Homework #1

Machine Learning, Fall 2016

Prof. J.T Chien

劉 淑雯, Rachel Lau

0540083, CS Dept., NCTU

Oct. 2016

---

1 Methodology

1.1 Bayesian Linear Regression

For a given input value $\boldsymbol{x}$, the corresponding target value $\boldsymbol{t}$ is assumed as a Gaussian distribution with a mean equal to the $\boldsymbol{y(x, \mathbf{w})}$. Thus we have

$$p(t|x, \mathbf{w}, \beta) = Normal(t|y(x, \mathbf{w}), \beta^{-1})$$

As the prior distribution is expressed as:

$$p(\mathbf{w}|\alpha) = Normal(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = (\frac{\alpha}{2\pi})^{(M+1)/2} \exp\{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\}$$

A predictive linear regression function is expressed by

$$y(x|\mathbf{w}) = \mathbf{w}^T\phi(x) = \phi^T(x)\mathbf{w}$$

Using Bayes' theorem, the posterior distribution for $\mathbf{w}$ is proportional to the product of the prior distribution and the likelihood function

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

A Bayesian treatment simply corresponds to a consistent application of the sum and product rules of probability, which allow the predictive distribution to be written in the form.

$$p(t|x, \mathbf{x}, \mathbf{t}) = \int p(t|x, \mathbf{w})p(\mathbf{w}|\mathbf{x},\mathbf{t})d\mathbf{w}$$

Now, we assume $\mathbf{w} = x, \mathbf{t} = y$, then use the equations on Page 93.

$$p(x) = p(\mathbf{w}|\alpha) = N(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = N(\mathbf{w}|\mu, \Lambda^{-1})$$
$$\Rightarrow \mu = 0, \Lambda = \alpha I$$
$$p(\mathbf{t}|,\mathbf{x},\mathbf{w}) = p(y|x) = \prod N(t_n|w^T\phi(x_n, \beta^{-1})$$
$$\Rightarrow A = \sum \phi(x_n), b = 0, L = \beta I$$

The multiplication of two gaussian functions is another gaussian function. Thus

$$p(\mathbf{w}|x, \mathbf{t}) = p(x|y) = N(x|\sum\{A^T L(y-b) + A\mu\}, \sum)$$
$$where \sum = (\alpha I + \beta \sum \phi(x_n)\phi(x)^T)^{-1} A^T L(y-b) = \sum \phi(x_n)\beta t_n$$

Assume $\sum = S$, thus

$$p(\mathbf{w}|, x, \mathbf{t}) = N(\mathbf{w}|\beta S^{-1} \sum \phi(x_n)t_n, S)$$

This time, we still assume $\mathbf{w} = x, t = y$.

$$p(x) = p(\mathbf{w}|\mathbf{x},\mathbf{t})$$
$$p(y|x) = p(t|x, \mathbf{w})$$

So use the quations on Page 93.

$$\mu = \beta S^{-1} \sum \phi(x_n)t_n$$
$$S = \Lambda^{-1}$$
$$A = \phi(x)^T, b = 0$$
$$L^{-1} = \beta^{-1} I$$

We only take the coefficients on the exponential into consideration for that the multiplication of two gaussian functions is another gaussian function.

$$p(t|x, \mathbf{x},\mathbf{t}) = p(y) = N(t|A\mu + b, L^{-1} + A\Lambda^{-1}A^T)$$
$$where\ A\mu + b = \beta\phi(x)^T S \sum \phi(x_n)t_n$$
$$L^{-1} + A\Lambda^{-1}A = \beta^{-1} + \phi(x)^T S\phi(x)$$

Assume

$$m(x) = \beta\phi(x)^T S \sum \phi(x_n)t_n$$
$$s^2(x) = \beta^{-1} + \phi(x)^T S\phi(x)$$

Thus

$$p(t|x, \mathbf{x},\mathbf{t}) = N(t|m(x), s^2(x))$$

1.2 Maximum Entropy

Entropy is average of information needed to specify the state of a random variable.

