CMPUT 652: Reinforcement Learning with Robots **Fall 2019**

Lecture 4: September 12

Instructor: Rupam Mahmood *Scribe: Dhawal Gupta*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

4.1 Discussion: Last Lecture

The last lecture discussed the formulation and optimization of the objective function, along with the methods of averaging, gradient descent and stochastic gradient descent as well as the bias-variance trade-off and bias and variance in using the square distance for Gaussian parameter estimation.

4.2 Notation

The notation that we are going to follow in these notes have been borrowed from [**?**] and is described below.

X	Random variables (scalar)
x	Values of random variables or scalar functions
$\mathbf{x}$	Quantities required to be real-valued coloumn vectors (even random variables)
$\mathbf{X}$	Matrices
$\mathbf{X}^T, \mathbf{x}^T$	The transpose of a matrix or vector
$[\mathbf{x}]_i$	The *i*-th element of the vector.
$[\mathbf{X}]_{i:}$	The *i*-th row of the matrix. A row vector
$[\mathbf{X}]_{:j}$	The *j*-th column of the matrix. A column vector
$[\mathbf{X}]_{ij}$	The *i*,*j*-th element of the matrix.
$\mathbf{x}_t$	The vector at time t
X_t	The random variable at time t

4.3 Linear Function Approximation

We will be using the mean squared error as the objective function for the approximation of the parameter in the case where our modelled function f is a linear model.

$$f(\mathbf{x}) = \mathbf{x}^T\theta \tag{4.1}$$

where $\mathbf{x}, \theta \in \mathbb{R}^p$, p is the number of features in our input. Y is the output variable which takes in real value, i.e. $Y \in \mathbb{R}$

We assume that both $(\mathbf{x}, Y)$ are sampled from a joint probability distribution that is not know, but we have access to the samples from that probability distribution i.e. $\{(\mathbf{x}_i, Y_i)\}_{i=0}^{t} \sim P(\mathbf{x}, Y)$ also reffered as a training data set.

4.3.1 Exact vs. Sample-based MSE objectives

Here we list the different forms of the objective function for the cases of exact and sampled based estimates of the objective function.

The exact form of the squared error loss function can also written as $\mathcal{V}(\theta)$ i.e.

$$\mathcal{V}(\theta) = MSE = \mathbb{E}[(Y - \mathbf{x}^T\theta)^2] \tag{4.2}$$

The sample based objective function for dataset $\{(\mathbf{x}_i, Y_i)\}_{i=1}^{t}$:

$$M\hat{S}E = \frac{1}{t}\sum_{i=1}^{t}(Y_i - \mathbf{x}_i^T\theta)^2 \tag{4.3}$$

And the squared error at time T = t, is written as

$$SE_t = (Y_t - \mathbf{x}_t^T\theta)^2 \tag{4.4}$$

By the law of large numbers as $t \to \infty$, $M\hat{S}E \to MSE$ given i.i.d sampling of $(\mathbf{x}_i, Y_i) \sim P(\mathbf{x}, Y)$.

4.3.2 Optimization of the Objective function

We will be going through the different approaches to optimizaiton of the objective functions starting from the one step , exact optimization procedure moving onto **Gradient Descent** (GD) and **Stochastic Gradient Descent** (SGD) finding the optimal θ i.e. θ^* for the linear model 4.1. Here we are taking the expectation over $(\mathbf{x}, Y)$.

Equations 4.5 provides an exact optimization process in the case of availability of Expectation of different variables.

$$\begin{aligned}
\mathcal{V}(\theta) &= \mathbb{E}[(\underset{1\times1}{Y} - \underset{1\times p}{\mathbf{x}}{}^T \underset{p\times1}{\theta})^2] \text{ (from 4.2)} \\
&= \mathbb{E}[(Y - \mathbf{x}^T\theta)^T(Y - \mathbf{x}^T\theta)] \\
&= \mathbb{E}[\underset{1\times1}{Y^2} + \underset{1\times p}{\theta^T}\underset{p\times1}{\mathbf{x}}\underset{1\times p}{\mathbf{x}^T}\underset{p\times1}{\theta} - 2\underset{1\times p}{\theta^T}\underset{p\times1}{\mathbf{x}}\underset{1\times1}{Y}] \\
&= \mathbb{E}[Y^2] + \theta^T\,\mathbb{E}[\mathbf{x}\mathbf{x}^T]\theta - 2\theta^T\,\mathbb{E}[\mathbf{x}Y] \\
\text{Doing the following substitution } \mathbf{A} &= \mathbb{E}[\mathbf{x}\mathbf{x}^T], \mathbf{b} = \mathbb{E}[\mathbf{x}Y], c = \mathbb{E}[Y^2] \text{we get..} \qquad (4.5) \\
&= \theta^T\mathbf{A}\theta - 2\theta^T\mathbf{b} + c \\
&\text{taking the gradient} \\
\nabla_\theta\mathcal{V}(\theta) &= 2\mathbf{A}\theta - 2\mathbf{b} \\
&\text{which gives us} \\
\theta^* &= \mathbf{A}^{-1}\mathbf{b}
\end{aligned}$$

