Written Assignment 7

Due: Friday 04/18/2025 @ 11:59pm EST

Disclaimer

I encourage you to work together, I am a firm believer that we are at our best (and learn better) when we communicate with our peers. Perspective is incredibly important when it comes to solving problems, and sometimes it takes talking to other humans (or rubber ducks in the case of programmers) to gain a perspective we normally would not be able to achieve on our own. The only thing I ask is that you report who you work with: this is **not** to punish anyone, but instead will help me figure out what topics I need to spend extra time on/who to help. When you turn in your solution (please use some form of typesetting: do **NOT** turn in handwritten solutions), please note who you worked with.

Question 1: Datasets with Weights (25 points)

Consider a dataset in which each data point $\left(x^{(i)}, y_{gt}^{(i)}\right)$ is associated with some weight $r^{(i)} > 0$. If we want to use a mean squared error for our loss function (like we want to do in temporal difference learning), our objective now becomes:

$$L(\vec{\theta}) = \frac{1}{2N} \sum_{i=1}^{N} r^{(i)} \left(y_{gt}^{(i)} - f_{\vec{\theta}}(x^{(i)})\right)^2$$

For now, lets simplify $f_{\vec{\theta}}$ to be a linear model (which in an earlier homework you showed that any completely-linear neural network could be reduced to this) $f_{\vec{\theta}}(x) = \vec{\theta}^T \phi(x)$. Plugging this in:

$$L(\vec{\theta}) = \frac{1}{2N} \sum_{i=1}^{N} r^{(i)} \left(y_{gt}^{(i)} - \vec{\theta}^T \phi(x^{(i)})\right)^2$$

Derive an expression for the optimum $\vec{\theta}^*$ that minimizes this loss function.

1 Solution

$$L(\vec{\theta}) = \frac{1}{2N}\sum_{i=1}^{N} r^{(i)}\Big(y_{gt}^{(i)} - \vec{\theta}^T\phi(x^{(i)})\Big)^2 \qquad \text{Given}$$

$$\frac{\partial L(\vec{\theta})}{\partial \vec{\theta}} = \frac{1}{2N}\sum_{i=1}^{N} \frac{\partial}{\partial \vec{\theta}} r^{(i)}\Big(y_{gt}^{(i)} - \vec{\theta}^T\phi(x^{(i)})\Big)^2 \qquad \text{Find derivative}$$

$$= \frac{1}{2N}\sum_{i=1}^{N} \frac{\partial}{\partial \vec{\theta}}\Big(r^{(i)}y_{gt}^{(i)2} - 2r^{(i)}y_{gt}^{(i)}\vec{\theta}^T\phi(x^{(i)}) + r^{(i)}(\vec{\theta}^T\phi(x^{(i)}))^2\Big) \qquad \text{Foil + distribute}$$

$$= \frac{1}{2N}\Big(\sum_{i=1}^{N} 0 - 2r^{(i)}y_{gt}^{(i)}\phi(x^{(i)}) + 2r^{(i)}\vec{\theta}^T\phi(x^{(i)})^2\Big) \qquad \text{Power rule}$$

$$= \frac{1}{2N}\Big(\sum_{i=1}^{N} 2r^{(i)}\vec{\theta}^T\phi(x^{(i)})^2 - 2r^{(i)}y_{gt}^{(i)}\phi(x^{(i)})\Big) \qquad \text{Rearranged}$$

$$= \frac{1}{N}\Big(\sum_{i=1}^{N} r^{(i)}\vec{\theta}^T\phi(x^{(i)})^2 - r^{(i)}y_{gt}^{(i)}\phi(x^{(i)})\Big) \qquad \text{Factor out 2}$$

$$\frac{1}{N}\Big(\sum_{i=1}^{N} r^{(i)}\vec{\theta}^T\phi(x^{(i)})^2 - r^{(i)}y_{gt}^{(i)}\phi(x^{(i)})\Big) = 0 \qquad \text{Set to 0 to find minimum}$$

$$\sum_{i=1}^{N} r^{(i)}\vec{\theta}^T\phi(x^{(i)})^2 - r^{(i)}y_{gt}^{(i)}\phi(x^{(i)}) = 0 \qquad \text{Multiply by } N$$

$$\sum_{i=1}^{N} r^{(i)}\vec{\theta}^T\phi(x^{(i)})^2 - \sum_{i=1}^{N} r^{(i)}y_{gt}^{(i)}\phi(x^{(i)}) = 0 \qquad \text{Break up sum}$$

