COMS 4721 HW1
Machine Learning for Data Science

Name: Sung-Yen Liu
UNI: sl3763

Problem 1.

Part 1.

(a) $P(x_1, x_2, \ldots, x_N|\pi) = p(x_1|\pi)\,p(x_2|\pi)\ldots p(x_N|\pi) = \prod_{i=1}^{N} p(x_i|\pi) = \prod_{i=1}^{N} \pi^{x_i}(1-\pi)^{1-x_i}$

(b) $\ln p(x_1, x_2, \ldots x_N|\pi) = \sum_{i=1}^{N} \ln p(x_i|\pi) = \sum_{i=1}^{N} [x_i \ln \pi + (1-x_i)\ln(1-\pi)]$

$$\nabla_\pi \sum_{i=1}^{N} [x_i \ln\pi + (1-x_i)\ln(1-\pi)] = 0 \Rightarrow \frac{1}{\pi}\sum_{i=1}^{N} x_i + \frac{1}{1-\pi}\sum_{i=1}^{N}(1-x_i) = 0$$

assume $\sum_{i=1}^{N} x_i = S \Rightarrow \frac{S}{\pi} - \frac{N-S}{1-\pi} = 0 \Rightarrow \pi = \frac{S}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i$

(c) Consider a simple problem: Given a bias coin which generates 60 'heads' and 40 'tails' in 100 toss, what would be the probability π of coin to get 'head' can maximize the chance of this observation?

① By using MLE: $\pi = \frac{1}{100} \times 60 = 0.6$

② By intuition: $\pi = 0.6$ is most likely to generate the same observation

Thus MLE explains the result mathematically and matches our intuition. Besides, when N becomes larger, the Law of large numbers makes both results closer.

Part 2.

$f(k;\lambda) = Pr(X=k) = \frac{\lambda^k e^{-\lambda}}{k!}$

(a) Let $D = (x_1, x_2, \ldots, x_N)$

$$P(D) = \prod_{i=1}^{N} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} = \frac{1}{\prod_{i=1}^{N} x_i!} \times \lambda^{\sum_{i=1}^{N} x_i} e^{-N\lambda}$$

(b) $\ln P(D) = \sum_{i=1}^{N} \ln\left(\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right) = \ln\left(\frac{e^{-N\lambda}\cdot\lambda^{\sum_{i=1}^{N} x_i}}{\prod_{i=1}^{N} x_i!}\right) = -N\lambda + \sum_{i=1}^{N} x_i \ln(\lambda) - \sum_{i=1}^{N} \ln(x_i!)$

$$\nabla_\lambda \ln P(D) = 0 \Rightarrow -N + \frac{1}{\lambda}\sum_{i=1}^{N} x_i = 0 \Rightarrow \lambda = \frac{1}{N}\sum_{i=1}^{N} x_i$$

(c) Similar to problem (c) of part 1.

With a set of observations of poisson (λ) (mean = variance = λ) $[x_1, x_2, \ldots, x_n]$

① By intuition, since $E[X] = \lambda$, one would guess λ_{ML} is the mean of the observations

② By MLE: $\lambda = \frac{1}{N}\sum_{i=1}^{N} x_i$, which is also the mean of X can maximize the probability to get the same observations.

By concluding (c) of part 1 & 2, one may incline to guess maximum likelihood parameter as the mean (expectation) of random variables at first glance intuitively in Bernoulli and Poisson random variables. On the other hand, MLE well explains the reason of making such guess to get the same observations when N is large and →∞.

COMS 4721 HW1

Machine Learning for Data Science

Name: Sung-Yen Liu

UNI: sl3763

Problem 2

$\lambda \sim \text{Gam}(\lambda | a, b)$

$\text{Gam}(\lambda | a, b) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda}$

$\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$

(a) With Bayes rule, $p(y_0 | x_0, y, X) = \int_{\mathbb{R}^d} p(y_0 | x_0, w)\, p(w | y, X)\, dw$

Given conditional probability $\Pr(N=n | \lambda) = \frac{\lambda^n e^{-\lambda}}{n!}$

$$\Pr(N=n) = \int_0^\infty \frac{\lambda^n e^{-\lambda}}{n!} \cdot \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda} d\lambda = \frac{b^a}{n! \cdot \Gamma(a)} \int_0^\infty \lambda^{n+a-1} \cdot e^{-(1+b)\lambda} d\lambda$$

