TTIC 31020: Intro to ML **Winter 2024**

Homework 3

Due: 6 p.m., January 23rd, 2024

Note *You may discuss these problems in groups. However, you must write up your own solutions and mention the names of the people in your group. Also, please do mention any books, papers or other sources you refer to. We recommend that you typeset your solutions in LaTeX. Please submit your solutions as a PDF document on Canvas.*

Challenge and Optional Questions *Challenge questions are marked with* challenge *and Optional questions are marked with* optional *. You can get extra credit for solving* challenge *questions. You are not required to turn in* optional *questions, but we encourage you to solve them for your understanding. Course staff will help with* optional *questions but will prioritize queries on non-*optional *questions.*

Question 3 will reappear in HW4. You are not required to turn it in as a part of your HW3.

1. **Realizable Online-to-Batch and Leave-One-Out Cross-Validation**

   In this question we will make a formal connection between the online mistake bound and the sample complexity in the statistical setting. We will also study the Leave-One-Out Cross-Validation (LOOCV) estimate of the error of a learning rule.

   Consider a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and an online learning rule $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$ (recall that a learning rule can be formalized as a mapping from the sequence of examples seen so far to a predictor), and suppose $\mathcal{A}$ enjoys a mistake bound M for sequences realized by $\mathcal{H}$ (that is, $\mathcal{A}$ makes at most M mistakes on any sequence $(x_i, y_i)_i$ s.t. there exists some $h \in \mathcal{H}$ for which $y_i = h(x_i)$ for all i.). We want to use this online rule to get a good learning rule in the *statistical* setting, where we are given a sample $S \sim \mathcal{D}^m$ of examples from some unknown source distribution $\mathcal{D}(\mathcal{X}, \mathcal{Y})$. We will use the following construction of a statistical (batch) rule $\tilde{\mathcal{A}}$ that uses the online rule $\mathcal{A}$:

Input: Training set $S = \left\{(x_i, y_i)_{i=1}^m\right\}$
Output: Predictor $h : \mathcal{X} \to \mathcal{Y}$ with zero training error, i.e. $h \in \mathcal{H}$ s.t. $\forall\, i, h(x_i) = y_i$.

Initialize $S' = \varnothing$ (empty sequence of examples) and $h = \mathcal{A}(\varnothing)$ the initial predictor used by the online rule before seeing any examples.
while *there exists* $(x_i, y_i) \in S$ *s.t.* $h(x_i) \neq y_i$ **do**
  Append (x_i, y_i) to S' and update $h = \mathcal{A}(S')$ (i.e. feed (x_i, y_i) to the online rule $\mathcal{A}$).
end
return h

   That is, $\tilde{\mathcal{A}}$ runs the online learning rule $\mathcal{A}$ on a sequence of (possibly repeated) examples chosen from S, keeping track of the examples the current predictor $\mathcal{A}$ uses in S'. As long as there is an example in S misclassified by the current predictor, this misclassified example is fed as the next example to $\mathcal{A}$. Note that since an example might be misclassified in multiple rounds, it might appear in the sequence S' multiple times, i.e. be fed to $\mathcal{A}$ multiple times. If in some round there are multiple misclassified example, we need to choose one of them—this might be, e.g. the example with the lowest index (although this arbitrary choice doesn't matter).

   To analyze $\tilde{\mathcal{A}}$ in the statistical setting, we will use the LOOCV error estimate for a learning rule. Given a training set $S = \left\{(x_i, y_i)_{i=1}^m\right\}$, denote $S_{-i} = \{(x_1, y_1), \ldots, (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), \ldots, (x_m, y_m)\}$ the training set with example (x_i, y_i) removed. The LOOCV error rate for a learning rule $\mathcal{F}$ on training set S is obtained by, for each i, running $\mathcal{F}$ on S_{-i} and testing on (x_i, y_i):

$$\mathrm{LOOCV}_S(\mathcal{F}) = \frac{1}{m} \left|\{i| \text{ s.t. } \mathcal{F}(S_{-i})(x_i) \neq y_i\}\right|. \tag{1}$$

(a) Prove that the LOOCV error on a set of m i.i.d. samples from $\mathcal{D}$ is an unbiased estimator of the expected error rate of $\mathcal{F}$ on $m-1$ i.i.d. samples from $\mathcal{D}$, that is:

