CS 7545: Machine Learning Theory **Fall 2018**

Lecture 8: Perceptron & Game Theory

Lecturer: Jacob Abernethy *Scribes: Guanyi Wang, Shimin Zhang*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

8.1 Online Linear Prediction

8.1.1 The Perceptron Algorithm

As mentioned in the last lecture, let T be the length of the sequence of our prediction problem (this sequence can be infinite in length). For each round of the sequence t, nature reveals an outcome $y^t \in \{0, 1\}$. Our perceptron algorithm $\mathcal{A}$ observes a feature vector $\vec{x} \in \mathbb{R}^d$ from nature, then its prediction $\hat{y}^t$ is obtained by setting $\hat{y}^t = \text{sign}(\vec{w}^\top \vec{x})$. Here is the pesudocode:

Initialize the number of mistakes of $\mathcal{A}$ to 0.
For each round $t = 1, \ldots, T$:

1. The algorithm $\mathcal{A}$ selects weights $\vec{w}_t \in \mathbb{R}^d$.
2. Nature selects $\vec{x}^t \in \mathbb{R}^d$ with $\|\mathbf{x}_t\|_2 \leq 1$.
3. The algorithm $\mathcal{A}$ sets $\hat{y}^t = \text{sign}(\vec{w}^{t\top} \vec{x}^t)$.
4. The algorithm $\mathcal{A}$ observes an outcome $y^t \in \{-1, 1\}$.
5. The algorithm updates its weights to $\vec{w}^{t+1}$
6. If $\hat{y}^t \neq y^t$, the number of mistakes increases one.

8.1.2 Perceptron Update Algorithm

The algorithm $\mathcal{A}$ updates its weights via the **perceptron update algorihtm**: initialize the weights vector $\vec{w}^1 = \vec{0}$ at time 0, and at the t^{th} round with $t = 1, \ldots, T$:

$$\vec{w}^{t+1} = \begin{cases} \vec{w}^t & \text{if } y^t \cdot (\vec{w}^{t\top} \vec{x}^t) > 0; \\ \vec{w}^t + y_t \vec{x}^t & \text{otherwise.} \end{cases}$$

As an aside, the professor mentioned that if we set the loss function $\ell(w, \langle x \cdot y \rangle)$ to $\max(0, -y\langle x \cdot y \rangle)$, then the perceptron update algorithm is equivalent to the "stochastic gradient descent" algorithm, where

$$w^{t+1} = w^t - \nabla_w \ell(w^t, \langle x^t \cdot y^t \rangle)$$

8.1.3 Perceptron Update Algorithm Result

Theorem 8.1 ***Upper bound on the number of mistakes in Perception algorithm*** *Assume there exists a weight vector $\vec{w}^*$ such that $\|\vec{w}^*\|_2 \leq \frac{1}{\gamma}$ for any given $\gamma > 0$, and $y^t(\vec{w}^* \vec{x}^t) \geq 1$ for all $t = 1, \ldots, T$. Then the perception algorithm will makes $\leq \frac{1}{\gamma^2}$ mistakes, a surprisingly strong result since the sequence can be infinite in length.*

Proof: Let $\Phi_t := \|\vec{w}^* - \vec{w}_{t+1}\|_2^2$. Notice that $\frac{1}{\gamma^2} \geq \|\vec{w}^*\|_2^2 \geq \|\vec{w}^* - \vec{0}\|_2^2 - \|\vec{w}^* - \vec{w}_{T+1}\|_2^2 = \Phi_0 - \Phi_T$. Then

$$\begin{aligned}
\|\vec{w}^*\|_2^2 \geq & \Phi_0 - \Phi_T \\
= & \sum_{i=1}^{T} \Phi_{t-1} - \Phi_t \\
= & \sum_{i=1}^{T} \|\vec{w}^* - \vec{w}_t\|_2^2 - \|\vec{w}^* - \vec{w}^{t+1}\|_2^2 \\
= & \sum_{\{i:\ y_t \cdot (\vec{w}^{t\top} \vec{x}^t) < 0\}} \|\vec{w}^* - \vec{w}^t\|_2^2 - \|\vec{w}^* - (\vec{w}^t + y^t \vec{y}^t)\|_2^2 \\
= & \sum_{\{i:\ y^t \cdot (\vec{w}^{t\top} \vec{x}^t) < 0\}} 2\Big(\underbrace{y^t((\vec{w}^*)^\top \vec{x}^t)}_{\geq 1} \underbrace{- y^t \cdot (\vec{w}^{t\top} \vec{x}^t)}_{\geq 0}\Big) \underbrace{- y^{t2} \|\vec{x}^t\|_2^2}_{\geq -1}.
\end{aligned}$$