Assume $(1+b)\lambda = t$

$\Rightarrow \lambda = \frac{t}{1+b}$

$$= \frac{b^a}{n!\,\Gamma(a)} \int_0^\infty \left(\frac{t}{b+1}\right)^{n+a-1} e^{-t}\, d\left(\frac{t}{b+1}\right) = \frac{\Gamma(n+a)}{n!\,\Gamma(a)} \cdot b^a \cdot \left(\frac{1}{b+1}\right)^{n+a}$$

$$= \frac{\Gamma(n+a)}{\Gamma(n+1)\Gamma(a)} \cdot \left(\frac{b}{b+1}\right)^a \cdot \left(\frac{1}{b+1}\right)^n = \binom{n+a-1}{n} \left(\frac{b}{b+1}\right)^a \left(\frac{1}{b+1}\right)^n$$

Poisson - Gamma mixture distribution is in the same form as Negative Binomial distribution

$f(k; r, p) = \Pr(X=k) = \binom{k+r-1}{k} p^k (1-p)^r \quad k=0,1,2,\ldots$

(b)

$\Pr(N=n) = \binom{n+a-1}{n} \left(\frac{1}{b+1}\right)^n \left(\frac{b}{b+1}\right)^a, \quad n=0,1,2,\ldots$

Assume $\frac{b}{b+1} = p \Rightarrow \Pr(N=n) = \binom{n+a-1}{n} p^a (1-p)^n$

While $\sum_{n=0}^{\infty} \Pr(N=n) = 1$

$$\Rightarrow E[N] = \sum_{n=0}^{\infty} n \binom{n+a-1}{n} p^a (1-p)^n = \sum_{n=1}^{\infty} \frac{(n+a-1)!}{(n-1)!(a-1)!} p^a (1-p)^n = \sum_{n=1}^{\infty} \frac{a(1-p)}{p} \binom{n+a-1}{n-1} p^{a+1} (1-p)^{n-1}$$

Assume $z = n-1$

$$= \frac{a(1-p)}{p} \sum_{z=0}^{\infty} \binom{z+1+a-1}{z} p^{a+1} (1-p)^z = \frac{a(1-p)}{p} \sum_{z=0}^{\infty} \binom{z+a}{z} p^{a+1} (1-p)^z = \frac{a(1-p)}{p} = \frac{a}{b}$$

$$E[N^2] = \sum_{n=0}^{\infty} n^2 \binom{n+a-1}{n} p^a (1-p)^n = \sum_{n=0}^{\infty} [(n-1)n + n] \binom{n+a-1}{n} p^a (1-p)^n = \frac{a}{b} + \sum_{n=2}^{\infty} \frac{(n+a-1)!}{(n-2)!(a-1)!} p^a (1-p)^n$$

$$= \frac{a}{b} + \sum_{n=2}^{\infty} \frac{a(a+1)(1-p)^2}{p^2} \binom{n+a-1}{n-2} p^{a+2} (1-p)^{n-2} = \frac{a}{b} + \frac{a(a+1)(1-p)^2}{p^2} \sum_{z=0}^{\infty} \binom{z+a+1}{z} p^{a+2} (1-p)^z = \frac{a}{b} + \frac{a^2+a}{b^2}$$

Assume $z=n-2$

$$\text{Var}[X] = E[N^2] - (E[N])^2 = \frac{a}{b} + \frac{a^2+a}{b^2} - \frac{a^2}{b^2} = \frac{a}{b} + \frac{a}{b^2} = \frac{a+ab}{b^2}$$

Negative Binomial distribution can be considered as a generalization of Poisson distribution (when $\lambda = \frac{a}{b} = \frac{a+ab}{b^2}$), and with additional parameter b to shape and scale the pdf of the distribution. In other special cases, Negative Binomial distribution can be Pascal distribution.

Machine Learning for Data Science Name: Sung-Yen Liu UNI: sl3763

COMS 4721 HW1

Problem 3.

Part 1.