We know the states x_i of a discrete random variable X, where $p(X = x_i) = p_i$. The entropy of the random variable X is then

$$H[p] = -\sum p(x_i) \ln p(x_i)$$

The maximum entropy configuration can be found by maximizing H using a Lagrange multiplier to enforce the normalization constraint on the probabilities. Thus we maximize

$$H = -\sum p(x_i)\ln p(x_i) + \lambda(\sum p(x_i) - 1)$$

If $\boldsymbol{x}$ is continuous random variable, the differential entropy is given by

$$H[x] = -\int p(\mathbf{x})\ln p(\mathbf{x})d\mathbf{x}$$

This theorem is proved with the calculus of variations and Lagrange multipliers. We therefore maximize the differential entropy with the three constraints

$$\int_{-\infty}^{\infty} p(x)dx = 1$$
$$\int_{-\infty}^{\infty} xp(x)dx = \mu$$
$$\int_{-\infty}^{\infty} (x-\mu)^2 p(x)dx = \sigma^2.$$

The constrained maximazation can be performed using Lagrange multipliers so that we maximize the following functional with respect to $\boldsymbol{p(x)}$

$$J(p(x)) = -\int_{-\infty}^{\infty} p(x)\ln p(x)dx + \lambda_1(\int_{-\infty}^{\infty} p(x)dx - 1) + \lambda_2(\int_{-\infty}^{\infty} xp(x)dx - \mu) + \lambda_3(\int_{-\infty}^{\infty} (x-\mu)^2 p(x)dx - \sigma^2)$$

The entropy sttains an extremum when the functional derivative is equal to zero:

$$\frac{\delta J(p(x))}{\delta p(x)} = -\ln p(x) - 1 + \lambda_1 + \lambda_2 x + \lambda_3(x-\mu)^2 = 0$$

The maximum entropy probability distribution in this case must be of the form

$$p(x) = \exp\{-1 + \lambda_1 + \lambda_2 x + \lambda_3(x-\mu)^2\}$$

Then put $\boldsymbol{p(x)}$ into the three constraints.

$$\int_{-\infty}^{\infty} \exp\{-1 + \lambda_1 + \lambda_2 x + \lambda_3(x-\mu)^2\}dx = 1$$
$$\int_{-\infty}^{\infty} x\exp\{-1 + \lambda_1 + \lambda_2 x + \lambda_3(x-\mu)^2\}dx = \mu$$
$$\int_{-\infty}^{\infty} (x-\mu)^2\exp\{-1 + \lambda_1 + \lambda_2 x + \lambda_3(x-\mu)^2\}dx = \sigma^2.$$

We can calculate the λ_1, λ_2 and λ_3.

$$\lambda_1 = 1 - \frac{1}{2}\ln(2\pi\sigma^2)$$
$$\lambda_2 = 0$$
$$\lambda_3 = -\frac{1}{2\sigma^2}$$

The Lagrange multipliers will be found by , leading finally to the result

$$p(x) = \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$$

1.3 Application for Polynomial Regression

In this exercise, we need to write a regression program for the net hourly electrical energy output estimation by minimizing the error function

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n\}^2$$

The data set contains 500 instances. Each instance has 5 attributes and the `EP` attribute is the target.

1.3.1 Question 1

Two general polynomials with coefficients up to order 2 and order 3 are formed by

$$y(\mathbf{x},\mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j$$

where $\mathbf{x} = [x_1, \ldots, x_D]^T$ and $\mathbf{w} = \{w_0, w_i, w_{ij}\}$.

$$y(\mathbf{x},\mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

where $\mathbf{x} = [x_1, \ldots, x_D]^T$ and $\mathbf{w} = \{w_0, w_i, w_{ij}, w_{ijk}\}$.

The parameter w_0 allows for any fixed offset in the data and is sometimes called a *bias* parameter.

