CPSC 486/586: Probabilistic Machine Learning 01/23/23

Lecture 3

Lecturer: Andre Wibisono *Scribe: Stephen Yin*

1 Announcements

1. Scribe note 1 posted, sign up for scribing
2. PS 1 due next Wed 12 pm
3. Discussion section Friday 2-3pm in Watson A60

2 Linear Regression

Given $x_1, \dots, x_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{R}$, we want to fit a linear model:

$$f(x) = w^T x, \text{ for some } w \in \mathbb{R}^d$$

with $w^T x = \langle w, x \rangle = w \cdot x$. Note: $\vec{x}$ is a column vector and w^T is a row vector.

How do we find this? By minimizing a loss function:

$$\min_{w \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

where we say $f(w) = \frac{1}{2} \sum_{i=1}^{n} (y_i - w^T x_i)^2$. We can rewrite this as a quadratic form in w by the following:

$$\begin{aligned}
f(w) &= \frac{1}{2} \sum_{i=1}^{n} (y_i - w^T x_i)^2 \\
&= \frac{1}{2} \sum_{i=1}^{n} (y_i^2 - 2y_i(w^T x_i) + (w^T x_i)^2) \\
&= \frac{1}{2} \sum_{i=1}^{n} y_i^2 - w^T \sum_{i=1}^{n} x_i y_i + \frac{1}{2} \sum_{i=1}^{n} w^T (x_i x_i^T) w \\
&\qquad \text{(numerically, note that } w^T x_i = x_i^T w \text{ as each are just numbers)} \\
&= \frac{1}{2} \sum_{i=1}^{n} y_i^2 - w^T \sum_{i=1}^{n} x_i y_i + \frac{1}{2} w^T \left(\sum_{i=1}^{n} x_i x_i^T \right) w \\
&= c + w^T b + \frac{1}{2} w^T A w
\end{aligned}$$

Where we have used:

$$A = \sum_{i=1}^{n} x_i x_i^T = XX^T,\ X = \begin{pmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{pmatrix},\ X^T = \begin{pmatrix} - & x_1^T & - \\ & \vdots & \\ - & x_n^T & - \end{pmatrix}$$

$$b = -\sum_{i=1}^{n} x_i y_i = Xy$$

$$c = \frac{1}{2}\sum_{i=1}^{n} y_i^2 = \frac{1}{2}\|y\|^2$$

Summarizing,

$$X = \begin{pmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{pmatrix},\ y = \begin{pmatrix} y_i \\ \vdots \\ y_n \end{pmatrix}$$

$$A = XX^T,\ \ b = -Xy,\ \ c = \frac{1}{2}\|y\|^2 \rightarrow \text{Objective function for Linear Regression}$$

$$f(w) = \frac{1}{2}w^T Aw + w^T b + c \in \mathbb{R}$$

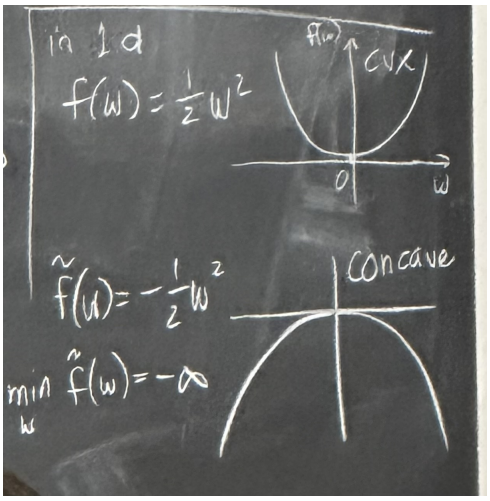
$$\nabla f(w) = Aw + b \in \mathbb{R}^d$$

$$\nabla^2 f(w) = A \succeq 0 \in \mathbb{R}^{d\times d} \implies f \text{ is convex}$$

Note: $A = XX^T \succeq 0 \iff \forall u \in \mathbb{R}^d, u^T Au \geq 0$. This is the Loewner partial order ($\succeq$) for positive semi-definite matrices.

Note 2: Sometimes the X is defined to be a matrix with the x_i vectors as the rows. Then everything still holds but with $A = X^T X$ and $b = -X^T y$. The X matrix is also sometimes called the design matrix.