So 4.5 gives us the following after putting the values of $\mathbf{A}, \mathbf{b}, c$

$$\theta^* = \mathbb{E}[\mathbf{x}\mathbf{x}^T]^{-1}\,\mathbb{E}[\mathbf{x}Y] \tag{4.6}$$

We can extend the above result to generalize to the sample based method i.e. placing this for 4.3[1]

$$\theta_{t+1} = (\frac{1}{t}\sum_{i=1}^{t}\mathbf{x}_i\mathbf{x}_i^T)^{-1}(\frac{1}{t}\sum_{i=1}^{t}\mathbf{x}_iY_i) \tag{4.7}$$

Writing the step wise update rule for GD i.e.

$$\begin{aligned}\theta_{t+1}^{GD} &= \theta_t^{GD} - \alpha_t\nabla_{\theta_t^{GD}}\mathcal{V}(\theta_t^{GD})\\ \theta_{t+1}^{GD} &= \theta_t^{GD} - 2\alpha_t(\mathbf{A}\theta_t^{GD} - \mathbf{b})(\text{ from 4.5})\end{aligned} \tag{4.8}$$

Similarly this can be written for the case of SGD i.e.

$$\begin{aligned}\theta_{t+1} &= \theta_t - \alpha_t\nabla_{\theta_t}(Y_t - \theta_t^T\mathbf{x}_t)^2\\ \theta_{t+1} &= \theta_t + 2\alpha_t(Y_t - \theta_t^T\mathbf{x}_t)\mathbf{x}_t\end{aligned} \tag{4.9}$$

Note : We can try to write these formulation of 4.3 in a time based manner as well, I tried but I am not able to figure one out, just similar to the case of moving average

4.3.3 Gradient Descent Method and Convergence

Below we expand the θ_{t+1}^{GD} term to understand the behaviour of convergence.

$$\begin{aligned}\theta_{t+1}^{GD} =&(I - \alpha\mathbf{A})\theta_t^{GD} + \alpha\mathbf{b} \quad (\ \mathbf{A},\, \mathbf{b} \text{ from 4.5 \& } \alpha_t = \frac{\alpha}{2}\ )\\ =&\alpha\mathbf{b} + (I - \alpha\mathbf{A})[(I - \alpha\mathbf{A})\theta_{t-1}^{GD} + \alpha\mathbf{b}]\\ =&\alpha\mathbf{b} + \alpha(I - \alpha\mathbf{A})\mathbf{b} + (I - \alpha\mathbf{A})^2\theta_{t-1}^{GD}\\ =&\alpha[I + (I - \alpha\mathbf{A}) + (I - \alpha\mathbf{A})^2 + ... + (I - \alpha\mathbf{A})^t]b + (I - \alpha\mathbf{A})^{t+1}\theta_0^{GD}\\ &\text{Using the Geometric Series Summation}\\ =&\cancel{\alpha}(\cancel{I} - \cancel{I} + \cancel{\alpha}\mathbf{A})^{-1}[I - (I - \alpha\mathbf{A})^{t+1}]b + (I - \alpha\mathbf{A})^{t+1}\theta_0^{GD}\\ =&\ \mathbf{A}^{-1}[I - (I - \alpha\mathbf{A})^{t+1}]\mathbf{b} + (I - \alpha\mathbf{A})^{t+1}\theta_0^{GD}\end{aligned} \tag{4.10}$$

This series is convergent as $(I - \alpha\mathbf{A})^t$ goes to zero for the following conditions

$$\rho(I - \alpha\mathbf{A}) < 1 \tag{4.11}$$

where the spectral radius $\rho(\mathbf{B})$ for a matrix $\mathbf{B}$ is defined as

$$\rho(\mathbf{B}) = \underset{i}{max}|eig_i(\mathbf{B})| \tag{4.12}$$

[1] I have tried to show a separate proof of this, which can be easily inferred from 4.5 as well

Hence for 4.11 we need the following to hold

$$Re(eig_i \mathbf{A}) > 0 \tag{4.13}$$

So in the limit of $t \to \infty$ the value converges to $\mathbf{A}^{-1}\mathbf{b} = \theta^*$, hence telling us that the bias goes to zero along with the more no of examples observed.

$$\lim_{t\to\infty} \theta_t^{GD} = \mathbf{A}^{-1}\mathbf{b} \tag{4.14}$$

As $\mathbf{A}$ is symmetric matrix, all eigen values of $\mathbf{A}$ are Real.

