Homework Assignment 4
Basic ideas of Bayesian machine learning and PCA

Kailin Zheng kz882

April 9, 2019

1 Linear regression I (10 points)

Let vector $y = (y_1, y_2, ..., y_k)$,

$$X = \begin{bmatrix} ...x_1... \\ ...x_2... \\ ... \\ ...x_q... \end{bmatrix}$$

$$y_k^* = X^T W$$

$$\begin{aligned}
\Delta(M^*(x), M, x) &= \frac{1}{2}\sum_{k=1}^{q}(y_k^* - y_k)^2 \\
&= \frac{1}{2}\sum_{k=1}^{q}(x_k^T W - y_k)^2 \\
&= \frac{1}{2}(XW - y)^T(XW - y) \\
&= \frac{1}{2}(W^T X^T - y^T)(XW - y) \\
&= \frac{1}{2}(W^T X^T XW - W^T X^T y - y^T XW + y^T y)
\end{aligned} \tag{1}$$

We solve for W when $\nabla_W \Delta(M^*(x), M, x) = 0$

Useful matrix derivative formula:

$$\frac{\partial u^T v}{\partial x} = u^T \frac{\partial v}{\partial x} + v^T \frac{\partial u}{\partial v}$$

$$\frac{dX^T A}{dx} = A^T$$

$$\frac{dAW}{dW} = A$$

$$
\begin{aligned}
\frac{\partial W^T X^T X W}{\partial W} &= W^T \frac{\partial X^T X W}{\partial W} + (X^T X W)^T \frac{\partial W}{\partial W} \\
&= W^T X^T X + W^T (X^T X)^T \\
&= W^T X^T X + W^T X^T X \\
&= 2W^T X^T X
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
\frac{\partial W^T X^T y}{\partial W} &= (X^T y)T \\
&= y^T X
\end{aligned} \tag{3}
$$

$$
\frac{\partial y^T X W}{\partial W} = y^T X \tag{4}
$$

Plug results of (2),(3),(4) to get derivative of (1):

$$
\begin{aligned}
\nabla_W \Delta(M^*(x), M, x) &= \frac{1}{2}\left(\frac{\partial W^T X^T X W}{\partial W} - \frac{\partial W^T X^T y}{\partial W} - \frac{\partial y^T X W}{\partial W}\right) \\
&= \frac{1}{2}(2W^T X^T X - y^T X - y^T X)
\end{aligned} \tag{5}
$$

Derivative = 0:

$$W^T X^T X - y^T X = 0$$

Solve for W:

$$W^T X^T X = y^T X$$

Take transpose of both sides, we get:

$$X^T X W = X^T y$$

When $X^T X$ is invertible, $W = (X^T X)^{-1} X^T y$. But

We are not sure whether $(X^T X)^{-1}$ exists, so: singular vector decomposition,

$$X = U \Lambda V^T$$

where U is unitary matrix and $U^T U = I$, Λ is a diagonal matrix

Plug SVD of X, we get:

$$
\begin{aligned}
(U\Lambda V^T)^T (U \Lambda V^T) W &= V \Lambda U^T y \\
V \Lambda U^T U \Lambda V^T W &= V \Lambda U^T y \\
V \Lambda^2 V^T W &= V \Lambda U^T y \\
W &= V \Lambda^{-1} U^T y
\end{aligned}
$$

We showed that $X^+ = V\Lambda^{-1} U^T$ is the Moore–Penrose pseudoinverse of $X = U \Lambda V^T$.

So $W = X^+ y$ where $X^+ = V \Lambda^{-1} U^T$; when $X^T X$ is invertible, it is equivalent to $W = (X^T X)^{-1} X^T y$.

