CPSC 532S: Assignment 3 – due Friday, 25 Mar 2022, 11:59pm

Default late policy: -5 points on the assignment if you're 0-24 hours late, -10 if you're 24-48 hours late, not accepted after that. If you have something significant going on / are sick / whatever, write to me, and I can be more flexible there.

This assignment is split into four questions (the parts with big section headers; most have sub-parts). You can solve each question in groups of up to three. Groups don't need to be consistent between problems; you can do Q1 and Q2 alone, Q3 with Alice, and Q4 with Bob and Carlos if you want.

Please **do not** just split the questions up and do the parts separately. **If your name is on a solution, you are pledging that you contributed significantly to the solution and understand it fully.**

There is a separate Gradescope assignment for each problem; use the Gradescope groups feature to submit once and associate with each of you, but also put all of your names on the first page as a backup.

Prepare your answers to these questions **using LaTeX**. Hopefully you're reasonably familiar with it, but if not, try using Overleaf and looking around for tutorials online. (Note that free Overleaf accounts can only share with one "named collaborator," but you can collaborate with more people by sending them an edit link. Make sure you only share the parts of the homework you're handing in together!)

Feel free to ask questions if you get stuck on things on Piazza (but remove any details about the actual answers... feel free to make a private post if that's tough). If you look stuff up anywhere other than in one of the two course textbooks, please **cite your sources**: just say in the answer to that question where you looked. (A link is fine, no need for a formal citation.) Please do not look at solution manuals or so on. If you accidentally come across a solution while looking for something related, still write the argument up in your own words, link to wherever you found it, and be clear about what happened.

If you like, the `.tex` source for this file is available on the course website, and you can put your answers in `\begin{answer} My answer here... \end{answer}` environments to make them stand out if so; feel free to delete whatever boilerplate you want (or not, I'm not printing them out). Or answer in a fresh document; just make it clear which question you're answering where.

If you're using a consistent group and want to write your answers in one document, you could split the PDF with e.g. `qpdf a2.pdf --pages . 2-3 -- q1.pdf` or through the GUI of a PDF viewer. Or you can upload the full file four times and just make sure you assign pages appropriately.

Submit your answers as a PDF on Gradescope: instructions on Piazza. You'll be prompted to mark where each sub-part is in your PDF; make sure you mark all relevant pages for each part. (This saves me a surprising amount of time in grading.) If something goes wrong, you can also email your assignment to me directly (`dsuth@cs.ubc.ca`).

1 ERM vs SRM vs RLM [25 points]

Let's say we're given a training set $S = ((x_1, y_1), \ldots, (x_n, y_n)) \sim \mathcal{D}^n$, with $x_i \in \mathbb{R}^d$ and $y_i \in [-1, 1]$. We're going to try to learning a homogeneous linear function h from $\mathcal{H} = \{x \mapsto w^\mathsf{T} x : w \in \mathbb{R}^d\}$, using the loss function $\ell^{\text{abs}}(h, (x, y)) = |h(x) - y|$. Let's use the notation w_h to refer to the w inside h, i.e. $h(x) = w_h^\top x$. You can assume $n > d \geq 3$, and that $\Pr(\|x\| \leq R) = 1$ for some finite R.

(a) [5 points] Suppose we just run empirical risk minimization, $h_S^{\text{ERM}} = \arg\min_{h \in \mathcal{H}} L_S^{\text{abs}}(h)$. What can we say about the generalization gap $L_{\mathcal{D}}^{\text{abs}}(h_S^{\text{ERM}}) - L_S^{\text{abs}}(h_S^{\text{ERM}})$ using a type of bound covered in class?

Instead, let's break $\mathcal{H}$ into parts: $\mathcal{H}_1 = \{x \mapsto w^\mathsf{T} x : \|w\| \leq a\}$, $\mathcal{H}_2 = \{x \mapsto w^\mathsf{T} x : \|w\| \leq 2a\}$, etc, for a constant $a > 0$ to be determined later. Now, run SRM, using "weight" $6/(\pi^2 k^2)$ for $\mathcal{H}_k$ as in class.

(b) [5 points] Write the solution h_S^{SRM} in the form $\arg\min_h f(L_S^{\text{abs}}(h), \|w_h\|, a, n, R, \delta)$, for some function f, where w_h is the vector w "inside" h. Don't include terms explicitly depending on $\mathfrak{R}_n(\mathcal{H}_k)$ or similar; it should be an expression that would be straightforward to implement in code. You can use the ceil function $\lceil \cdot \rceil$ (rounding up) or floor $\lfloor \cdot \rfloor$ (rounding down), if you'd like.

Hint: When we discussed SRM before, we focused on 0-1 loss. But there's nothing actually in the argument that requires it, as long as we have a uniform convergence bound on $\mathcal{H}_k$. You may want to just re-derive it from the framework of minimizing an upper bound on $L_{\mathcal{D}}(h)$.

(c) [5 points] What can we say about the generalization gap $L_{\mathcal{D}}^{\text{abs}}(h_S^{\text{SRM}}) - L_S^{\text{abs}}(h_S^{\text{SRM}})$ using a type of bound covered in class? Again, this should be a closed-form expression in terms of $\|w_h\|$, a, R, n, and δ.