Note: This part has been added by scriber (need to check validity) : Finding the conditions on eigen values of matrix $\mathbf{A}$ i.e. λ_a

taking $\mathbf{x}$ as the eigen vector and λ as the corresponding eigen value for $(I - \alpha\mathbf{A})$

$$\begin{aligned}(I - \alpha\mathbf{A})\mathbf{x} &= \lambda\mathbf{x} \\ (1-\lambda)\mathbf{x} &= \alpha\mathbf{A}\mathbf{x} \\ \frac{(1-\lambda)}{\alpha}\mathbf{x} &= \mathbf{A}\mathbf{x} \\ \text{from above it seems that } \mathbf{x} \text{ is eigen vector for } A \text{ where } A \text{ has eigen values } &\frac{(1-\lambda)}{\alpha}\end{aligned} \tag{4.15}$$

$$\begin{aligned}\frac{(1-\lambda)}{\alpha} &= \lambda_a \\ \lambda &= (1 - \alpha\lambda_a)\end{aligned} \tag{4.16}$$

From 4.16 we can get the conditions that need to be applied on top of the eigenvalues of $\mathbf{A}$ i.e. λ_a.

From 4.11 and 4.12 we get the following conditions. As λ_a is always real we dont need to worry about complex eigenvalues. We have that

$$\begin{aligned}|\lambda| &< 1 \text{ (for all } \lambda) \\ |(1-\alpha\lambda_a)| &< 1 \text{ (from 4.16)}\end{aligned} \tag{4.17}$$

Case 1: $\lambda_a \leq \frac{1}{\alpha}$

$$\begin{aligned}(1 - \alpha\lambda_a) &< 1 \\ \lambda_a &> 0\end{aligned} \tag{4.18}$$

This gives us

$$0 < \lambda_a \leq \frac{1}{\alpha} \tag{4.19}$$

Case 2: $\lambda_a > \frac{1}{\alpha}$

$$\begin{aligned}-(1 - \alpha\lambda_a) &< 1 \\ \lambda_a &< \frac{2}{\alpha}\end{aligned} \tag{4.20}$$

This gives us

$$\frac{1}{\alpha} < \lambda_a < \frac{2}{\alpha} \tag{4.21}$$

From results 4.19 and 4.21, we get:

$$0 < \lambda_a < \frac{2}{\alpha}. \tag{4.22}$$

4.4 Non Linear Function Approximation

We define a non linear function as

$$f(\mathbf{x}, \theta, \mathbf{W}) = \theta^T g(\mathbf{W}\mathbf{x}) \tag{4.23}$$

$$\Phi = g(\mathbf{W}\mathbf{x}) \tag{4.24}$$

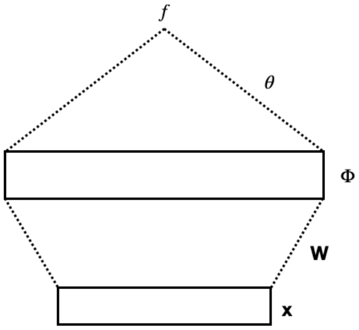

Figure 4.1: A visualization of the said non linear function

Where $\theta \in \mathbb{R}^{n\times 1}$, $\mathbf{W} \in \mathbb{R}^{n\times p}$ is the weight matrix, and both are the parameters for our non linear model. The feature vector is $\mathbf{x} \in \mathbb{R}^{p\times 1}$. The function $g : \mathbb{R}^{n\times 1} \to \mathbb{R}^{n\times 1}$ where g an element-wise applied non-linearity e.g. tanh, ReLU. Φ is an additional notation to aid in understanding further.

4.4.1 SGD and Backpropagation

Putting the non linear approximation 4.23 in the SGD equation and optimizing for parameter θ_{t+1}.

$$\begin{aligned}\theta_{t+1} &= \theta_t - \alpha_t \nabla_{\theta_t}(Y_t - f(\mathbf{x}_t, \theta_t, \mathbf{W}_t))^2 \\ &= \theta_t + 2\alpha_t(Y_t - \theta^T\Phi_t)\Phi_t\end{aligned} \tag{4.25}$$

Optimizing w.r.t. to the Weight matrix $\mathbf{W}$

$$\begin{aligned}\mathbf{W}_{t+1} &= \mathbf{W}_t - \alpha_t \nabla_{\mathbf{W}_t}(Y_t - \theta_t^T\Phi)^2 \\ &= \mathbf{W}_t + 2\alpha_t(Y_t - \theta_t^T\Phi_t)\nabla_{\mathbf{W}_t}(\theta_t^T\Phi_t) \\ &= \mathbf{W}_t + 2\alpha_t(Y_t - \theta_t^T\Phi_t)(\theta_t \circ g'(\mathbf{W}_t\mathbf{x}_t))\mathbf{x}_t^T \text{ (where $\circ$ is the Hadamard Product)}\end{aligned} \tag{4.26}$$

Hence the final equations

$$\mathbf{W}_{t+1} = \mathbf{W}_t + 2\alpha_t(Y_t - \theta_t^T\Phi_t)(\theta_t \circ g'(\mathbf{W}_t\mathbf{x}_t))\mathbf{x}_t^T \tag{4.27}$$

So Equations 4.25 and4.27 gives us the update rule for the parameters of the Non linear function approximator.