First, read the dataset and split fetures $\mathbf{x} = [x_1, \ldots, x_N]$ and target $\mathbf{y} = [y_1, \ldots, y_N]$.

```
[data,~,raw]=xlsread('...\data.xlsx');
x = data(:,1:4);
y = data(:,5);
```

Calculate the $N * M$ design matrix Φ of order 2 and 3, whose elements are given by $\Phi_{nj} = \phi((x)_n)$, so that

$$\Phi = \begin{Bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \cdots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{Bmatrix}$$

```
function [phi] = calculatePhi(data)
siz = size(data);
% bias parameter
phi(1:siz(1),1) = 1;
% 1d
phi(1:siz(1),2:siz(2)+1) = data(1:siz(1),1:siz(2));
% 2d
for i=1:siz(1)
   v = phi(i,2:siz(2)+1);
   d2 = v.*v';
   B = reshape(d2,[1,siz(2)*siz(2)]);
   phi(i,siz(2)+2:siz(2)+1+siz(2).^2)=B;
end
% 3d
for i=1:siz(1)
    A = phi(i,siz(2)+2:siz(2)+1+siz(2).^2)'*phi(i,2:siz(2)+1);
    A = reshape(A, [1,siz(2).^3]);
    phi(i,siz(2)+1+siz(2).^2+1: siz(2)+1+siz(2).^2 + siz(2).^3) = A;
end
```

We obtain

$$\mathbf{w}_{ML} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{t}$$

```
w = pinv(phi_train'*phi_train)*phi_train'*y_train;
```

Then do the predict on training set and test set.

$$\mathbf{y}_{predict} = \Phi\mathbf{w}_{ML}$$

```
y_train_pre = phi_train * w;
y_test_pre = phi_test * w;
```

Evaluate the corresponding root-mean-square RMS error on the training and tet set.

$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^{N}(\mathbf{y}_{predict} - \mathbf{y})^2}{N}}$$

```
E_train = sqrt(sum((y_train_pre - y_train).^2)/400);
E_test = sqrt(sum((y_test_pre - y_test).^2)/100);
```

The result is shown as below

Oder	Training Error	Test Error
2	3.8952	4.1974
3	3.8796	4.1991

1.3.2 Question 2

To select the most contributive attribute, we take the method that remove each attribute once and take the remaining three attributes to train the model and we will know which attribute's absense will make the E_{RMS} largest.

Similar to the above calculation process, we got the result.

Removed Attribute	Training Error	Test Error
T	6.8926	7.2934
V	4.2114	4.6055
AP	4.1052	4.6566
RH	3.9865	4.3508