Now, the last of our candidate algorithms is regularized loss minimization:

$$h_S^{\text{RLM}} = \arg\min_{h \in \mathcal{H}} L_S^{\text{abs}}(h) + \lambda \|w_h\|^2.$$

(d) [5 points] What can we say about the generalization gap $L_{\mathcal{D}}^{\text{abs}}(h_S^{\text{RLM}}) - L_S^{\text{abs}}(h_S^{\text{RLM}})$ using a type of bound covered in class?

(e) [5 points] Can we pick λ and a (maybe depending on n) so that RLM looks roughly like SRM? (It's okay to be a little approximate here, e.g. ignoring logarithmic terms; we're just talking about motivations.)

2 Logistic regression [25 points]

Let $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\| \leq R\}$, and $\mathcal{H} = \{w \in \mathbb{R}^d : \|w\| \leq B\}$. We'll learn a linear predictor based on logistic loss,

$$\ell(w, (x, y)) = \log(1 + \exp(-yw^\mathsf{T} x)).$$

Let S be a sample $((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$, and let $X \in \mathbb{R}^{n \times d}$, $y \in \mathbb{R}^n$ stack up the features and labels accordingly.

(a) [9 points] Show that $L_S(w)$ is a convex function of w.

(b) [8 points] Show $L_S(w)$ is ρ-Lipschitz, and give a (reasonably tight) upper bound on ρ.

(c) [8 points] Show $L_S(w)$ is β-smooth, and give a (reasonably tight) upper bound on β.

As $\|w\|$ is bounded, this means that logistic regression is both Convex-Lipschitz-Bounded and Convex-Smooth-Bounded.

3 From Bounded Expected Risk to Agnostic PAC Learning [25 points]

Our stability and SGD analyses mostly bounded only the expected risk; we'll now show this implies PAC learning.

Let A be a proper learning algorithm (one returning hypotheses in $\mathcal{H}$) that guarantees: if $n \geq n_{\mathcal{H}}(\varepsilon)$, then for every distribution $\mathcal{D}$, it holds that

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^n} L_{\mathcal{D}}(A(S)) \leq \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \varepsilon.$$

You can assume that $L_{\mathcal{D}}(h) = \mathbb{E}_{z\sim\mathcal{D}}\,\ell(h,z)$ for a loss $\ell(h,z)$ bounded in $[0,1]$.

(a) [10 points] Show that for every $\delta \in (0,1)$, if $n \geq n_{\mathcal{H}}(\varepsilon\delta)$, then with probability of at least $1-\delta$ it holds that $L_{\mathcal{D}}(A(S)) \leq \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \varepsilon$.

Hint: Observe that the random variable $L_{\mathcal{D}}(A(S)) - \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$ is nonnegative, and rely on Markov's inequality.

(b) [15 points] For every $\delta \in (0,1)$, let

$$n_{\mathcal{H}}(\varepsilon,\delta) = n_{\mathcal{H}}(\tfrac{\varepsilon}{4}) \left\lceil \log_2 \tfrac{2}{\delta} \right\rceil + 8 \left\lceil \frac{\log \frac{4}{\delta} + \log\left\lceil \log_2 \frac{2}{\delta} \right\rceil}{\varepsilon^2} \right\rceil.$$

Suggest a procedure that agnostic PAC learns the problem with sample complexity of $n_{\mathcal{H}}(\varepsilon,\delta)$, assuming that the loss function is bounded by 1.

Hint: Let $k = \lceil \log_2(\frac{2}{\delta}) \rceil$. Divide the data into $k+1$ chunks, where each of the first k chunks has at least $n_{\mathcal{H}}(\varepsilon/4)$ examples. Train the first k chunks using A. Using part (a), argue that the probability that for all of these chunks we have $L_{\mathcal{D}}(A(S)) > \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \varepsilon/2$ is at most $2^{-k} \leq \delta/2$. Finally, use the last chunk as a validation set.

4 Perceptrons [25 points]

Let $S = ((x_1, y_1), \ldots, (x_n, y_n)) \in \left(\mathbb{R}^d \times \{\pm 1\}\right)^n$. Assume that a

$$w^* \in \operatorname*{arg\,min}_{w \in \mathbb{R}^d:\ \forall i \in [n],\ y_i w^\mathsf{T} x_i \geq 1} \|w\|$$

exists. Let $R = \max_i \|x_i\|$, and let

$$f(w) = \max_{i \in [n]} 1 - y_i w^\mathsf{T} x_i.$$

(a) [5 points] Show that $\min_{w:\|w\| \leq \|w^*\|} f(w) = 0$, and that any w for which $f(w) < 1$ achieves $L_S^{0-1}(w) = 0$.

(b) [5 points] Show how to calculate a subgradient of f.

Hint: Recall that, at points for which f is differentiable, the gradient is a valid subgradient. For points where it's not, think about the structure of f: try drawing a sketch with $d = 1$ and $n = 2$.

(c) [5 points] Describe subgradient descent on the function f (initializing from $w = 0$). UPDATE: no need to analyze the algorithm.

(d) [5 points] Compare the resulting algorithm (not its analysis anymore) to the Batch Perceptron algorithm given in section 9.1.2 of SSBD. (We didn't discuss this in class, but just reading the algorithm block should be enough.)

(e) [5 points] Suppose that the training set S is such that the training instances are linearly separable with a margin of γ. Refine the bound of SSBD's Theorem 9.1 to include γ.