Nonlinear Regression, Classification

EE219: Large Scale Data Mining

Professor Roychowdhury

Jan 23, 2017

Summary

- Review
  - Linear Regression, MSE,MSPE
  - Correlation coefficient
- Nonlinear regression
  - polynomial regression
  - Logistic regression
  - Generalized additive model, neural network
- Structural Regularization
- Classification

Review

In the previous lecture, we introduced a linear regression problem. Assume we have n observations, $(y_1, x_1), (y_2, x_2), ...(y_n, x_n)$, $X_i \in \mathrm{R}^d$.

- It can be expressed as $y_{d\times 1} = B_{n\times d}\theta_{d\times 1} + \epsilon_{n\times 1}$.
- Define cost function(MSE): $g(\theta) = \sum_{i=1}^{n} \epsilon_i^2$, optimal solution is $\hat{\theta} = argmin_\theta(\sum_{i=1}^{n} \epsilon_i^2) = argmin_\theta(\|y - B\theta\|_2^2)$
- Mean square prediction error: $\text{MSPE} = \frac{1}{m} \sum_{y_i, x_i \in \mathrm{testset}} (y_i - x_i^T\theta^*)^2$, m is the size of the test set.
- Pearson correlation coefficient

MSE: orthogonality

$$
\begin{aligned}
y &= B\theta + \epsilon, \theta^* = (B^T B)^{-1} B^T y \\
\hat{y} &= B(B^T B)^{-1} B^T y, \epsilon = y - \hat{y} = (I - B(B^T B)^{-1} B^T) y \\
\hat{y}^T \epsilon &= y^T B(B^T B)^{-1} B^T (I - B(B^T B)^{-1} B^T) y \\
&= y^T (B(B^T B)^{-1} B^T - B(B^T B)^{-1} B^T B(B^T B)^{-1} B^T) y \\
&= y^T (B(B^T B)^{-1} B^T - B(B^T B)^{-1} B^T) y \\
&= 0
\end{aligned}
$$

- $\hat{y}^T \epsilon = \hat{y}^T (y - \hat{y}^T) = 0$, this is called the orthogonality principle. The linear estimator $\hat{y}$ achieves minimum mean square error if and only if $E[(\hat{y} - y)^T x_i] = 0$, and $E[y - \hat{y}] = 0$.

MSE: orthogonality

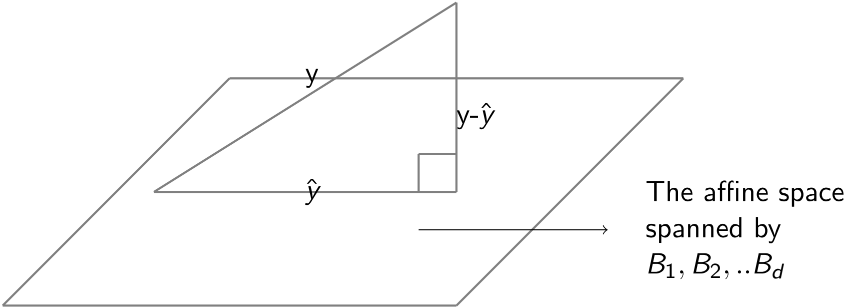


Figure 1: when d = 2

Actually, $\epsilon = y - \hat{y}$ is orthogonal to every x_i and lie in the nullspace of B^T. $B^T\epsilon = B^T(y - \hat{y}) = B^Ty - B^TB(B^TB)^{-1}B^T\hat{y} = 0$

MSE: offset inside X

As we mentioned before, the offset in the linear model can be absorbed into the variable x by adding one dimension(dummy variable).
Without generality, the model with a constant term can be expressed as $y_i = \sum_{j=2}^{d} \theta_j x_i(j) + \theta_1 + \epsilon$ ($x_i(1) = 1$ for all i = 1,2,3..n)

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1(2) & x_1(3) & \dots & x_1(d) \\ 1 & x_2(2) & x_2(3) & \dots & x_2(d) \\ \vdots & & & & \\ \vdots & & & & \\ 1 & x_n(2) & x_n(3) & \dots & x_n(d) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \vdots \\ \epsilon_n \end{bmatrix}$$

MSE: shift x with its mean

We first subtract the m_j from all $x_i(j)$ where $m_j = \frac{1}{n}\sum_{i=1}^{n} x_i(j)$, $j \neq 1$.Thus $x_i^{new}(j) = x_i(j) - m(j)$, so for every input its mean is 0. Then for the new $x_i = [1, x_i(2), ..x_i(d)]^T$, its covariance matrix $\frac{1}{n}B^T B$ has specific structure.