$$\sum_{i=1}^{N} r^{(i)}\vec{\theta}^T\phi(x^{(i)})^2 = \sum_{i=1}^{N} r^{(i)}y_{gt}^{(i)}\phi(x^{(i)}) \qquad \text{Rearrange terms}$$

$$\sum_{i=1}^{N} \vec{\theta}^T\phi(x^{(i)})^2 = \sum_{i=1}^{N} y_{gt}^{(i)}\phi(x^{(i)}) \qquad \text{Cancel out } r^{(i)}$$

$$\vec{\theta}^T\sum_{i=1}^{N} \phi(x^{(i)})^2 = \sum_{i=1}^{N} y_{gt}^{(i)}\phi(x^{(i)}) \qquad \text{Bring out } \vec{\theta}^T \text{ term}$$

$$\vec{\theta}^T = \frac{\sum_{i=1}^{N} y_{gt}^{(i)}\phi(x^{(i)})}{\sum_{i=1}^{N} \phi(x^{(i)})^2} \qquad \text{Solve for } \vec{\theta}$$

$$\vec{\theta}^T = \sum_{i=1}^{N} \frac{y_{gt}^{(i)}\phi(x^{(i)})}{\phi(x^{(i)})^2} \qquad \text{Simplify}$$

$$\vec{\theta}^T = \sum_{i=1}^{N} \frac{y_{gt}^{(i)}}{\phi(x^{(i)})} \qquad \text{Simplify}$$

$$\therefore \boxed{\vec{\theta}^* = \sum_{i=1}^{N} \frac{y_{gt}^{(i)}}{\phi(x^{(i)})}}$$

Note: $\vec{\theta}^*$ is a column vector

Question 2: The Precariousness of RL (25 points)

In the previous problem, you considered a loss function where each point was weighted with a nonzero coefficient $r^{(i)}$:

1. Show that whenever we have a dataset containing dupicate data points, we are effectively creating the scenario from the previous problem.
2. Let us relax the weighting terms so that every weight $r^{(i)} \geq 0$ instead of > 0. Show that this loss function over a dataset of samples can be converted into a weighted sum over all possible data points.
3. What happens to terms with weights of 0? Do they impact the solution at all? How do we know that the model will perform well on these points?

2.1 Solution

$$L_{weighted}(\vec{\theta}) = \frac{1}{2N}\sum_{i=1}^{N} r^{(i)}\left(y_{gt}^{(i)} - \vec{\theta}^T\phi(x^{(i)})\right)^2 \quad \text{Weighted MSE loss}$$

$$L_{unweighted}(\vec{\theta}) = \frac{1}{2N}\sum_{i=1}^{N} \left(y_{gt}^{(i)} - \vec{\theta}^T\phi(x^{(i)})\right)^2 \quad \text{Unweighted MSE loss}$$

$$\mathcal{D}^{(i)} = \left\{\left(x^{(i_1)}, y^{(i_1)}\right), \ldots, \left(x^{(i_{n_i})}, y^{(i_{n_i})}\right)\right\} \quad \text{Data point } i \text{ has } n_i \text{ duplicates}$$

$$\left(x^{(i_1)}, y^{(i_1)}\right) = \left(x^{(i_2)}, y^{(i_2)}\right) = \cdots = \left(x^{(i_{n_i})}, y^{(i_{n_i})}\right)$$

$$\mathcal{D} = \bigcup_{i=1}^{M} \mathcal{D}^{(i)}, \quad N = \sum_{i=1}^{M} n_i \quad \text{Dataset has points with } n_i \text{ dupes}$$

$$\mathcal{U} = \left\{(x, y) \in \mathcal{D}\right\}, \quad |\mathcal{U}| = M \quad \text{Set of points w/o duplicates of size } M$$

$$L_{unweighted}(\vec{\theta}) = \frac{1}{2N}\sum_{i=1}^{M} n_i\left(y_{gt}^{(i)} - \vec{\theta}^T\phi(x^{(i)})\right)^2 \quad \text{Unweighted MSE loss with dupes}$$

$$\therefore L_{weighted} \equiv L_{unweighted} \quad n_i \text{ functions as weight } r^{(i)}$$

$\therefore$ Including duplicate data points in the dataset allows you to weight the data points where points are weighted by how often they appear in the dataset as a duplicate. This results in the same loss function as weighted MSE loss. $\square$