2 Linear regression II (20 points)

1. For an example x, y, the L2 loss is:

$$\mathbb{E}[(y - \hat{f}(x;\Theta))^2] = \mathbb{E}^2[y] - 2\mathbb{E}[y]\mathbb{E}[\hat{f}(x;\Theta)] + \mathbb{E}^2[\hat{f}(x;\Theta)]$$

plug in $Y = f(X) + \epsilon$,

$$\begin{aligned}
\mathbb{E}[(y - \hat{f}(x;\Theta))^2] &= (f(x)+\epsilon)^2 - 2(f(x)+\epsilon)\mathbb{E}[\hat{f}(x;\Theta)] + \mathbb{E}^2[\hat{f}(x;\Theta)] \\
&= f^2(x) + 2\epsilon f(x) + \epsilon^2 - 2f(x)\mathbb{E}[\hat{f}(x;\Theta)] - 2\epsilon\mathbb{E}[\hat{f}(x;\Theta)] + \mathbb{E}^2[\hat{f}(x;\Theta)] \\
&= f^2(x) - 2f(x)\mathbb{E}[\hat{f}(x;\Theta)] + \mathbb{E}^2[\hat{f}(x;\Theta)] + 2\epsilon f(x) - 2\epsilon\mathbb{E}[\hat{f}(x;\Theta)] + \epsilon^2 \\
&= (f(x) - \mathbb{E}[\hat{f}(x;\Theta)])^2 + 2\epsilon(f(x) - \mathbb{E}[\hat{f}(x;\Theta)]) + \epsilon^2
\end{aligned} \tag{6}$$

Note that ϵ is given as a constant with zero mean. The derivative of the whole thing is $2(f(x) - \mathbb{E}[\hat{f}(x;\Theta)])$. Since this is a quadratic function, its minimum value is achieved when its derivative =0, that is to say, $f(x) - \mathbb{E}[\hat{f}(x;\Theta)] = 0$.

The overall minimum L2 loss is achieved when $f(x) = \hat{f}(x;\Theta)$ for all x.

2. By definition,

$$var(\hat{f}(x_0;\Theta)) = \mathbb{E}[(\hat{f}(x_0;\Theta))^2] - \mathbb{E}^2[\hat{f}(x_0;\Theta)]$$

$$bias(\hat{f}(x_0;\Theta)], f(x_0)) = \mathbb{E}[f(x) - \hat{f}(x_0;\Theta)]$$

$$bias^2 = \mathbb{E}^2[f(x)] - 2\mathbb{E}[f(x)]\mathbb{E}[\hat{f}(x_0;\Theta)] + \mathbb{E}^2[\hat{f}(x_0;\Theta)]$$

$$\begin{aligned}
\mathbb{E}[(y - \hat{f})^2] &= \mathbb{E}[(f + \epsilon - \hat{f})^2] \\
&= \mathbb{E}[(f + \epsilon - \hat{f} + \mathbb{E}(\hat{f}) - \mathbb{E}(\hat{f}))^2] \\
&= \mathbb{E}[(f - \mathbb{E}(\hat{f}) + \epsilon + \mathbb{E}(\hat{f}) - \hat{f})^2] \\
&= \mathbb{E}[(f - \mathbb{E}(\hat{f}))^2] + \mathbb{E}[\epsilon^2] + \mathbb{E}[(E(\hat{f}) - \hat{f})^2] + 2\mathbb{E}[(f - \mathbb{E}(\hat{f}))\epsilon] \\
&+ 2\mathbb{E}[(E(\hat{f}) - \hat{f})\epsilon] + 2\mathbb{E}[(f - \mathbb{E}(\hat{f}))(\mathbb{E}(\hat{f}) - \hat{f})] \\
&= (f - \mathbb{E}(\hat{f}))^2 + \mathbb{E}[\epsilon^2] + \mathbb{E}[(E(\hat{f}) - \hat{f}))^2] \\
&= (f - \mathbb{E}(\hat{f}))^2 + Var[y] + Var[\hat{f}] \\
&= (\mathbb{E}[f(x) - \hat{f}(x_0;\Theta)])^2 + Var(\hat{f}(x_0;\Theta)) + \sigma^2
\end{aligned} \tag{7}$$

3 Dimensionality reduction (10 points)

(a) With lower dimensionality, we can simplify calculations - we can converge faster. Also, since data points with dimensionality <=3 are easy to plot, we can visualize data better with lower dimensionality. We can then understand and improve our model more efficiently.

(b) We may have some data loss for key information.

For example, for PCA, the less q we have, the more efficient to do the computation, but also the larger reconstruction loss we have. It means we lose some relatively important information when we reduce dimensionality.