Therefore, $\frac{1}{\gamma^2} \geq \|\vec{w}^*\|_2^2 \geq \sum_{\{i:\ y^t \cdot (\vec{w}^{t\top} \vec{x}^t) < 0\}} 1 = |\{i:\ y^t \cdot (\vec{w}^{t\top} \vec{x}^t) < 0\}| = \#\text{mistakes}[\mathcal{A}]$. ■

8.2 Game Theory

8.2.1 Two-Player Game

Definition 8.2 *Two player game: A (finite-strategies) **two-player game** is defined by a pair of matrices $N, M \in \mathbb{R}^{n \times m}$ where N_{ij} be the pay off to P1 (player 1) when P1 selects action i and P2 selects action j; M_{ij} be the pay off to P2 (player 2) when P1 selects action i and P2 selects action j.*

Definition 8.3 *Zero-sum game: A game is **zero-sum** if $N = -M$.*

Example 1 ***Rock-Paper-Scissors:***

$$M = -N = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

.

8.2.2 Nash's Theorem

Definition 8.4 *Nash Equilibrium: In a two player game, a **Nash Equilibrium**(Neq) is a pair of $\tilde{\mathbf{p}} \in \Delta_n, \tilde{\mathbf{q}} \in \Delta_m$ such that*

$$\begin{aligned}
\forall \mathbf{p} \in \Delta_n : \quad & \tilde{\mathbf{p}}^\top M \tilde{\mathbf{q}} \geq \mathbf{p}^\top M \tilde{\mathbf{q}} \\
\forall \mathbf{q} \in \Delta_m : \quad & \tilde{\mathbf{p}}^\top N \tilde{\mathbf{q}} \geq \tilde{\mathbf{p}}^\top N \mathbf{q}
\end{aligned}$$

note that $\mathbf{p} \top M \mathbf{q}$ is the expectation of the playoff to player one since

$$\mathbf{p} \top M \mathbf{q} = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i q_j M_{ij} = \mathbb{E}_{i \sim p}[\mathbb{E}_{j \sim q}[M_{ij}]]$$

Theorem 8.5 ***Nash's Theorem*** . *Every two player game has a (possibly non-unique) Nash Equilibrium(Neq).*

8.2.3 Von Neumann's Minimax Theorem

Theorem 8.6 ***Von Neumann's Minimax Theorem***

$$\min_{\mathbf{p}\in\Delta_n}\max_{\mathbf{q}\in\Delta_m}\mathbf{p}^\top M\mathbf{q} = \max_{\mathbf{q}\in\Delta_m}\min_{\mathbf{p}\in\Delta_n}\mathbf{p}^\top M\mathbf{q}.$$

Proof:

($\geq$) The $\geq$ direction is straight forward. For any $\mathbf{p}\in\Delta_n, \mathbf{q}\in\Delta_m$, we have $\mathbf{p}^\top M\mathbf{q} \geq \min_{\mathbf{p}\in\Delta_n}\mathbf{p}^\top M\mathbf{q}$. Then for any $\mathbf{p}\in\Delta_n$, take maximum over $\mathbf{q}$ on both sides, we have $\max_{\mathbf{q}\in\Delta_m}\mathbf{p}^\top M\mathbf{q} \geq \max_{\mathbf{q}\in\Delta_m}\min_{\mathbf{p}\in\Delta_n}\mathbf{p}^\top M\mathbf{q}$. Therefore $\min_{\mathbf{p}\in\Delta_n}\max_{\mathbf{q}\in\Delta_m}\mathbf{p}^\top M\mathbf{q} \geq \max_{\mathbf{q}\in\Delta_m}\min_{\mathbf{p}\in\Delta_n}\mathbf{p}^\top M\mathbf{q}$ holds.

($\leq$) We will use regret minimization to sketch a proof due to time constraint. Imagine that the game is played repeatedly with each player using the exponential weights average algorithm to update their strategy. For round $t = 1\ldots T$:

- Player 1 select strategy $\mathbf{p}^t \in \Delta_n$
- Player 2 select strategy $\mathbf{q}^t \in \Delta_m$
- Player 1 observes loss vector $M\mathbf{q}^t$
- Player 2 observes loss vector $-\mathbf{p}^t M$

Since we are using the exponential weighted average algorithm, our average payoff over the sequence is less than or equal to the payoff of the best expert plus some regret term:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{p}^t\langle M\cdot\mathbf{q}^t\rangle = \frac{1}{T}\sum_{t=1}^{T}\mathbf{p}^t\ell^t \leq \min_i \frac{1}{T}\sum_{t=1}^{T}e_i\langle M\cdot\mathbf{q}^t\rangle + \frac{\text{Regret}}{T}$$

We also know that the regret term $\frac{\text{regret}}{T}$ grows sublinearly and is $O(\frac{1}{\sqrt{T}})$ Therefore it vanishes with sufficiently large values of T, completing the $\leq$ direction.

■