Matlab Code:

```
%% ML hw 1 - 3.1 %%
rec = zeros(1000,1); % for recording
for i = 1:1000
        % split to train / test
        rp = randperm(392);
        Xtrain = X(rp(1:372),:);
        Xtest = X(rp(373:end),:);
        Ytrain = y(rp(1:372),:);
        Ytest = y(rp(373:end),:);

        % Analytical form of solving linear regression
        w = inv(Xtrain'*Xtrain)*Xtrain'*Ytrain;

        % prediction
        ypred = Xtest*w;
        ydiff = abs(ypred - Ytest);
        rec(i) = sum(ydiff)/20;
end

w
mean(rec)
std(rec)
```

(a) w_{ML} = [23.4649; -0.5736; 0.8888; -0.0686; -5.7935; 0.1379; 2.8009]

The vector specifies the relationship (positive or negative) of each variable with the result [but not causality].

X2: The more number of cylinders, the less miles a car can run per gallon. Since more cylinders might consume more gasoline at same time, it doesn't guarantee to run longer distance. [Negative correlation]

X3: Displacement and miles per gallon are in positive correlation.

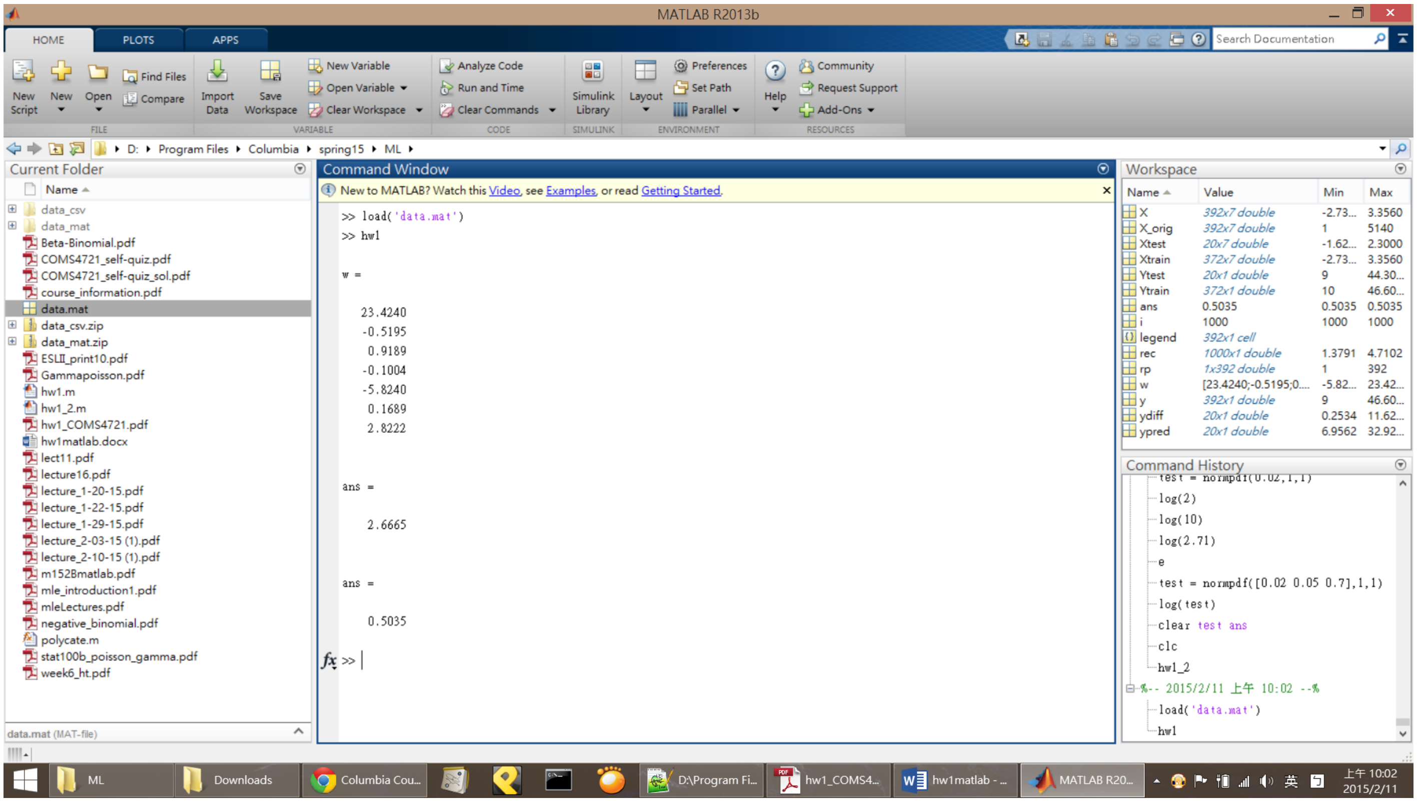
X4: Horsepower and miles per gallon are in negative correlation.