$$\frac{1}{n}B^T B = \frac{1}{n}\begin{bmatrix} | & \vdots & | \\ x_1 & \vdots & x_n \\ | & \vdots & | \end{bmatrix} \begin{bmatrix} \text{——} & x_1^T & \text{——} \\ \text{——} & x_2^T & \text{——} \\ \vdots & & \\ \vdots & & \\ \text{——} & x_n^T & \text{——} \end{bmatrix} = \frac{1}{n}\sum_{i=1}^{n}(x_i x_i^T)$$

- $\frac{1}{n}B^T B(1,1) = 1$
- $\frac{1}{n}B^T B(1,j) = \frac{1}{n}B^T B(j,1) = \frac{1}{n}\sum_{i=1}^{n} x_i(j) = 0$
- $\frac{1}{n}B^T B(i,j) = \frac{1}{n}\sum_{k=1}^{n} x_k(i) x_k(j) = cov(x(i), x(j))$

the covariance matrix

If we consider $x_1, x_2, ..xn$ to be samples of d-1 dimensional random variable $[x(2)x(3)..x(d)]^T$, thus

$$\frac{1}{n}B^TB = \frac{1}{n}\left[\begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \hline 0 & & & \\ \vdots & & R_{(d-1)\times(d-1)} & \\ 0 & & & \end{array}\right]$$

$R_{(d-1)\times(d-1)}$ is the sample covariance matrix of the non-constant input.

MSE: unbiased estimation

$\hat{y}$ is actually the unbiased estimation of y, which means $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i$

and $\bar{\epsilon} = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i = 0$

proof

We want to prove $1^T\epsilon = 0$. Then we can easily get $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i$.

$$1^T\epsilon = 1^T(y - \hat{y}) = (1^T - 1^T B(B^T B)^{-1} B^T)y$$

First let's look at $1^T B$

$$1^T B = [1\ 1..1]^T \begin{bmatrix} 1 & x_1(2) & ... & x_1(d) \\ 1 & x_2(2) & ... & x_2(d) \\ \vdots & & & \\ \vdots & & & \\ 1 & x_n(2) & ... & x_n(d) \end{bmatrix} = [1\ \sum_{i=1}^{n} x_i(2) ... \sum_{i=1}^{n} x_i(d)]^T$$

$$= [1\ 0\ 0..0]^T$$

MSE: unbiased estimation

$$(B^TB)^{-1} = \left[\begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \hline 0 & & & \\ \vdots & & R^{-1}_{(d-1)\times(d-1)} & \\ 0 & & & \end{array}\right] \text{,then } 1^TB(B^TB)^{-1} = [1\ 0\ 0..0]^T$$

finally,

$$[1\ 0\ 0..0]^TB^T = [1\ 0\ 0..0]^T \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1(2) & x_2(2) & \dots & x_n(2) \\ \vdots & & & \\ \vdots & & & \\ x_1(d) & x_2(d) & \dots & x_n(d) \end{bmatrix}$$

$$= 1^T$$

so $1^TB(B^TB)^{-1}B^T = 1^T, 1^T\epsilon = (1^T - 1^TB(B^TB)^{-1}B^T)y = 0$ (1^T is the first row of B)

MSE: Pearson correlation coefficient

Variance of y: $\sigma_y = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$ Let's look at $\sum_{i=1}^{n}(y_i - \bar{y})^2$.

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}[(y_i - \hat{y}_i) - (\bar{y} - \hat{y}_i)]^2$$
$$= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \sum_{i=1}^{n}(\bar{y} - \hat{y}_i)^2 - 2\sum_{i=1}^{n}(y_i - \hat{y}_i)(\bar{y} - \hat{y}_i)$$

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)(\bar{y} - \hat{y}_i) = \bar{y}\sum_{i=1}^{n}(y_i - \hat{y}_i) - \sum_{i=1}^{n}\hat{y}_i(y_i - \hat{y}_i)$$

- $\sum_{i=1}^{n}(y_i - \hat{y}_i) = 1^T\epsilon = 0$
- $\sum_{i=1}^{n}\hat{y}_i(y_i - \hat{y}_i) = \hat{y}^T(y - \hat{y}) = 0$

MSE: Pearson correlation coefficient

So $\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \sum_{i=1}^{n}(\bar{y} - \hat{y}_i)^2$