2.2 Solution

$$L(\vec{\theta}) = \frac{1}{2N}\sum_{i=1}^{N} r^{(i)}\Big(y_{gt}^{(i)} - \vec{\theta}^T\phi(x^{(i)})\Big)^2 \qquad \text{Weighted MSE loss}$$

$$r^{(i)} \geq 0 \qquad \text{Weights can be 0 by assumption}$$

$$\mathcal{Z} = \mathcal{X} \times \mathcal{Y} \qquad \text{Infinite set of possible data points}$$

$$\mathcal{D} = \big\{(x^{(i)}, y^{(i)})\big\}_{i=1}^{N} \subset \mathcal{Z} \qquad \text{Our finite dataset}$$

$$\mathcal{U} = \big\{(x, y) \in \mathcal{D}\big\}, \quad \big|\mathcal{U}\big| = M \qquad \text{Set of points w/o duplicates of size } M$$

$$\rho(x^{(i)}, y^{(i)}) = \begin{cases} r^{(i)} & (x^{(i)}, y^{(i)}) \in \mathcal{U} \\ 0 & (x^{(i)}, y^{(i)}) \notin \mathcal{U} \end{cases} \qquad \text{Only give nonzero weight to points in } \mathcal{U}$$

$$\boxed{L(\vec{\theta}) = \frac{1}{2M}\sum_{(x^{(i)},y^{(i)})\in\mathcal{Z}} \rho(x^{(i)}, y^{(i)})\Big(y - \vec{\theta}^T\phi(x^{(i)})\Big)^2}$$

$\therefore$ Although the set of all possible data points $\mathcal{Z}$ is infinitely large, since $r^{(i)} \geq 0$, we can express our finite loss function as an infinite loss function by ignoring points not in our finite dataset and points that have already been seen by our model by setting their weights to 0. We handle duplicate data points in $\mathcal{D}$ by including them in the loss function once, and setting the weight to all remaining duplicates to 0. Our adjusted weight function ρ only returns a nonzero weight for points in our finite dataset that have yet to be seen. Instead of trying to normalize by infinite points, we are able to normalize by the size of the finite dataset without duplicates. $\square$

2.3 Solution

We follow the same notation and assumptions as part 2.2

$$\forall (x^{(i)}, y^{(i)}) \notin \mathcal{U}, \quad \rho(x^{(i)}, y^{(i)}) = 0 \qquad \text{As defined in 2.2}$$

$$L(\vec{\theta}) = \frac{1}{2N}\sum_{(x^{(i)},y^{(i)})\in\mathcal{Z}} \rho(x^{(i)}, y^{(i)})\Big(y - \vec{\theta}^T\phi(x^{(i)})\Big)^2 \qquad \text{Infinite loss function}$$

$$= \frac{1}{2N}\Big(\sum_{(x^{(i)},y^{(i)})\in\mathcal{U}} \rho(x, y)\big(y - \vec{\theta}^T\phi(x^{(i)})\big)^2 + \sum_{(x^{(i)},y^{(i)})\notin\mathcal{U}} 0 \cdot \big(y - \vec{\theta}^T\phi(x^{(i)})\big)^2\Big) \qquad \text{Rewritten}$$

$$= \frac{1}{2M}\sum_{(x^{(i)},y^{(i)})\in\mathcal{U}} \rho(x, y)\big(y - \vec{\theta}^T\phi(x^{(i)})\big)^2 \qquad 0 \text{ cancels out}$$

$$\therefore L(\vec{\theta}) = \frac{1}{2N}\sum_{(x^{(i)},y^{(i)})\in\mathcal{Z}} \rho(x, y)\Big(y - \vec{\theta}^T\phi(x^{(i)})\Big)^2 = \frac{1}{2M}\sum_{(x^{(i)},y^{(i)})\in\mathcal{U}} \rho(x, y)\big(y - \vec{\theta}^T\phi(x^{(i)})\big)^2$$

So we have proven that the infinite loss function is equivalent to the weighted sum of data points in our finite dataset without duplicates. This clearly shows that the terms with weights of 0 are completely ignored and therefore **do not impact the solution at all**. The model does **not** learn from these points, meaning we cannot guarantee generalization to those points since they were never learned by the model. If the model happened to learn from points similar to the ignored point, the model may perform well. On the other hand, if there were no similar points in the finite dataset, the model would perform poorly. Since we do not know the contents of the finite dataset, we cannot know for certain the performance of the model on these points.