Statistical Modelling & Machine Learning HW1 Solution

1. For fixed σ^2,

$$\begin{aligned}
\max_{\boldsymbol{\beta}} l(\boldsymbol{\beta}; \boldsymbol{y}, \boldsymbol{x}, \sigma^2) &= \max_{\boldsymbol{\beta}} -n \log 2\sigma - \frac{1}{\sigma} \sum |y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}| \\
&\equiv \max_{\boldsymbol{\beta}} -n \log 2\sigma - \frac{1}{\sigma} \sum |y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}| \\
&\equiv \min_{\boldsymbol{\beta}} \frac{1}{\sigma} \sum |y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}| \\
&\equiv \min_{\boldsymbol{\beta}} \sum |y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}|.
\end{aligned}$$

Thus, MLE of $\boldsymbol{\beta}$ can be obtained by minimizing $\sum |y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}|$.

2. $Var(\epsilon_t) = Var(e_t + \theta e_{t-1}) = Var(e_t) + \theta^2 Var(e_{t-1}) = (1+\theta^2)\sigma^2$.
Since $E(\epsilon_t) = E(e_t + \theta e_{t-1}) = 0$,

$$\begin{aligned}
Cov(\epsilon_t, \epsilon_{t-h}) &= E(\epsilon_t \epsilon_{t-h}) - E(\epsilon_t)E(\epsilon_{t-h}) \\
&= E(\epsilon_t \epsilon_{t-h}) = E[(e_t + \theta e_{t-1})(e_{t-h} + \theta e_{t-h-1}) \\
&= E(e_t e_{t-h}) + \theta E(e_{t-1} e_{t-h}) + \theta E(e_t e_{t-h-1}) + \theta^2 E(e_{t-1} e_{t-h-1}).
\end{aligned}$$

For $h = 1$, since $Cov(e_t, e_{t'}) = 0,\ t \neq t'$, $Cov(\epsilon_t, \epsilon_{t-1}) = \theta\sigma^2$
For $h = 2$, $Cov(\epsilon_t, \epsilon_{t-2}) = 0$
For $h > 2$, $Cov(\epsilon_t, \epsilon_{t-h}) = 0$
Since $Cov(\boldsymbol{\epsilon}) = Cov(\boldsymbol{y})$,

$$Cov(\boldsymbol{y}) = \sigma^2 \begin{pmatrix} (1+\theta)^2 & \theta & 0 & \cdots & 0 \\ \theta & (1+\theta)^2 & \theta & \cdots & 0 \\ 0 & \theta & (1+\theta)^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & (1+\theta)^2 \end{pmatrix}.$$

3. Let d_{ij} be the Euclidean distance between obs i and obs j, $i, j = 1, \ldots, 4$.

$$d_{12} = d_{21} = \sqrt{17},\ d_{13} = d_{31} = \sqrt{10},\ d_{14} = d_{41} = \sqrt{5},$$

$$d_{23} = d_{32} = \sqrt{5},\ d_{24} = d_{42} = \sqrt{18},\ d_{34} = d_{43} = \sqrt{5}.$$

Since the ith row and jth column of the spatial weight matrix is $w_{ij} = e^{d_{ij}}$,

$$\boldsymbol{W} = \begin{pmatrix} 0 & 0.016 & 0.042 & 0.106 \\ 0.016 & 0 & 0.016 & 0.014 \\ 0.042 & 0.106 & 0 & 0.106 \\ 0.106 & 0.014 & 0.106 & 0 \end{pmatrix}.$$

4. (1)

$$
\begin{aligned}
f(y) &= \frac{e^{-\mu}\mu^y}{y!} \\
&= \exp(-\mu + y\log\mu - \log y!).
\end{aligned}
$$

$\theta = \log\mu$, $\phi = 1$, $a(\phi) = \phi$, $b(\theta) = e^{\theta}$, $c(y,\phi) = -\log y!$. Thus, the Poisson distribution belongs to the exponential family.

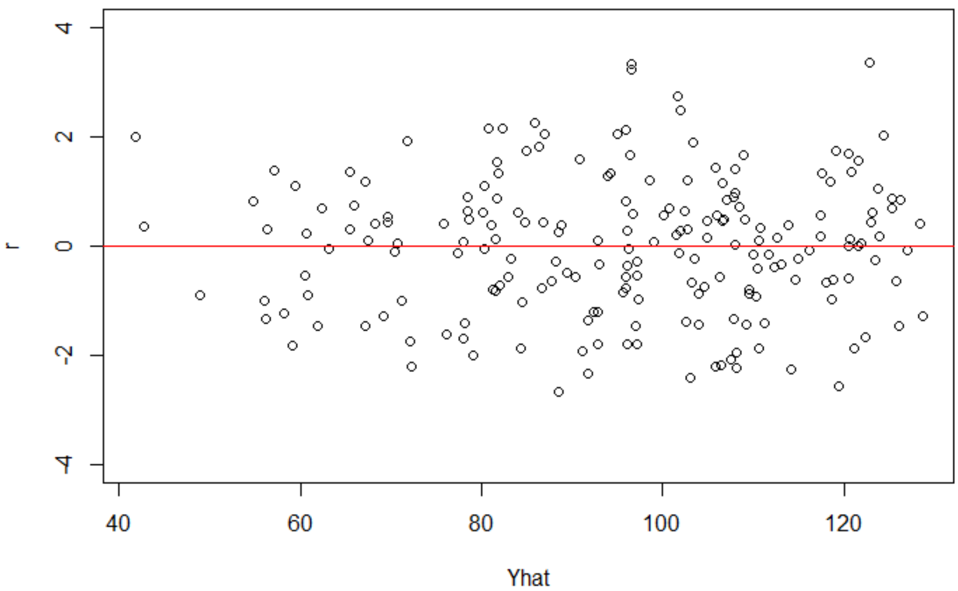
(2) The canonical link function: $\theta = g(\mu) = \log\mu$.