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[\text{LOOCV}_S(\mathcal{F})\right] = \mathbb{E}_{S\sim\mathcal{D}^{m-1}}\left[L_{\mathcal{D}}(\mathcal{F}(S))\right]. \tag{2}$$

(b) Consider running $\tilde{\mathcal{A}}$ on m i.i.d. samples $S\sim\mathcal{D}^m$ from a distribution realizable by $\mathcal{H}$[1]. If $\tilde{\mathcal{A}}$ runs for T iterations, i.e. feeds a total of T examples to $\mathcal{A}$, and so $\mathcal{A}$ is run for T steps, how many mistakes does $\mathcal{A}$ make on examples it is fed? Determine a bound on the number of iteration $\tilde{\mathcal{A}}$ might run.

(c) For a set S of $m+1$ examples, bound $\text{LOOCV}_S(\tilde{\mathcal{A}})$ in terms of the number of iterations $\tilde{\mathcal{A}}$ would run on S.

(d) By combining (a) and (c), determine a bound on the expected error rate of the predictor output by $\tilde{\mathcal{A}}$, $\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(\tilde{\mathcal{A}}(S))\right]$, when run on m i.i.d. samples from a distribution realized by $\mathcal{H}$, in terms of the mistake bound M and sample size m. How many samples, as a function of M and ϵ are sufficient to ensure $\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(\tilde{\mathcal{A}}(S))\right] < \epsilon$?

(e) optional Why is it not sufficient to just use the rule $\mathcal{A}$ itself, i.e. run $\mathcal{A}$ on the sequence of examples S and output the final predictor? Give a counterexample where $\mathcal{A}$ has a small mistake bound, but for some large m (using the output of the predictor after m steps on i.i.d. examples from a realizable $\mathcal{D}$) results in a very bad predictor.

2. **Online Perceptron and Perceptron Analysis**

In class we presented the Perceptron as an algorithm implementing the CONSISTENT_ϕ learning rule for the class of linear classifiers $\left\{h_w : x\mapsto\langle w,\phi(x)\rangle \middle| w\in\mathbb{R}^d\right\}$ over some feature map $\phi(x)\in\mathbb{R}^d$. That is, an algorithm that takes as input a training set S and returns a linear predictor $h^{\text{sign}}_{\hat{w}} : x\mapsto \text{sign}\left(\langle\hat{w},\phi(x)\rangle\right)$ such that $L^{01}_S\left(h^{\text{sign}}_{\hat{w}}\right)=0$. That is, such that $\forall_{(x_i,y_i)\in S} y_i\langle\hat{w},\phi(x_i)\rangle > 0$.

PERCEPTRON can also be thought of an online learning rule: start with $w_1=0$. At each round t, given an input x_t, we predict a label $\hat{y}_t = h^{\text{sign}}_{w_t}(x_t) = \text{sign}\left(\langle w_t,\phi(x_t)\rangle\right)$. Upon receiving the correct y_t, if we made a mistake and $\hat{y}_t\neq y_t$, we update $w_{t+1}=w_t+y_t\phi(x_t)$. Otherwise (if we didn't make a mistake), we continue with $w_{t+1}=w_t$.

In this problem we will analyze the Perceptron as an online learning rule, an implementation CONSISTENT_ϕ, and as a statistical learning rule.

To simplify our formulas, we will assume throughout this problem that $\|\phi(x)\|=1$, i.e. that the feature vectors all have unit Euclidean norm. Non-normalized feature vectors will introduce an additional dependence on $\|\phi(x)\|$, which we will not get into in this homework.

Part I: Online Analysis *This part is a walk-through tutorial on the Perceptron mistake bound analysis. You will need the results for the following parts, and we strongly encourage you to do it, but you do not have to write it up and submit and it will not be graded.*

For any sequence $(x_t,y_t)_{t=1,2,\ldots}$ (of any length), assume it is realizable by linear predictor w°. We will also need to assume this holds with a *margin* of γ:

$$\forall_t \frac{y_t\langle w^\circ,\phi(x_t)\rangle}{\|w^\circ\|}\geq\gamma. \tag{3}$$

We will prove that for any sequence such that there exists w° satisfying (3), the PERCEPTRON online rule will make at most $1/\gamma^2$ mistakes. We will denote by M_t the number of mistakes made by PERCEPTRON in the first t rounds, i.e. on $x_1,\ldots,x_t$.