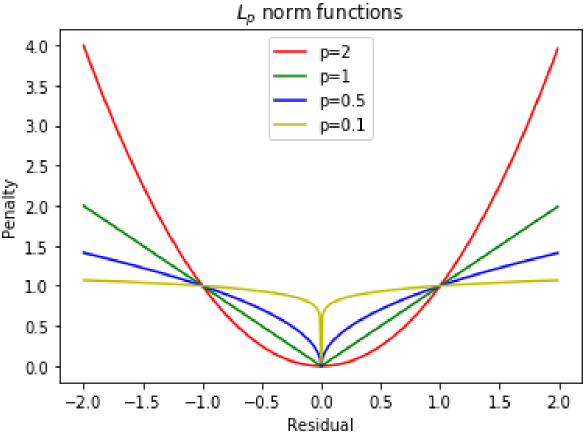
or $1 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} + \frac{\sum_{i=1}^{n}(\bar{y} - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$

- $R^2 = \frac{\sum_{i=1}^{n}(\bar{y} - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$ is the fraction of variance explained by $\hat{y}$
- $0 \leq R^2 \leq 1$

MSE: conclusion

- The linear estimator $\hat{y}$ achieves minimum least square error if and only if $E[(\hat{y}-y)^T x_i] = 0$, and $E[y - \hat{y}] = 0$, which means the estimator is the projection into the space spanned by $x_1, ..x_n$ and is unbiased.
- The linear model with constant term and d-1 dimension input can be transformed into d dimension input.
- Pearson correlation coefficient(R) $0 \leq R^2 \leq 1$ can be derived from above properties.

L_P norm as cost function



L_P norm as cost function

- The least square (L_2) estimator is sensitive to outliers, since ϵ_i^2 increases with ϵ_i
- If $1 \leq |\epsilon_i|$ then L_1 norm penalizes less for outliers than L_2. Using L_1 loss, we solve the following:

$$\hat{\theta} = argmin_\theta \sum_{i=1}^{n} |y_i - \theta^T x_i|$$

- Generally L_p norms can be used:

$$\hat{\theta}_p = argmin_\theta \left\| Y - X^T\theta \right\|_p$$

- Both problems have global minimum and can be solved by numerical methods.

Bayesian rules review

- Assume we have sample space of S such that: $S = \bigcup_{i=1}^{n} A_i$ and $A_i \cap A_j = \emptyset$ for each $i \neq j$
- The probability of event B is

$$P(B) = \sum_{i=1}^{n} P(B \cap A_i) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

  where $P(A_i)$ is a priori probability and $P(A_i|B)$ is called a posteriori Probability of A_i
- Bayes' law applies here:

$$P(A_i|B) = \frac{P(A_i \cap B)}{P(B)} = \frac{P(B|A_i)P(A_i)}{\sum_{i=1}^{n} P(B|A_i)P(A_i)}$$

Bayesian rules review

- Assume we get new data B, the distribution we had was $P(A_i)$ and the updated distribution is $P(A_i|B)$
- In general model is $y = f_\theta(x) + \epsilon$ and P_θ is prior distribution of θ
- θ is picked from a zero mean independent Gaussian distribution.
- A posterior distribution given data D is:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

- $P(D|\theta)$ is the likelihood of the $D = \{(y_i, x_i) \text{ for } i = 1 \dots n\}$ for fixed value of θ
- Here the best θ with Maximum Likelihood estimator is achieved with:

$$\hat{\theta}_{MLE} = \underset{\theta}{\text{argmax}} P(D|\theta)$$

Bayesian framework for regression

- In regression the model is:

$$y_i = \theta^T x_i + \epsilon_i \text{ and } \epsilon_i = y_i - \theta^T x_i$$

- ϵ_i are assumed independent and zero mean Gaussian random variables with variance σ
- Under these assumption:

$$P(D|\theta) = P(\epsilon_1, \ldots, \epsilon_n|\theta_1, \ldots, \theta_d) \quad = \Pi_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\epsilon_i^2}{2\sigma^2}}$$

$$= \Pi_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - x_i^T\theta)^2}{2\sigma^2}}$$

- Taking logarithm from cost function:

$$lnP(D|\theta) = \sum_{i=1}^{n} [ln(\frac{1}{\sqrt{2\pi\sigma^2}}) - \frac{(y_i - x_i^T\theta)^2}{2\sigma^2}]$$

Bayesian framework for regression

- Therefore we will solve the following optimization:

$$\hat{\theta}_{MLE} = \max_{\theta} lnP(D|\theta) = \min_{\theta} -lnP(D|\theta) = \min_{\theta} \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - x_i^T\theta)^2$$

$$\hat{\theta}_{MLE} = \min_{\theta} \sum_{i=1}^{n} (y_i - x_i^T\theta)^2$$

- The above equation proves that Least squares estimate is same as MLE under Gaussian error model.

