAP) inference with a Kolmogorov complexity prior. The "free lunch" algorithm outlined by Nakkiran [Nak21] is closely related to this where $\mathcal{H}$ is just the set of all Turing machines. The fully-Bayesian analogue is (basically) something called

Solomonoff induction. For fuller introductions to these concepts, see the textbooks of Li and Vitányi [LV19] and Hutter [Hut05].

REFERENCES

[Hut05] Marcus Hutter. *Universal Artificial Intelligence. Sequential Decisions based on Algorithmic Probability*. 2005.

[LV19] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. 4th edition. 2019.

[Nak21] Preetum Nakkiran. *Turing-Universal Learners with Optimal Scaling Laws*. 2021. arXiv: 2111.05321.

[Zho+22] Lijia Zhou, Frederic Koehler, Pragya Sur, Danica J. Sutherland, and Nathan Srebro. "A Non-Asymptotic Moreau Envelope Theory for High-Dimensional Generalized Linear Models." *NeurIPS*. 2022. arXiv: 2210.12082.