Example 1. *As an example in 1D, we examine* $f(w) = \frac{1}{2}w^2$,



The minimzer of this is $w^* = \arg\min_w f(w)$ *that satisfies* $\nabla f(w^*) = 0 \implies Aw^* + b = 0 \implies w^* = -A^{-1}b$ *if* A *is invertible. In the case* A *is not invertible, we can run some optimization algorithm.*

3 Optimization of a Quadratic

Take $f : \mathbb{R}^d \to \mathbb{R}$, where $f(x) = \frac{1}{2}x^T Ax,\ A \in \mathbb{R}^{d\times d},\ A = A^T \succeq 0$, where we want to find $\min_{x\in\mathbb{R}^d} f(x)$. We can do the following:

1. (In continuous time) Run Gradient Flow dynamics $(X_t)_{t\geq 0}$. Here, we want to have X_t for time t converge to a minimum of $f(x)$. To do so, we can take the rate of change of X_t as follows:

$$\frac{d}{dt}X_t = \dot{X}_t = -\nabla f(X_t)$$

$$f(x) = \frac{1}{2}x^T Ax \implies \dot{X}_t = -AX_t \implies X_t = e^{-tA}X_0$$

Where $e^M = I + M + \frac{M^2}{2!} + \frac{M^3}{3!} + \cdots \iff$ If $M = M^T = U\Sigma U^t$, then $e^M = Ue^{\Sigma}U^t$.

Again we take the example in one dimension $d = 1$. $X_t \in \mathbb{R}$, $A > 0$.

$$\boxed{\dot{X}_t = -AX_t} \in \mathbb{R}$$
$$\iff \frac{d}{dt}\log X_t = \frac{\dot{X}_t}{X_t} = -A$$
$$\overset{\int_0^t}{\iff} \log X_t - \log X_0 = -At$$
$$\iff X_t = X_0 e^{-At}$$

2. (In discrete time) Dynamics to algorithms. Consider the general continuous-time dynamics of the form

$$\dot{X}_t = v(X_t), \quad v : \mathbb{R}^d \to \mathbb{R}^d$$

where $v(x) = (v_1(x), \ldots, v_n(x))$. (E.g., $v(x) = -\nabla f(x)$ for gradient flow). We can discretize this continuous process by taking learning rate η.

(a) Forward Method.

$$\frac{x_{k+1} - x_k}{\eta} = v(x_k) \iff x_{k+1} = x_k + \eta v(x_k)$$
$$\lim_{\eta \to 0} \frac{x_{t+\eta} - x_t}{\eta} = \dot{X}_t$$

(b) Backward Method.

$$\frac{x_{k+1} - x_k}{\eta} = v(x_{t+1}) \iff x_{k+1} = x_k + \eta v(x_{k+1})$$

3. Why Gradient Flow?

$$\dot{X}_t = -\nabla f(X_t)$$

It is nice for optimization because it is a descent flow, meaning that such an optimization update will let us minimize our objective function.

$$\frac{d}{dt} f(X_t) \overset{\text{chain rule}}{=} \langle \nabla f(X_t), \dot{X}_t \rangle \overset{\text{GF}}{=} -\|\nabla f(X_t)\|^2 \leq 0$$

Also, there is nice convergence theory in continuous and discrete time.

$$\underline{\text{GF}}\text{: } \dot{X}_t = -\nabla f(X_t).$$

Forward method = Gradient descent with

$$X_{k+1} = X_k - \eta \nabla f(X_k) = (I - \eta \nabla f)(X_k)$$

Backward method = Proximal point method with

$$\begin{aligned} X_{k+1} &= \arg\min_X f(X) + \frac{1}{2\eta} \|X - X_{k+1}\|^2 \\ \Rightarrow X_{k+1} &= X_k - \eta \nabla f(X_{k+1}) \\ (I + \eta \nabla f)(X_{k+1}) &= X_k \\ X_{k+1} &= (I + \eta \nabla f)^{-1}(X_k) \end{aligned}$$

Fact: This operator $\arg\min_{x \in \mathbb{R}^d} \{f(x) + \frac{1}{2\eta} \|x - x_k\|^2\} = \text{prox}_{f,\eta}(x_k)$ is called the proximal operator, or the prox function.

For $\min_x f(x) = \frac{1}{2} x^T A x$:

- Gradient Flow. $\dot{X}_t = -AX_t \implies X_t = e^{-tA}x_0$.
- Gradient Descent. $X_{k+1} = X_k - \eta AX_k = (I - \eta A)X_k \Rightarrow X_k = (I - \eta A)^k X_0$
- Proximal Gradient. $X_{k+1} = X_k - \eta AX_{k+1} = (I + \eta A)^{-1}X_k \Rightarrow X_k = (I + \eta A)^{-k} X_0$

4 Review

Linear Regression. We want to solve $\frac{1}{2}\sum_{i=1}^{n}(y_i - w^T x_i)^2 = \frac{1}{2}w^T Aw + b^T w + c$ which is to use optimization on $\frac{1}{2}w^T Aw + b^T w + c$ which implies exact solution $w^* = -A^{-1}b$ which we can compute via algos like Gradient Descent or Proximal Gradient. Next class: MLE of Gaussian model.