ECE-GY 6143 Machine Learning HW 03

Tongda Xu, tx506

February 8, 2019

1 Problem 1

1.1 a

sales data

1.2 b

consider i^{th} trial as $\mathbf{x}^T = x_1, x_2, ..., x_n$, mapping the numerical value and frequency to $[0, 1]$ and feed into $\mathbf{x}^T$, and let prediction $y = \mathbf{x}^T\beta$

1.3 c

both mapping to $[0, 1]$

1.4 d

mapping numeric score to $[0, 1]$
mapping binary score to $\{0, 1\}$
no review neutral to the mean of all other scores $\{x\}$

1.5 e

proportion of *good* in all reviews

2 Problem 3

2.1 a

$\Phi_1(\mathbf{x}) = x_1 e^{-x_1 - x_2}, \beta_1 = a_1$
$\Phi_2(\mathbf{x}) = x_2 e^{-x_1 - x_2}, \beta_2 = a_2$

2.2 b

$\Phi_1(\mathbf{x}) = u(x-1), \beta_1 = a_1$
$\Phi_2(\mathbf{x}) = u(x-1)x, \beta_2 = a_2$
$\Phi_3(\mathbf{x}) = u(1-x), \beta_3 = a_3$
$\Phi_4(\mathbf{x}) = u(1-x)x, \beta_4 = a_4$

2.3 c

$\Phi_1(\mathbf{x}) = e^{-x_2}, \beta_1 = e^{a_2}$
$\Phi_2(\mathbf{x}) = x_1 e^{-x_2}.\beta_2 = a_1 e^{a_2}$

3 Problem 4

3.1 a

$\beta.shape = [M+N+1,]$

3.2 b

$$\begin{bmatrix} 0 & \dots & 0 & x_0 & \dots & 0 \\ y_0 & \dots & 0 & x_1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y_{k-1} & \dots & y_{k-M} & x_k & \dots & x_{k-N} \end{bmatrix}$$

$A.shape = [T, M+N+1]$

3.3 c

considering the dot product $A^T * A$ can be blocked into four parts, for the product, **only difference between row and column index would impact result**:

$$B = \begin{bmatrix} A^T[:M,:] * A[:,:M] & A^T[:M,:] * A[:,M:] \\ A^T[M:,:] * A[:,:M] & A^T[M:,:] * A[:,M:] \end{bmatrix}$$

$A^T[:M,:] * A[:,:M] : B[i,j] = R_{yy}(|i-j|)$
$A^T[:M,:] * A[:,M:] : B[i,j] = R_{xy}(i-j)$
$A^T[M:,:] * A[:,:M] : B[i,j] = R_{xy}(j-i)$
$A^T[M:,:] * A[:,M:] : B[i,j] = R_{xx}(|i-j|)$

For same reason, $C = A^T y$ could be simplified as:
$C[:M] : C[i,j] = R_{yy}(|i-j|)$
$C[M:] : C[i,j] = R_(xy)((i-j))$

4 Problem 6

4.1 a

```
x[:,2] *= x[:,1]
yhat = np.dot(x, beta)
```

4.2 b

```
A = np.dot(x.reshape[x.shape[0],1],beta.reshape[1,beta.shape[0]])
yhat = np.dot(-np.exp(A), alpha)
```

4.3 c

```
dim = np.ones(x.shape[1])
Sig_X = np.dot(x**2, dim).reshape(x.shape[0],1)
Sig_Y = np.dot(y**2, dim).reshape(1,y.shape[0])
dist = Sig_X + Sig_Y - 2*np.dot(x,y)
```