TTIC 31020: Introduction to Machine Learning **Winter 2025**

Homework 4

Due: 3 p.m., January 30th, 2025

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 prefer and encourage that you typeset solutions in LaTeX for written assignments. Handwritten solutions that are not neat, organized and with clear and easy-to-read handwriting will not be graded. 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.*

1. **Online Perceptron and Perceptron Analysis**

   In class we presented PERCEPTRON as an algorithm implementing the $\mathsf{CONSISTENT}_\phi$ 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 $\phi(x) \in \mathbb{R}^d$. The algorithm will take a training set S as input and return a linear predictor $h^{\text{sign}}_{\hat{w}} : x \mapsto \text{sign}\left(\langle \hat{w}, \phi(x)\rangle\right)$ such that $L^{01}_S\left(h^{\text{sign}}_{\hat{w}}\right) = 0$, i.e., $y_i\langle \hat{w}, \phi(x_i)\rangle > 0$ for any $(x_i, y_i) \in S$.

   PERCEPTRON can also be thought as an online learning rule. Start with $w_1 = 0$. At each round t, given an input x_t, we predict the label $\hat{y}_t = h^{\text{sign}}_{w_t}(x_t) = \text{sign}\left(\langle w_t, \phi(x_t)\rangle\right)$. Then we will receive the correct y_t. If we make a mistake, i.e., $\hat{y}_t \neq y_t$, we update $w_{t+1} = w_t + y_t\phi(x_t)$. If the prediction is correct, i.e., $\hat{y}_t = y_t$, we continue with $w_{t+1} = w_t$.

   In this problem we will analyze PERCEPTRON as an online learning rule, an implementation of $\mathsf{CONSISTENT}_\phi$, and as a statistical learning rule.

   To simplify our formulas, we will assume throughout this problem that $\|\phi(x)\| = 1$, i.e., 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:

   $$\forall_t \frac{y_t\langle w^\circ, \phi(x_t)\rangle}{\|w^\circ\|} \geq \gamma. \tag{1}$$

   We will prove that for any sequence such that there exists w° satisfying (1), 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 $x_1, \ldots, x_t$.

   (a) optional Prove that $\frac{\langle w^\circ, w_{t+1}\rangle}{\|w^\circ\|} \geq \gamma M_t$ for all $t = 0, 1, \ldots$.
   Hint: Start with $t = 0$ and prove by induction on t, using (1) to show that whenever PERCEPTRON makes a mistake and updates w_t, $\langle w^\circ, w_{t+1}\rangle \geq \langle w^\circ, w_t\rangle + \gamma\|w^\circ\|$. It means that

the margin condition ensures that we make substantial progress in the direction of w°, with the margin lower bounding this amount of progress.
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) optional Prove that for all $t = 0, 1, \ldots$, $\|w_{t+1}\|^2 \leq M_t$.
Hint: Use induction again, expanding $\|w_t + y_t \phi(x_t)\|$, and noting that we only update w_t when $h_{w_t}(x_t) \neq y_t$, i.e., when $y_t \langle w_t, \phi(x_t) \rangle \leq 0$.

(c) optional Combine the above two inequalities, with the Cauchy-Schwarz inequality $\langle w^\circ, w_t \rangle \leq \|w^\circ\| \|w_t\|$, to conclude that $M_t \leq 1/\gamma^2$. Then for *any* sequence of any length satisfying (1), PERCEPTRON will make at most $1/\gamma^2$ mistakes.

Part II: Implementing CONSISTENT.

We now return to viewing PERCEPTRON as an implementation of the $\mathsf{CONSISTENT}_\phi$ learning rule. Here, given a training set S, we start with $w_1 = 0$. At each iteration t, we 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 $\mathsf{CONSISTENT}_\phi$. 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 3 in homework 3 (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:

$$\gamma(S) = \sup_w \min_{(x_i, y_i) \in S} \frac{y_i \langle w, \phi(x_i) \rangle}{\|w\|}. \tag{2}$$

(a) Prove that if a finite S is realizable by linear predictors, i.e., there exists a function $h_{\hat{w}}^{\text{sign}} : x \mapsto \text{sign}(\langle \hat{w}, \phi(x) \rangle)$ such that $L_S^{01}(h_{\hat{w}}) = 0$, 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 analyzed 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 rely on S being realizable (linearly separable), i.e., 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, PERCEPTRON might never stop. You need to provide 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 only 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 for 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 $w^\star \in \mathbb{R}^d$ such that $\frac{y \langle w^\star, \phi(x) \rangle}{\|w^\star\|} \geq \gamma$ 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 3 in homework 3 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?