[1] Recall that a distribution $\mathcal{D}$ is realizable by $\mathcal{H}$ if $\exists\, h\in\mathcal{H}$ s.t. $L_{\mathcal{D}}(h)=0$.

(a) Do not submit Prove that for all $t = 0, 1, \ldots$, $\frac{\langle w^\circ, w_{t+1} \rangle}{\|w^\circ\|} \geq M_t$. Hint: Start with $t = 0$ and prove by induction on t, showing that whenever PERCEPTRON makes a mistake and updates, $\langle w^\circ, w_{t+1} \rangle \geq \langle w^\circ, w_t \rangle + \gamma \|w^\circ\|$ (i.e. we make progress in the direction of w°).
Just tracking the inner product $\langle w^\circ, w_t \rangle$ is not enough, since it could also be large just because the magnitude of w_t increases, rather than it becoming more aligned with w°. We will thus balance this with also bounding the norm $\|w_t\|$:

(b) Do not submit Prove that for all $t = 0, 1, \ldots$, $\|w_{t+1}\|^2 \leq M_t \|w^\circ\|^2$. Hint: again use induction, expanding $\|w_t + y_t \phi(x_t)\|$, and noting that we only update when $h_{w_t}(x_t) \neq y_t$, i.e. $y_t \langle w_t, \phi(x_t) \rangle \leq 0$.

(c) Do not submit Combine the above two inequalities, with the Cauchy-Schwarz inequality $\langle w^\circ, w_t \rangle \leq \|w^\circ\| \|w_t\|$, to conclude that $M_t \leq 1/\gamma^2$. That is, for *any* sequence of any length satisfying (3), PERCEPTRON will make at most $1/\gamma^2$ mistakes.

Part II: Implementing CONSISTENT We now return to viewing Perceptron as an implementation of the CONSISTENT$_\phi$ learning rule. Here, given a training set S, we start with $w_1 = 0$, and at each iteration t, find $(x_{i_t}, y_{i_t}) \in S$ such that $y_{i_t} \langle w_t, \phi(x_{i_t}) \rangle \leq 0$ and update $w_{t+1} = w_t + y_{i_t} \phi(x_{i_t})$, until there is no such $(x_{i_t}, y_{i_t}) \in S$ and we can return $\tilde{w} = w_t$ and be assured that $L_S^{\text{sign}}(h_{\tilde{w}}) = 0$. That is, the output is indeed a valid solution for CONSISTENT. We want to ensure that this algorithm terminates after a finite number of steps, and bound its runtime.

First, verify that the above implementation of CONSISTENT matches the Online-to-Batch conversion $\widetilde{\text{PERCEPTRON}}$ of the online rule PERCEPTRON, based on the conversion in Question 1 (be sure you understand this, but we are not asking you to submit anything).

We will bound the runtime of $\widetilde{\text{PERCEPTRON}}$ in terms of the margin on the training set, defined as:

$$\gamma(S) = \sup_w \min_{(x_i, y_i) \in S} \frac{y_i \langle w, \phi(x_i) \rangle}{\|w\|}. \tag{4}$$

(a) Prove that if a finite S is realizable by linear predictors, i.e. there exists $h_{\hat{w}}^{\text{sign}} : x \mapsto \text{sign}(\langle \hat{w}, \phi(x) \rangle)$ s.t. $L_S^{01}(h_{\hat{w}}) = 0$, then the margin is positive
Note 1: it is important here that S is finite. optional Can you give an example of an infinite S that is realizable but has zero margin?
Note 2: You might also want to convince yourself that the margin will always be finite, but this is not important for our analysis.

(b) Use the online analysis in Part I to bound the number of iterations (of the batch version analysed in this part) before stopping and returning $\tilde{w}$.

(c) To complete the runtime analysis: how would you implement the step "$(x, y) \in S$ such that $y \langle w, \phi(x) \rangle \leq 0$", and what would be the required runtime per iteration? Write the runtime per iteration, and the overall runtime (relying on the iteration bound above) in terms of the sample size $m = |S|$, the feature space dimension d, and the margin $\gamma(S)$. You may assume that calculating $\phi(x)$ takes time $O(d)$ and each arithmetic operation takes $O(1)$ time.