X5: Weight – the heavier a car is, the less miles per gallon a car can run; this is an obvious observation and really close to our intuition. [Negative correlation]

X6: Acceleration – the higher the acceleration value is, the more miles a car can run with per gallon gasoline. (A car with higher acceleration value can run faster thus longer distance with same amount of gasoline.)

X7: Model year – the earlier a car been produced, the smaller the value of this dimension, thus the less miles a car can run with a gallon gasoline. The cars produced nowadays are more gasoline-saving. [Positive correlation]

(b)



Current Folder

- data_csv
- data_mat
- Beta-Binomial.pdf
- COMS4721_self-quiz.pdf
- COMS4721_self-quiz_sol.pdf
- course_information.pdf
- data.mat
- data_csv.zip
- data_mat.zip
- ESLII_print10.pdf
- Gammapoisson.pdf
- hw1.m
- hw1_2.m
- hw1_COMS4721.pdf
- hw1matlab.docx
- lect11.pdf
- lecture16.pdf
- lecture_1-20-15.pdf
- lecture_1-22-15.pdf
- lecture_1-29-15.pdf
- lecture_2-03-15 (1).pdf
- lecture_2-10-15 (1).pdf
- m152Bmatlab.pdf
- mle_introduction1.pdf
- mleLectures.pdf
- negative_binomial.pdf
- polycate.m
- stat100b_poisson_gamma.pdf
- week6_ht.pdf

data.mat (MAT-file)

Command Window

New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> load('data.mat')
>> hw1

w =

   23.4240
   -0.5195
    0.9189
   -0.1004
   -5.8240
    0.1689
    2.8222


ans =

    2.6665


ans =

    0.5035

fx >> 
```

Workspace

Name	Value	Min	Max
X	392x7 double	-2.73...	3.3560
X_orig	392x7 double	1	5140
Xtest	20x7 double	-1.62...	2.3000
Xtrain	372x7 double	-2.73...	3.3560
Ytest	20x1 double	9	44.30...
Ytrain	372x1 double	10	46.60...
ans	0.5035	0.5035	0.5035
i	1000	1000	1000
legend	392x1 cell		
rec	1000x1 double	1.3791	4.7102
rp	1x392 double	1	392
w	[23.4240;-0.5195;0....	-5.82...	23.42...
y	392x1 double	9	46.60...
ydiff	20x1 double	0.2534	11.62...
ypred	20x1 double	6.9562	32.92...

Command History

```
test = normpdf(0.02,1,1)
log(2)
log(10)
log(2.71)
e
test = normpdf([0.02 0.05 0.7],1,1)
log(test)
clear test ans
clc
hw1_2
%-- 2015/2/11 上午 10:02 --%
load('data.mat')
hw1
```

Mean: 2.6665   Standard Deviation: 0.5035

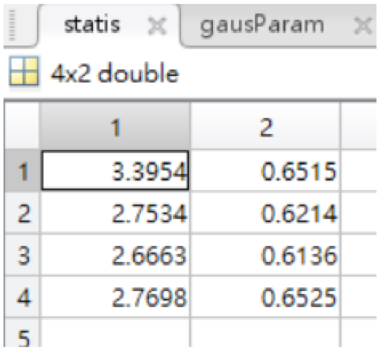
Part 2.

Matlab Code:

```
%% ML hw 1 - 3.2 %%

rec = zeros(20,1000,4);

for p = 1:4
        for i = 1:1000
                xnew = polycate(X,p);

                % split to train / test
                rp = randperm(392);
                Xtrain = xnew(rp(1:372),:);
                Xtest = xnew(rp(373:end),:);
                Ytrain = y(rp(1:372),:);
                Ytest = y(rp(373:end),:);