5.
```
> # (1) ---------------------------------------------------
> dat = read.csv('Q5.csv')
>
> #install.packages('gam')
> library(gam)
> fit = gam(Y ~ s(X1,5) + s(X2,5)+s(X3,5),data=dat)
> summary(fit)

Call: gam(formula = Y ~ s(X1, 5) + s(X2, 5) + s(X3, 5), data = dat)
Deviance Residuals:
    Min      1Q  Median      3Q     Max
-46.745 -10.857   1.169  10.824  60.287

(Dispersion Parameter for gaussian family taken to be 353.1694)

    Null Deviance: 149538.9 on 199 degrees of freedom
Residual Deviance: 64983.04 on 183.9996 degrees of freedom
AIC: 1758.289

Number of Local Scoring Iterations: NA

Anova for Parametric Effects
           Df Sum Sq Mean Sq  F value    Pr(>F)
s(X1, 5)    1  42086   42086 119.1658 < 2.2e-16 ***
s(X2, 5)    1    109     109   0.3097    0.5785
s(X3, 5)    1  13501   13501  38.2271 3.967e-09 ***
Residuals 184  64983     353
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova for Nonparametric Effects
            Npar Df  Npar F     Pr(F)
(Intercept)
```

```
s(X1, 5)          4  0.5276    0.7156
s(X2, 5)          4  0.6030    0.6609
s(X3, 5)          4 14.7288 1.858e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> par(mfrow=c(1,3))
> plot(fit)

> # X2 is irrelevant because the fitted line is moving around zero
> # and it is not significant from ANOVA.

> # (2) ----------------------------------------------------
> fit = lm(Y~X1 + I(X3^2) + X3, data=dat)
> summary(fit)

Call:
lm(formula = Y ~ X1 + I(X3^2) + X3, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-46.137 -11.729   0.961  11.023  61.109

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  31.6220     4.5233   6.991 4.19e-11 ***
X1            4.8742     0.4554  10.704  < 2e-16 ***
I(X3^2)      -0.9626     0.1293  -7.446 2.97e-12 ***
X3           13.8073     1.5747   8.768 8.71e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.62 on 196 degrees of freedom
Multiple R-squared:  0.5454,Adjusted R-squared:  0.5384
F-statistic: 78.38 on 3 and 196 DF,  p-value: < 2.2e-16

>
> X = model.matrix(Y~X1 + I(X3^2) + X3, data=dat)
> Y = as.vector(dat$Y)
> beta.new = fit$coefficient      # initial parameter.
> W = diag(rep(1,length(Y)))
> mdif = 100000
>
```

```
> while(mdif > 0.000001)
+ {
+   Yhat = X %*% beta.new
+   r = Y - Yhat
+   Z = cbind(1,Yhat)
+   gam.hat = solve(t(Z) %*% W %*% Z) %*% t(Z) %*% W %*% abs(r)
+   sigma = Z %*% gam.hat
+   S = diag(as.vector(sigma^2))
+
+   if (is.non.singular.matrix(S)) W = solve(S)
+   else W = solve(S + 0.000000001*diag(rep(1,nrow(S))))
+
+   beta.old = beta.new
+   beta.new = solve(t(X) %*% W %*% X) %*% t(X) %*% W %*% Y
+   mdif = max(abs(beta.new - beta.old))
+ }
>
> beta.new
                  [,1]
(Intercept) 32.9021286
X1           4.8814112
I(X3^2)     -0.9258888
X3          13.2910214
>
> # (3) --------------------------------------------------
> Yhat = X %*% beta.new
> sigma = Z %*% gam.hat
> r = (Y - Yhat)/sigma
>
> # Residual plot
> par(mfrow=c(1,1))
> plot(Yhat,r,ylim=c(-4,4))
> lines(c(0,150),c(0,0),col='red')
```

```
> # (4) --------------------------------------------------
# >Y and X1 have linear relationship and
# >Y and X3 have quadratic relationship.
```


4
2
0
-2
-4
r
40
60
80
100
120
Yhat