(d) In the analysis above we relied on S being realizable (linearly separable), i.e. that there exists a linear separator h_w^{sign} such that $L_S^{01}(h_w^{\text{sign}}) = 0$. Show that if S is not linearly separable, the Perceptron algorithm might never stop. Show an explicit training set S, and the sequence of iterates w_t of running Perceptron on S, that demonstrate the algorithm never terminates (hint: S can have two training points).

Part III: Statistical Guarantee Not only can we use the online mistake bound of Part I to bound the runtime of $\widetilde{\text{PERCEPTRON}}$, we can also use it to get a bound on its generalization error! To do so, we will rely on the margin *of a source distribution* $\mathcal{D}$. We will say that $\mathcal{D}$ is separable with margin γ if there exists some $w^\star \in \mathbb{R}^d$ s.t. $\frac{y \langle w^\star, \phi(x) \rangle}{\|w^\star\|}$ almost surely (i.e. with probability 1) for $(x, y) \sim \mathcal{D}$. That is, the margin condition holds for all (x, y) in the support of $\mathcal{D}$.

(a) Combine the mistake bound in Part I of this question with the online-to-batch analysis of Question 1 to bound the expected generalization error $\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{\text{sign}}(\widetilde{\text{PERCEPTRON}}(S))\right]$ in terms of the sample size m and margin of the source distribution $\mathcal{D}$. How many samples are required to ensure expected generalization error at most ϵ?

(b) Amazingly, the error and sample size do not depend on the dimension d, and apply even if d is unbounded! But the VC-dimension of linear predictors is d. Why is there no contradiction to the Fundamental Theorem of Statistical Learning?

(c) optional To obtain the above guarantee, we need to rely on our particular implementation of CONSISTENT using Perpcetron. I.e., this gurantee is specific to the Perceptron rule, and cannot be obtained by any implementation of CONSISTENT. Show that with other implementations, we cannot bound the error even if the margin is small. For any m, show a source distribution (and a feature map, or just a source distribution over $\mathbb{R}^d \times \{\pm 1\}$ with $\|x\| = 1$ almost surely) separable with margin $\gamma = 0.5$ (you can actually do this with any $\gamma < 1$), and implementation A of CONSISTENT (i.e. returning a linear predictor with zero training error, $L_S^{01}(A(S)) = 0$) such that $\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{\text{sign}}(A(S))\right] \geq 0.5$ (you can actually make this arbitrarily close to 1).
Hint: the dimension d will have to increase with m, but you can do this without decreasing the margin.

3. **Hinge Loss vs 0/1 Loss.** ***(This question will reappear in HW4. You are not required to turn it in as a part of your HW3.)***

In this question, you will show that minimizing the hinge loss can be very different from minimizing the 0/1 error. Consider points in $\mathcal{X} = \mathbb{R}^2$ with binary labels and the class of linear predictors $\mathcal{H} = \{h_w(x) = \langle w, x\rangle \mid w \in \mathbb{R}^2\}$.

(a) Show an explicit set of points $S = \{(x_i, y_i) : i \in [m]\}$ such that the following hold simultaneously:

i. $\inf_{h\in\mathcal{H}} L_S^{01}(h) \leq 0.1$

ii. For a predictor minimizing the hinge loss, $\hat{h}_{hinge} = \arg\min_{h\in\mathcal{H}} L_S^{hinge}(h)$, we have $L_S^{01}(\hat{h}_{hinge}) \geq 0.5$.

Write down:

i. The set of points S (you may pick any number of points such that this occurs).

ii. A predictor h minimizing that $L_S^{01}(h)$. Evaluate $\ell^{01}(h)$ and $\ell^{hinge}(h)$.

iii. The predictor $\hat{h}_{hinge}$. Evaluate $\ell^{01}(\hat{h}_{hinge})$ and $\ell^{hinge}(\hat{h}_{hinge})$.

(b) Is it possible that $\inf_{h\in\mathcal{H}} L_S^{01}(h) = 0$ but $L_S^{01}(\hat{h}_{hinge}) \geq 0.5$? Explain why.

4. **Programming Assignment**

In the programming assignment, you will learn about hyper-parameters and use hold-out validation sets and cross-validation to select a trained model among several. Look at the Jupyter Notebook for more details.