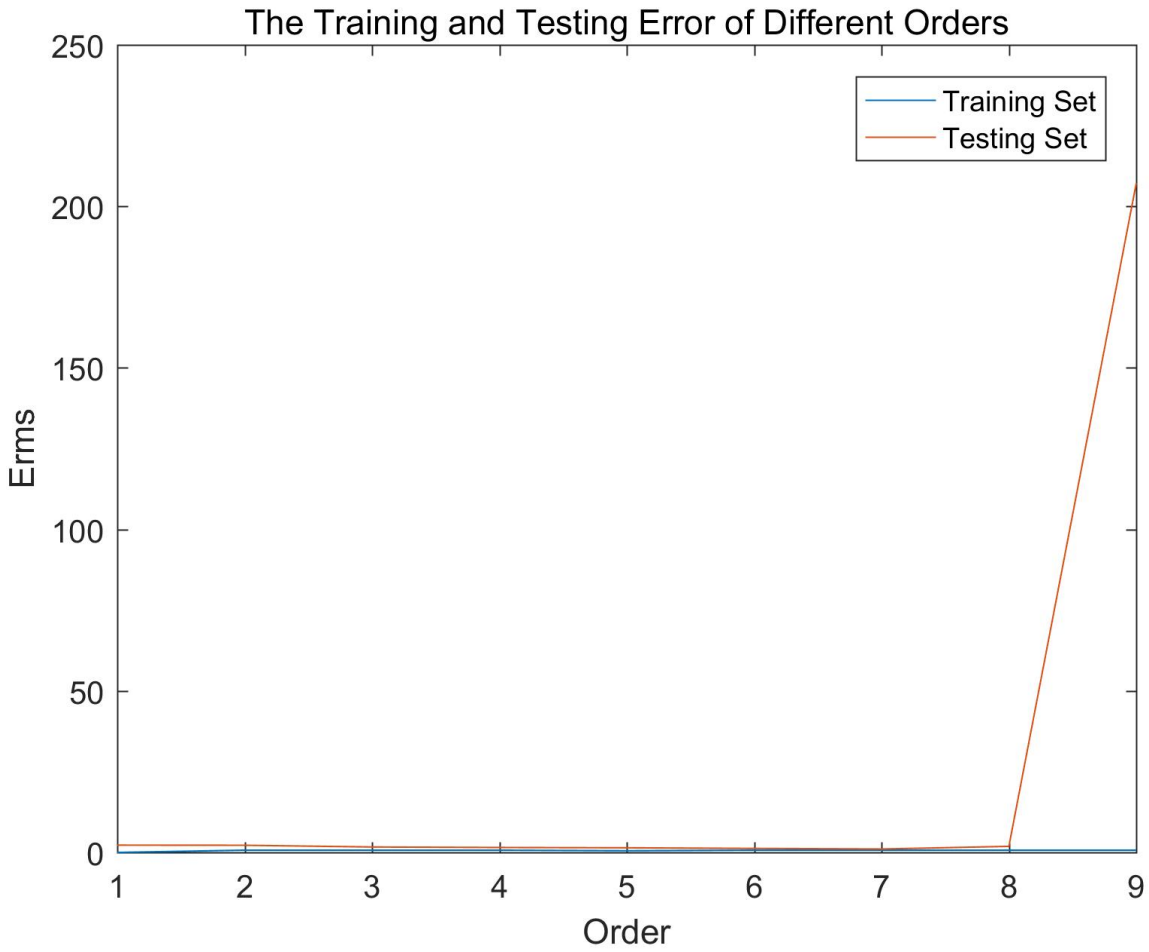
Obviously, *T* is the most contributive attribute.

1.4 Cross Validation

1.4.1 Question 1

To do cross validation, I equally divide the training set into 3 validation sets which means partitioning the training set into complementary subsets, performing the analysis on one subset which called the training set, and validating the analysis on the other subset which called the validation set or testing set.

The figure of training and testing error is shown as below.



Order	Training Error 1	Test Error 1	Training Error 2	Test Error 2	Training Error 3	Test Error 3
1	0.695	1.635	1.195	0.852	1.142	1.134
2	0.695	1.641	1.167	0.883	1.014	1.830
3	0.519	1.371	0.846	0.635	0.819	1.019
4	0.514	0.982	0.627	0.765	0.618	1.158
5	0.514	1.079	0.476	0.718	0.612	0.692
6	0.401	6.600	0.411	0.855	0.478	23.248

7	0.138	49.721	0.386	0.993	0.373	94.588
8	0.110	216.339	0.370	0.884	0.076	3836.677
9	0.110	178.061	0.053	109.260	0.033	6554.821

As we can see from the graphics, it causes a overfitting problem when the orders are 7, 8 and 9. To give an intuitive explanation of overfitting, overfitting means that we have a great result in our training set but poor results in our test set. Overfitting occurs when the capability of the model to adapt to different configurations in our data is too high. When the order increases, our learning algorithm will find the polunomial that better adjusts to the training points and return it. The more degrees we have the more the polynomial can adjust to the training set. The problem is that the algorithm is bad at generalization. That's why the testing error boosts after the order increases to 7.

Getting more data is always a way to mitigate overfitting. However, when more data doesn't help, we have to restrict the model with regularization.

1.4.2 Question 2

I choose the order 9, and add a regularization term into the model. Reply again and below is the result.