Computational and Statistical Learning Theory

Problem sets 3 and 4

Due: June 5th

Please send your solutions to `learning-submissions@ttic.edu`

Notation:

- Input space: $\mathcal{X}$
- Label space: $\mathcal{Y} = \{\pm 1\}$
- Sample: $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X}$
- Hypothesis Class: $\mathcal{H}$
- Risk: $L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbf{1}_{h(x)\neq y}\right]$
- Empirical Risk: $L_S(h) = \frac{1}{m}\sum_{(x,y)\in S}\mathbf{1}_{h(x)\neq y}$

1. A k-layer L_1-norm neural network is given by function class $\mathcal{F}_k$ which is in turn defined recursively as follows.

$$\mathcal{F}_1 = \left\{ x \mapsto \sum_{j=1}^{d} w_j^1 x_j \;\middle|\; \left\|w^1\right\|_1 \leq B_1 \right\}$$

and further for each $2 \leq i \leq k$,

$$\mathcal{F}_i = \left\{ x \mapsto \sum_{j=1}^{d_i} w_j^1 \sigma(f_j(x)) \;\middle|\; \forall j \in [d_i],\ f_j \in \mathcal{F}_{i-1},\ \left\|w^i\right\|_1 \leq B_i \right\}$$

where d_i is the number of nodes in the ith layer of the network. Function $\sigma : \mathbb{R} \mapsto [-1, 1]$ is called the squash function and is generally a smooth monotonic non-decreasing function (typical example is the $\tanh$ function). Assume that input space $\mathcal{X} = [0, 1]^d$ and that σ is L-Lipschitz. Prove that

$$\widehat{\mathcal{R}}_S(\mathcal{F}_k) \leq \left(\prod_{i=1}^{k} 2B_i\right) L^{k-1}\sqrt{2T\log d}$$

where $T = 1/|\mathcal{S}| = 1/m$. Notice that the above bound the d_i's don't appear in the bound indicating the number of nodes in intermediate layers don't affect the upper bound

on Rademacher complexity.

Hint : prove bound on Rademacher complexity of $\mathcal{F}_i$ recursively in terms of Rademacher complexity of $\mathcal{F}_{i-1}$.

2. Prove replace-one stability of RERM with $\lambda \|w\|_2^2$ regularization.

3. Using $\Psi(w) = \|w\|_p^2$ as a regularizer,

   (a) Prove $\Psi(w)$ is α-strongly convex w.r.t. $\|w\|_p$.

   (b) Calculate

      i. Link function $\nabla\Psi(w)$.

      ii. Link function $\nabla\Psi^{-1}(\nu)$

      iii. Bregman divergence $D_\Psi(w\|w')$

   (c) Consider supervised learning, with linear predictor, hinge loss, $\|\phi(x)\|_\infty \leq 1$, $\phi(x) \in \mathbb{R}^d$, and learning the hypothesis class $\mathcal{H} = \left\{w \;\middle|\; \|w\|_1 \leq B\right\}$. We would like to learn this hypothesis class using L-FTRL with the regularizer $\Psi(w) = \|w\|_p^2$.

      i. Show that for an appropriate value of p, we get online regret $O(\sqrt{B^2 \log(d)/m})$. State the value of p exactly, specify the learning rule, and in particular what sequence of regularization parameters you would use, and give the resulting regret bound exactly (not using big-O notation).

      ii. Derive explicit pseudo-code for the update

      iii. Instead consider using non-linearized online mirror descent. Derive explicit pseudo-code for the update

4. Perceptron:

   (a) For any $\gamma > 0$, let $d \geq \frac{1}{\gamma^2}$ and $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\| \leq 1\}$ and $\mathcal{Y} = \{\pm 1\}$. Show that for any online learning algorithm, there exists a sequence of instances $(x_1, y_1), \ldots, (x_m, y_m)$ which is separable by a margin of γ by some linear separator with ℓ_2 norm bounded by 1, such that the online algorithm makes at least $\lfloor \frac{1}{\gamma^2} \rfloor$ mistakes on this sample. This shows that the perceptron bound is tight.

   Hint : Pick appropriate m (depending on γ) and provide instances adversarially so that the algorithm makes a mistake on every round. However show that the selected instances are separable by a linear separator of norm 1 with a margin of at least γ.

   (b) Lower Bound for Perceptron:

      i. Recall the perceptron rule : if $y \langle w, x \rangle \leq 0$, then add yx to w.
      Instead of assuming the existence of w s.t. for all t, $y_t \langle w, x_t \rangle > 1$ (setting $\gamma = 1$), we will derive a mistake bound that bounds the number of mistakes the (standard) perceptron makes in terms of best possible total hinge loss on the sequence.

For any sequence $(x_t, y_t)_{t=1\ldots m}$, where $\|x_t\| \leq 1$ and $y_t \in \{\pm 1\}$, let $|M_m|$ be the number of mistakes made by the perceptron:

$$|M_m| = \{t = 1 \ldots m | y_t \langle w_t, x_t \rangle \leq 0\}$$

For any w^*, let H^*_m be the total hinge loss of w^* on the sequence:

$$H^*_m = \sum_{t=1}^{m} [1 - y_t \langle w^*, x_t \rangle]_+$$

Prove the following:

$$M_m \leq H^*_m + \|w^*\|^2 + \|w^*\| \sqrt{H^*_m}$$

Hint: follow the perceptron analysis as in class: Bound $\|w_{t+1}\|^2$ from above in terms of $|M_t|$. Then bound $\langle w^*, w_{t+1} \rangle$ from below in terms of both $|M_t|$ and H^*_t. Combine the two bounds and solve a quadratic equation to obtain the bound on $|M_m|$.

ii. Use an online-to-batch conversion to obtain a learning rule A for which, with high probability, for every w^* with $\|w^*\|_2 \leq B$:

$$L_{01}(A(S)) \leq L^* + O(B^2/m + \sqrt{B^2 L^*/m})$$

where $L^* = L_{\text{hinge}}(w^*)$. State the learning rule explicitly and prove the learning guarantee. Is this a proper learning rule? What form of predictors does it output?