Liner regression and classification

1 Data and objective

I used the iris data set. The data set has five attributes; sepal length, sepal_width, petal_length, petal_width and species. I tried both regression and classification. This regression is to predict petal length of iris setosa from sepal length of iris setosa. This classification is binary classifcation of iris setosa and iris versicolor by sepal length and petal length.

2 Method of regression

I will briefly explain the regression method I implemented and applied.

2.1 Liner basis function model

When y is predicted from the observation data $\boldsymbol{x}$, linear basis function model can be used;

$$y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$$

where $\boldsymbol{w} = (w_0, \ldots, w_{M-1})^T$, $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^T$, and $\phi_0(\boldsymbol{x}) = 1$.

The weight $\boldsymbol{w}$ is decided from training data $\boldsymbol{X} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_N})^T$ together with their corresponding values $\boldsymbol{t} = (t_1, \ldots, t_N)^T$;

$$\boldsymbol{w} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{t}$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_0(\boldsymbol{x_1}) & \phi_1(\boldsymbol{x_1}) & \cdots & \phi_{M-1}(\boldsymbol{x_1}) \\ \phi_0(\boldsymbol{x_2}) & \phi_1(\boldsymbol{x_2}) & \cdots & \phi_{M-1}(\boldsymbol{x_2}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\boldsymbol{x_N}) & \phi_1(\boldsymbol{x_N}) & \cdots & \phi_{M-1}(\boldsymbol{x_N}) \end{bmatrix}.$$

The basis function $\boldsymbol{\phi}$ is decided. I tried the three following functions.

· Polynomial (In this case, called linear regression.)

$$\phi_j(x) = x^j$$

· Gaussian basis function

$$\phi_j(x) = \exp\left[-\frac{(x-\mu_j)^2}{2s^2} \right]$$

· Sigmoid basis function

$$\phi_j(x) = \sigma(\frac{x - \mu_j}{s})$$
$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

where s, μ are decided appropriately, but it may be better to decide μ covering the range of x.

This method to decide the weight is derived from maximum likelihood estimation of data generation probability assuming the data follows gaussian distribution.

2.2 Liner basis function model with regularization

Further I tried regularized least squares method when ϕ is polynomial. This method prevents over-fitting. This method decides the weight by the following equation;

$$\boldsymbol{w} = (\lambda \boldsymbol{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{t}$$

where λ is decided appropriately.

2.3 Validation test

10% of the total data is test data, 90% of the rest is training data and 10 % of the rest is validation data. If the dimension of model is too high, model is over-fitting to data. If the dimension of model is too low, model don't predict data well. Validation test find the optimal dimension of model. How to split data is changed serveral times, and the dimension is choosed when the expected value of E_{RMS} is minimum;

$$E_{RMS} = \sqrt{\frac{\sum_{i=0}^{N} (y_i - t_i)^2}{N}}$$

where predict data $\boldsymbol{y} = (y_1, \ldots, y_N)^T$ and test data $\boldsymbol{t} = (t_1, \ldots, t_N)^T$.

3 Result of regression

When the dimension of model is 5, E_{RMS} is higher than value I expected. This cause may be that iris data is not suitable for regression. In fact, the associated task of iris data is classification (http://archive.ics.uci.edu/ml/datasets/Iris).

In regualrization liner regression, E_{RMS} of even 10 dimensional model is as low as them of other dimensions.

I decide that the dimension of model is 4 based on the policy of the method I mentioned above.

All algorithms seems to be able to predict test data well.

4 Method of classification

I will briefly explain the binary classification method I implemented and applied.

4.1 Least squares classification

When the obseravation data $\boldsymbol{x}$ is classified into classes C_1 and C_2, the discriminant function is

$$\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{W}^T\boldsymbol{x}$$

where

$$\boldsymbol{W} = \begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,M-1} \\ w_{1,0} & w_{1,1} & \cdots & w_{M-1,M-1} \end{bmatrix}^T.$$

If $y_1(x) > y_2(x)$, the obseravation data $\boldsymbol{x}$ is classified into C_1. If $y_1(x) < y_2(x)$, the obseravation data $\boldsymbol{x}$ is classified into C_2.

The weight w is decided from training data $\boldsymbol{X} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_N})^T$ ($\boldsymbol{x} = (1, x_1, \ldots, x_{M-1})^T$) together with their corresponding values $\boldsymbol{T} = (\boldsymbol{t_1}, \ldots, \boldsymbol{t_N})$ (If $\boldsymbol{x} \in C_1, \boldsymbol{t} = (1,0)^T$ and if $\boldsymbol{x} \in C_2, \boldsymbol{t} = (0,1)^T$);

$$\boldsymbol{W} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{T}.$$

This method to decide weight is derived from minimaization of square error between data and the discriminant function.

4.2 Perceptron algorithm

When the observation data $\boldsymbol{x}$ is classified into classes C_1 and C_2, the perceptron function is

$$y(\boldsymbol{x}) = f(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}))$$

where

$$f(a) = \begin{cases} +1 & a \geq 0(C_1) \\ -1 & a < 0(C_2) \end{cases}$$

If $y(\boldsymbol{x}) = +1$, the observation data $\boldsymbol{x}$ is classified into C_1. If $y(\boldsymbol{x}) = -1$, the observation data $\boldsymbol{x}$ is classified into C_2.

The weight w is decided from training data $\boldsymbol{x} = (x_1, \ldots, x_N)$ together with their corresponding values t (If $\boldsymbol{x} \in C_1, t = +1$ and if $\boldsymbol{x} \in C_2, t = -1$);

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} + \eta\boldsymbol{\phi}_n t_n.$$

where η is decided appropriately. If all points or most points are classified correctly, iteration is stopped.

This method to decide weight is derived from minimaization of

$$E_p(\boldsymbol{w}) = -\sum_{n \in M} \boldsymbol{w}^T\boldsymbol{\phi}_n t_n.$$

I tried perceptron algorithm in the case of $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}$.

4.3 Validation test

10% of the total data is test data, 90% of the rest is training data and 10 % of the rest is validation data. Validation test checks if the discriminant function can classify serveral data correctly. How to split data is changed serveral times, and the expected value of the accuracy rate is calculated.

5 Result of classification

The expected value of the accuracy rate in validation data is 1.0. Both algorithms can classify data correctly in test data.