Theory and Algorithm for Machine Learning 2020
National University of Singapore CS5339
Prof Lee Wee Sun

Homework 3 (Due Date Sunday April 19 11.59pm)

Please write the following on your homework:

- Name
- Collaborators (write none if no collaborators)
- Source, if you obtained the solution through research, e.g. through the web.

While you may collaborate, you *must write up the solution yourself*. While it is okay for the solution ideas to come from discussion, it is considered as plagiarism if the solution write-up is highly similar to your collaborator's write-up or to other sources. You solution should be submitted to IVLE workbin. Scanned handwritten solutions are acceptable but must be legible.
Late Policy: A late penalty of 20% per day will be imposed (no submission accepted after 5 late days) unless prior permission is obtained.

---

1. **Perceptron Algorithm as Online Convex Optimization**
   The perceptron algorithm does online learning of a linear threshold function, i.e. $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$ and the prediction $\hat{y}_t = \text{sign}(\langle \mathbf{w}^{(t)}, \mathbf{x}_t \rangle)$. Each time the algorithm makes a mistake, it updates the weight vector $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \eta y_t \mathbf{x}_t$. If the algorithm makes the correct prediction, the weight vector is unchanged.

   (a) Argue that the perceptron algorithm is doing online gradient descent with the loss function $\ell_t(\mathbf{w}^{(t)}, \mathbf{x}_t, y_t) = 0$ for rounds when the prediction is correct, and $\ell_t(\mathbf{w}^{(t)}, \mathbf{x}_t, y_t) = \max\{0, 1 - y_t \langle \mathbf{w}^{(t)}, \mathbf{x}_t \rangle\}$ when the prediction is incorrect.

   Solution: When there is no error, the weight is unchanged, which is consistent with update with the gradient 0, which is also the gradient of the zero function. When the algorithm makes a mistake, the gradient of the loss function is $-y_t \mathbf{x}_t$ which agrees with the online gradient update of $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla \ell(\mathbf{w}^{(t)}, \mathbf{x}, y)$.

   (b) Argue that the total loss upper bounds the total number of mistakes.

   Solution: When the classifier is correct, the loss is zero; hence the loss upper bounds the 0-1 loss in those cases. When the classifier is incorrect, we have $y_t \langle \mathbf{w}^{(t)}, \mathbf{x}_t \rangle \leq 0$ and the loss function $\max\{0, 1 - y_t \langle \mathbf{w}^{(t)}, \mathbf{x}_t \rangle\}$ takes a value that is greater than or equal to 1. Hence the loss upper bounds the 0-1 loss again in those cases and the total loss of the algorithm upper bounds the total 0-1 losses, which is the number of mistakes.

(c) Assume that there exists $\mathbf{w}^*$ such that $y_t\langle \mathbf{w}^*, \mathbf{x}_t\rangle \geq 1$ for all t (i.e. correct with margin at least 1), and argue that $\mathbf{w}^*$ has zero total loss.

Solution: We have $y_t\langle \mathbf{w}^*, \mathbf{x}_t\rangle \geq 1$ for all t, hence loss function always take the value 0 for all t regardless of whether the zero loss or $\max\{0, 1 - y_t\langle \mathbf{w}^*, \mathbf{x}_t\rangle\}$ is used at any time t.

(d) Using SSBD Lemma 14.1, argue that the number of mistakes $M \leq R^2\|\mathbf{w}^*\|^2$, where $R = \max \|\mathbf{x}_t\|$.

Solution: The loss function is convex, hence

$$\ell_t(\mathbf{w}^{(t)}, z_t) - \ell_t(\mathbf{w}^*, z_t) \leq (\mathbf{w}^{(t)} - \mathbf{w}^*)^T \nabla \ell_t(\mathbf{w}^{(t)}, z_t).$$

From SSBD Lemma 14.1,

$$\sum_{t=1}^{T} (\mathbf{w}^{(t)} - \mathbf{w}^*)^T \mathbf{v}_t \leq \frac{||\mathbf{w}^*||^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} ||\mathbf{v}_t||^2.$$

Combining, we get

$$\sum_{t=1}^{T} (\ell_t(\mathbf{w}^{(t)}, z_t) - \ell_t(\mathbf{w}^*, z_t)) \leq \frac{||\mathbf{w}^*||^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} ||\mathbf{v}_t||^2.$$