Discussion. At each step in the online-to-batch algorithm, we need to compute all the labels for training data to check whether there is a error. This implementation will be highly inefficient in practice. In the programming homework, you will see how we can run perceptron on a fixed training set in a better way.

(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 PERCEPTRON, i.e., this guarantee is specific to the PERCEPTRON rule, and cannot be obtained by all the implementations of CONSISTENT. Show that with other implementations, we cannot bound the error even if the margin is small. For any m, give a feature map and a source distribution (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.

2. $0/1$ **Loss vs Squared Loss vs Hinge Loss**

In machine learning, while we ultimately care about minimizing the $0/1$ loss, it is often difficult to optimize directly because the $0/1$ loss is discontinuous and non-convex. Instead, we minimize a continuous and often convex surrogate loss, such as the squared loss and the hinge loss, which are easier to optimize. However, this does not guarantee that the solution will also achieve a small $0/1$ loss.

Recall that the squared loss is defined as $\ell^{\text{sq}}(h(x); y) = (y - h(x))^2$ and $L_S^{\text{sq}}(h) = \frac{1}{m}\sum_{i=1}^m \ell^{\text{sq}}(h(x_i); y_i)$. The hinge loss is defined as $\ell^{\text{hinge}}(h(x); y) = [1 - yh(x)]_+$ and $L_S^{\text{hinge}}(h) = \frac{1}{m}\sum_{i=1}^m \ell^{\text{hinge}}(h(x_i); y_i)$. We further define $\hat{h}_{\text{sq}} = \arg\min_{h\in\mathcal{H}} L_S^{\text{sq}}(h)$ being the predictor minimizing the squared loss and $\hat{h}_{\text{hinge}} = \arg\min_{h\in\mathcal{H}} L_S^{\text{hinge}}(h)$ being the predictor minimizing the hinge loss

In the questions below, you can just 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) Is it possible that $\inf_{h\in\mathcal{H}} L_S^{01}(h) = 0$ but $L_S^{01}(\hat{h}_{\text{sq}}) \geq 0.5$? Explain why.

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

(c) optional We only consider realizable case in the above. Now we will see the optimal solution of the hinge loss won't guarantee a small $0/1$ loss in the unrealizable setting.
Provide an explicit set of points $S = \{(x_i, y_i) \mid i \in [m]\}$ such that the following two conditions hold simultaneously:

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

ii. $L_S^{01}(\hat{h}_{\text{hinge}}) \geq 0.5$.

Write down:

i. The set of points S such that the above conditions hold. You may pick any number of points to achieve this.

ii. A predictor $h \in \mathcal{H}$ minimizing 0/1 loss. Evaluate $L_S^{01}(h)$ and $L_S^{\text{hinge}}(h)$.

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

(d) challenge We will see the optimal solution of any convex surrogate loss won't guarantee a small 0/1 loss in the unrealizable setting.

For any convex surrogate loss function $\ell^{\text{surrogate}}(z; y)$, provide an explicit set of points $S = \{(x_i, y_i) \mid i \in [m]\}$ such that the following two conditions hold simultaneously:

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

ii. If $\hat{h}_{\text{surrogate}} = \arg\min_{h \in \mathcal{H}} L_S^{\text{surrogate}}(h)$ is the predictor minimizing the surrogate loss, then we have $L_S^{01}(\hat{h}_{\text{surrogate}}) \geq 0.5$.

Write down:

i. The set of points S such that the above conditions hold. You may pick any number of points to achieve this.

ii. A predictor $h \in \mathcal{H}$ minimizing 0/1 loss. Evaluate $L_S^{01}(h)$ and $L_S^{\text{surrogate}}(h)$.

iii. The predictor $\hat{h}_{\text{surrogate}}$. Evaluate $L_S^{01}(\hat{h}_{\text{surrogate}})$ and $L_S^{\text{surrogate}}(\hat{h}_{\text{surrogate}})$.

3. **Programming Assignment**

In the programming assignment, you will learn more about linear predictors. Look at the Jupyter Notebook for more details.