Midterm

Nurseiit Abdimomyn – 20172001

18/10/2021

1 (a) Supervised learning is when the inputs have their corresponding truth outputs prepared beforehand and the model is trained on the set of input and desired output. While the unsupervised learning does not necessarily have the desired/expected output. The reinforcement learning is when the model itself tries to pave the course for the solution on its own given the dimensions and the criterion of the problem at hand.

(b) Zero-one loss function is discontinuous and does not necessarily provide full context to the binary classification problem to say, for example, how confident is the current model of the answers given.

(e) Overfitting in a model can happen when there is not much input variants to train over and the model keeps getting "fit" over those values again and again.

(d) A* search is complete as it will find a solution if it exists. Also, in terms of optimality, it will make sure the next estimate is kept at minimum while expanding to further branches.

2 (a) We have $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$

So we derive for the gradient:

$\frac{\partial L(\theta)}{\partial \theta_0} = \frac{\partial}{\partial \theta_0} \sum\limits_i (\theta_0 + \theta_1 x + \theta_2 x^2 - y_i)^2$

$= \sum\limits_i \frac{\partial}{\partial \theta_0} (\theta_0 + \theta_1 x + \theta_2 x^2 - y_i)^2$

$= \sum\limits_i 2(\theta_0 + \theta_1 x + \theta_2 x^2 - y_i)$

Doing the same for θ_1 and θ_2 we get:

$\frac{\partial L(\theta)}{\partial \theta_1} = \sum_i 2(\theta_0 + \theta_1 x + \theta_2 x^2 - y_i) * x$

$\frac{\partial L(\theta)}{\partial \theta_2} = \sum_i 2(\theta_0 + \theta_1 x + \theta_2 x^2 - y_i) * x^2$

(b) $\frac{\partial L(\theta)}{\partial \theta} = X^T X\theta + X^T X\theta - 2X^T y =$

$= 2X^T X\theta - 2X^T y = 0$. So, $X^T X\theta = X^T y$.

Solving for θ we get:

$\theta^* = (X^T X)^- 1X^T y$.

(c) computations can take longer if dimensions are high, but also there might be some precision errors while computing the inverse of the matrix, if it exists in the first place that is.

3 Sorry, I am too lazy to derive and then compute aaaall this by hand.

4 (a) False – dfs could find solution way earlier than it's tail (backtracking of visiting other nodes) could start.

(b) True – (just a guess, can't explain)

(d) True – (technically you could apply anything, but will it be optimal? that's the question :))

(d) True – if the graph is finite

(e) False – not admissible, can overestimate