From the previous part, the total loss of the algorithm is at least as large as the number of mistakes M, and the total loss of $\mathbf{w}^*$ is 0. Furthermore, the subgradient $\|\mathbf{v}_t\| = 0$ when iteration t is classified correctly, and $\|\mathbf{v}_t\| = \|\mathbf{x}_t\| \leq R$ when iteration t is classified incorrectly. Applying all these, we get

$$M \leq \frac{||\mathbf{w}^*||^2}{2\eta} + \frac{\eta}{2} M R^2.$$

Setting $\eta = \frac{\|w^*\|}{\sqrt{M}R}$, we get

$$M \leq \|\mathbf{w}^*\| \sqrt{M} R,$$

giving

$$M \leq \|\mathbf{w}^*\|^2 R^2.$$

(e) Further argue that the mistake bound still holds when η is set to 1.

Solution: The hypothesis always has the form $\mathbf{w}^{(t)} = \sum_{t' \in \mathcal{M}_t} \eta y_{t'} \mathbf{x}_{t'} = \eta \sum_{t' \in \mathcal{M}_t} y_{t'} \mathbf{x}_{t'}$ where $\mathcal{M}_t$ is the set of time instances where the algorithm made mistakes prior to time

t. Note that the sign of $\mathbf{w}^{(t)T}\mathbf{x}_t$ does not depend on η hence it can be set to 1 without changing the classifications made by the algorithm. This is true for all t, hence the mistake bound still holds when η is set to 1.

Comment on the question (not part of grading): The guarantees used here for online learning for convex function with gradient descent is very general. Note that the loss function is arbitrary (although restricted to convex and Lipschitz) and can change from step to step and the instances can be selected adversarially.

2. **Switching Predictors**
Assume that we have a finite class H of expert binary predictors and we have to do T rounds of online predictions. Instead of assuming that there is a perfect predictor in H, we assume that we have a sequence of predictors $h_1, \ldots, h_{k+1}$ from H where we switch from h_i to h_{i+1} at time t_i such that the switching sequence of predictors give perfect predictions. We call such a predictor a k-switching predictor. We would like to apply a version of the Halving algorithm to the problem where a k-switching predictor which makes perfect prediction exists (a version of the weighted majority algorithm can also be used when the k-switching predictor can make mistakes).

(a) We first modify the Halving algorithm to handle weighted hypotheses. Assume that hypothesis h is given a weight w_h such that $\sum_{h\in H} w_h = 1$. In the modified Halving algorithm, we remove a hypothesis from H whenever it makes a mistake (set its weight to 0), and predict using $\arg\max_{r\in\{0,1\}} \sum_{h\in H, h(x)=r} w_h$, i.e. we predict using the label which agrees with the weighted majority of surviving hypotheses. Show that the mistake bound for this algorithm is no more than $\log_2 1/w_{h^*}$ where h^* is a predictor that does not make any error.

Solution: Each mistake will remove at least half of the remaining weights, so after t mistakes the sum of the weights of surviving hypotheses is $W_t \leq (1/2)^t$. We know that h^* makes no mistakes, so the remaining weights must always be at least w^*. This gives

$$\begin{aligned} &(1/2)^t \geq w^* \\ \implies\ &t \leq \log_2 1/w^*. \end{aligned}$$

Note that if the optimal coding scheme is used $\log_2 1/w^*$ is the optimal codelength for h^* giving a connection to minimum description length.

(b) We now assume that there exists a k-switching predictor that makes no mistake. Assume that we use the following weighting scheme for each k switching predictor (predictor with k switches): $\frac{1}{|H|(|H|-1)^k} p^k (1-p)^{T-k-1}$, where p is a user-selected parameter. We will argue that the sum of the weights of all such switching predictors (with $k = 0, \ldots, T-1$) is 1 by mathematical induction.

i. Argue that the statement is true for $T = 1$.

ii. Assume that the statement is true for predictor sequences of length $T - 1$, then show that the statement is true for predictor sequences of length T.