                % Analytic form of solving linear regression
                w = inv(Xtrain'*Xtrain)*Xtrain'*Ytrain;

                ypred = Xtest*w;
                rec(:,i,p) = Ytest - ypred;
        end
end

sqr = rec.*rec;
mn = sum(sqr,1)/20;
rt = sqrt(mn);
RMSE = squeeze(rt);

statis = zeros(4,2);  % record mean, std of RMSE
gausParam = zeros(4,2); % record mean and var of original error
loglike = zeros(4,1);  % record log likelihood

for p = 1:4
        statis(p,1) = mean(RMSE(:,p));
        statis(p,2) = std(RMSE(:,p));
        tmp = reshape(rec(:,:,p),1,20000);
        subplot(2,2,p), hist(tmp,100), title(['p=' num2str(p)]);
        mu = mean(tmp); sigma = std(tmp);
        gausParam(p,1) = mu; gausParam(p,2) = sigma;
        prob = normpdf(tmp, mu, sigma); % calculate the probability of generating data
         under given Gaussian distribution
        loglike(p) = sum(log(prob)); % sum them up to get log-likelihood
end
```

```
function X = polycate(a,b)
        if (b==1)
                X=a;
        end

        tmp = a(:,2:end);
        for i = 2:b
                tmp = tmp.*tmp;
                a = [a, tmp];
        end
        X=a;
end
```

(a) The result is the best when p=3 (slightly better than the result of p=2).

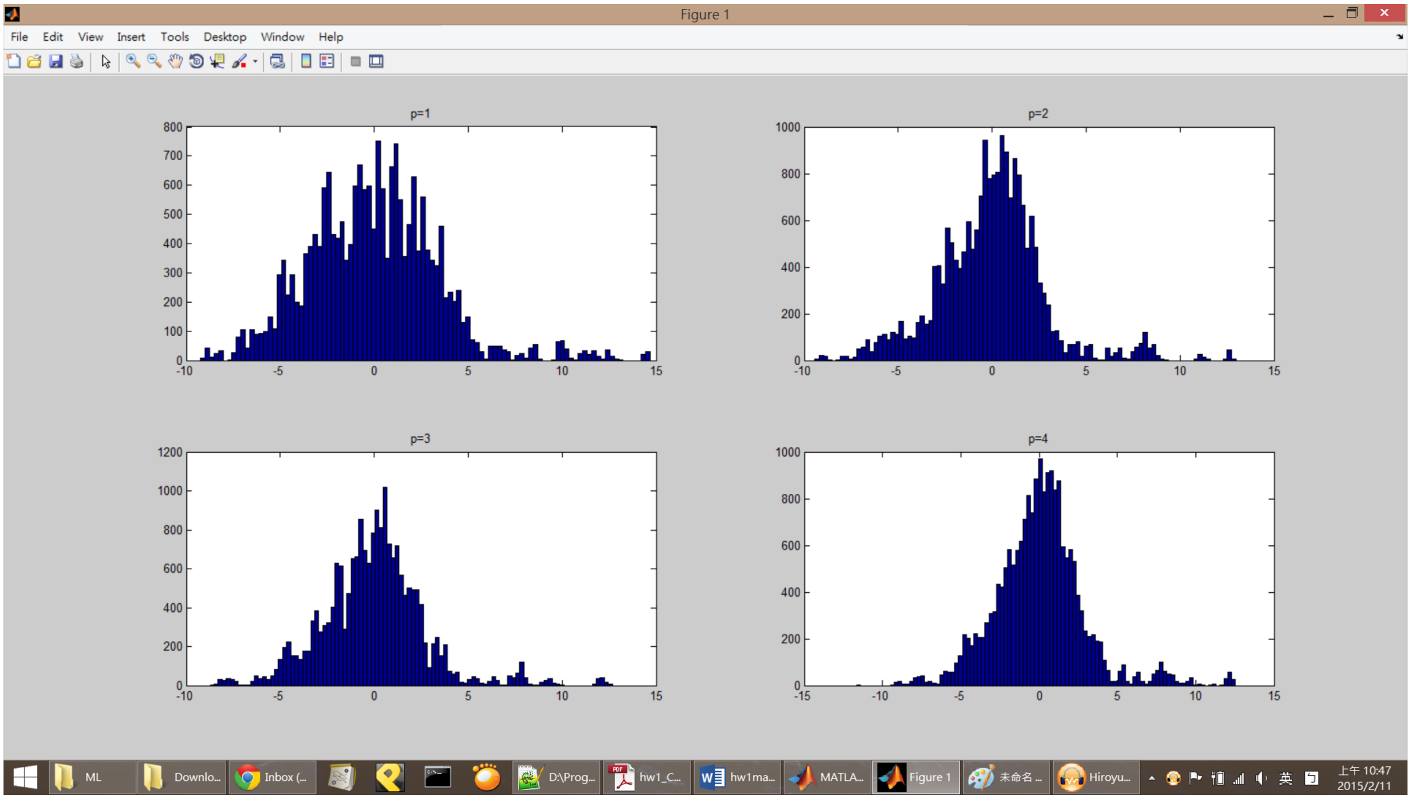

statis
gausParam
4x2 double

	1	2
1	3.3954	0.6515
2	2.7534	0.6214
3	2.6663	0.6136
4	2.7698	0.6525
5		

P	Mean	Std
1	3.3954	0.6515
2	2.7534	0.6214
3	2.6663	0.6136
4.	2.7698	0.6525

With the result, we can observe that the distribution of p=3 is more concentrating and thus better in four different p values.

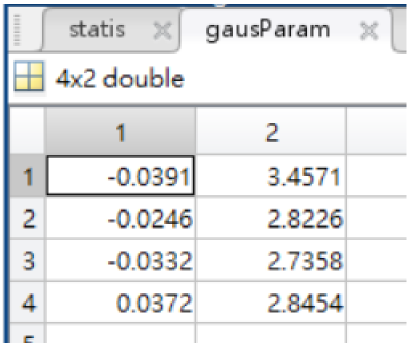
(b) Histogram:



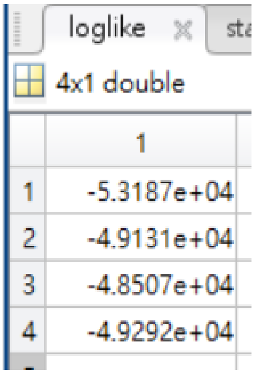
(c) Log-likelihood

By the derivation of univariate Gaussian in the course slide, the two parameter of Gaussian distribution can be calculated by maximum likelihood estimation on a set of observations. While it’s univariate:

$$\mu_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i \ \ (mean), \qquad \sigma_{ML} = std(X) \ \ (standard\ deviation)$$



	1	2
1	-0.0391	3.4571
2	-0.0246	2.8226
3	-0.0332	2.7358
4	0.0372	2.8454



	1
1	-5.3187e+04
2	-4.9131e+04
3	-4.8507e+04
4	-4.9292e+04

P	mean	std	Log-likelihood
1	-0.0391	3.5471	-5.3187e+04
2	-0.0246	2.8226	-4.9131e+04
3	-0.0332	2.7358	-4.8507e+04
4	0.0372	2.8454	-4.9292e+04

With the results show above, we can conclude that when p=3, which is 3rd order regression is the best model in four to fit the testing data. The reason is as below:

(1) With the larger the size of data, we can assume the distribution of predictions would be closer to Gaussian distribution. Therefore, we are going to find parameters of Gaussian distribution based on observations, and to see if the calculated Gaussian distribution fits the original data well. If not, we can assume that the predictions are not good enough, or even influenced by unexpected noise.

(2) To see if the curve fits the data well, we calculate the likelihood of generating same data as observations with given Gaussian parameters. The larger the likelihood (also the larger the log-likelihood), the closer between observations and the distribution curve, thus we consider it as better model.

$$\mathrm{y} = \arg\max_{p} \prod_{i=1}^{N} p(x_i|P = p, \mu, \sigma)\,, \mathrm{p} = 1, 2, 3, 4$$

$$\Rightarrow \mathrm{y} = \arg\max_{p} \ln \prod_{i=1}^{N} p(x_i|P = p, \mu, \sigma)\,, \mathrm{p} = 1, 2, 3, 4$$

$$\Rightarrow \mathrm{y} = \arg\max_{p} \sum_{i=1}^{N} \ln p(x_i|P = p, \mu, \sigma)\,, \mathrm{p} = 1, 2, 3, 4$$

Therefore, as shown in above table, we see that log-likelihood is maximize when p=3, we claim that p=3 provides the best polynomial regression model in the four.