Solution:

i. When $T = 1$, k can only be 0, and we have the uniform distribution over predictors with weight $1/|H|$ each, summing up to 1.

ii. Assume that the statement is true for $T - 1$ rounds. Note that we are assigning weights to every possible predictor sequences, i.e. for $|H|$ predictors and T rounds, there are $|H|^T$ predictor sequences. Assuming that we have all length $T-1$ predictor sequences with appropriate weights assigned, we construct all length T predictor sequences and corresponding weights as follows: for each existing predictor sequence of length $T - 1$, we lengthen the sequence by to be length T by constructing all $|H|$ possible ways to lengthen it. We can do that by either keeping the current predictor unchanged, which multiplies the weight of the predictor sequence by $(1 - p)$, or by switching the current predictor with one of $|H| - 1$ other predictors, to construct $|H| - 1$ new predictors each of which has weight the same as the old sequence weight multiplied by $p/(|H| - 1)$ because we have one more switch than previously. Let the old weight of a predictor sequence s of length $T-1$ be w. Then the total weights of the length T predictor sequences constructed by extending s by one is $w(1-p)+\sum_{h\in H,h\neq h'} w\frac{p}{|H|-1} = w$, which is the same as the weight of s before extension. As the sum of the weights of all the previous length $T - 1$ sequences is 1 by the inductive hypothesis, the sum of the weights of the new sequences of length T will still be 1.

(c) Argue that the number of errors made by the modified Halving algorithm is not more than $\log_2 |H|+k\log_2(|H|-1)+k\log_2 1/p+(T-k-1)\log_2 1/(1-p)$ (approximately $k(\log_2 |H| + \log_2 1/p)$ when p is small) when there is a k-switching predictor that predicts the sequence prefectly.

Solution: Applying $-\log_2$ to $\frac{1}{|H|(|H|-1)^k}p^k(1-p)^{T-k-1}$ gives $\log_2 |H| + k\log_2(|H| - 1)+k\log_2 1/p+(T-k-1)\log_2 1/(1-p)$. Applying part (a) gives the required result.

(d) Give an efficient algorithm for running the modified Halving algorithm on all possible k-switching predictors. The algorithm should run in time $O(|H|^2)$ (or $O(|H|)$ after some optimization) per iteration. (Hint: Try constructing an algorithm by modifying the inductive proof in part (b).)

Solution: The algorithm maintains a weight w_h for each $h \in H$, initialized to $1/|H|$. Prediction is done by weighted majority of the predictors. The weights for predictors that predict incorrectly are set to zero after the label is received. We then construct new

weights for each h, assuming its current weight is w_h (may be 0) as follows: initialize with $(1-p)w_h$, then for each $h' \in H, h' \neq h$, add $w_{h'}p/(|H|-1)$ to the weight.

The runtime of the algorithm as described is $O(|H|^2)$ per iteration. To optimize it to $O(|H|)$, note that $\sum_{h' \neq h} w_{h'}p/(|H|-1)$ can be computed by $(W - w_h)p/(|H|-1)$ where $W = \sum_{h \in H} w_h$. Hence we only need to compute W in time $O(|H|)$ and then update each weight for each h in constant time.

We now argue that it works correctly (that it sums up the weight of all *surviving* predictor sequences that predict the same label at each time step $T' \leq T$). Consider a switching predictor $\bar{h}$ for a sequence of length $T' < T$ with weights $w_{\bar{h}}$ according to the weighting scheme for sequences of length T'. We first argue that the weight associated with predicting using $\bar{h}$ is the same as the weight associated with the sum of all length T predictor sequences that are extensions of $\bar{h}$, where an extension $\bar{h}$ is a predictor sequence of length T which has the same predictors as $\bar{h}$ at time 1 to T'. To see this, note that by the argument in part (b), extending the predictor to multiple predictors at length $T'+1$ then summing up the corresponding weights leaves the weight unchanged at $w_{\bar{h}}$. The argument can be repeated to show that the sum of all predictor sequences of length T that agree with $\bar{h}$ up to time T' (but may disagree in the future) has total weight $w_{\bar{h}}$. Hence summing up weights of all the *surviving* sequences up to time T' that predicted the output r, weighted by the weighting scheme for sequences of length T', would also give the same sum as summing up weights of all *surviving* sequences up to time T weighted by the weighting scheme for sequences of length T.

We now argue that the algorithm computes the sum of weights of all *surviving* sequences that ends with predictor h at time T'. This is true for $T' = 1$. Assume that it is true at time $T'-1$. We see that the algorithm sets all sequences that predicts incorrectly at time $T'-1$ to have weight 0, hence correctly removed all sequences that have ever made a mistake. At time T', for sequences that end with hypothesis h, we sum up all the weights from sequences that did not switch, $(1-p)w_h$, with the weights of all sequences that switched from h' to h (weight $w_{h'}(1-p)/(|H|-1)$) for all $h' \neq h$. This gives the total weight for all *surviving* sequences that end with h at time T'.