MAP: Maximum A Posteriori

- here we assume that $P(\epsilon_1, \ldots, \epsilon_n|\theta) = \Pi_{i=1}^{n} \frac{c}{\lambda} e^{-\lambda|\epsilon_i|}$
- Therefore,

$$\hat{\theta} = \min_{\theta} - lnP(\epsilon_1, \ldots, \epsilon_n|\theta) = \min_{\theta} \lambda + \sum_{i=1}^{n} |y_i - \theta^T x_i|$$

- In $P(\theta|D) = \frac{P(D|\theta|)P(\theta)}{P(D)}$, $P(D)$ is independent of θ therefore

$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

- In maximum a posteriori probability we solve the following problem:

$$\hat{\theta}_{MAP} = \underset{\theta}{\operatorname{argmax}}\ P(\theta|D)$$

MAP and regularization(s)

- Here we model the $P(\epsilon_i) \propto e^{-\frac{\epsilon_i^2}{2\sigma^2}}$ and $P(\theta) \propto e^{-\frac{\theta_i^2}{2\lambda^2}}$
- As a result:

$$\hat{\theta}_{MAP} = \underset{\theta}{\operatorname{argmax}}\ lnP(D|\theta) + lnP(\theta)$$
$$= \underset{\theta}{\operatorname{argmin}}\ \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x_i^T\theta)^2 + \frac{1}{2\lambda^2}\sum_{i=1}^{n}\theta_i^2$$

- The above equation is Least squares loss function with L_2 regularization
- If we take the model of $P(\theta_i) = \frac{\lambda}{2}e^{-\lambda|\theta_i|}$ then the problem will be reduced to

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\ \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x_i^T\theta)^2 + \lambda\sum_{i=1}^{n}|\theta_i|$$

Which indicates least squares with L_1 regularization.

Nonliear regression

Polynomial regression

- $f(x_i) = b + \sum_{j=1}^{d} \theta_j x_i(j) + \sum_{j=1}^{d} \sum_{k=1}^{d} c_{jk} x_i(j) x_i(k),$
- $y_i = f(x_i) + \epsilon_i$
- number of coefficient $= \mathrm{d} + 1 + \binom{d}{2} \sim d^2$

Logistic regression

- $y_i = g(\theta^T x_i) + \epsilon_i$
- $g(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{e^z+1}$
- $0 \leq y_i \leq 1$, pick better function for arbitrary y_i?

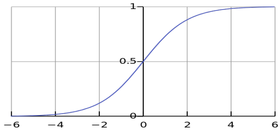
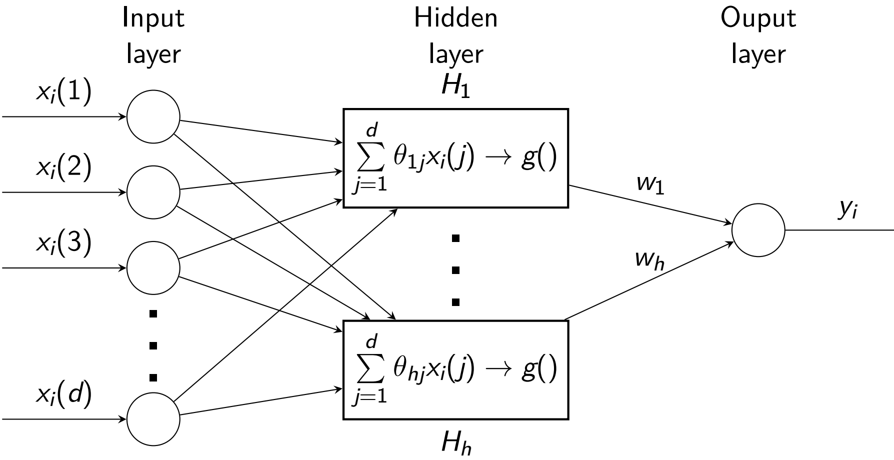


Figure 2: Logistic curve

Generalized additive models

$$y_i = \sum_{k=1}^{h} w_k f_k(x_i) + w_i$$

$$f_k(x_i) = g(\sum_{j=1}^{d} \theta_{kj} x_i(j))$$



Neural network

Property

- Number of parameters: $h \cdot d + h$.
- One of the term in the feature vector is always 1, thus there is no the constant term.
- No matter what f* is, by increasing h, we can approximate any f* arbitrarily close.
- Check notes on SGD for training neural networks.

Problem

Suppose arbitrarily approximation happens, overfitting is a big problem: for large enough h, training error can be 0,but MSPE could be very large, which means high generalization error.
We can learn all the data and smooth out by the form.

Overfitting

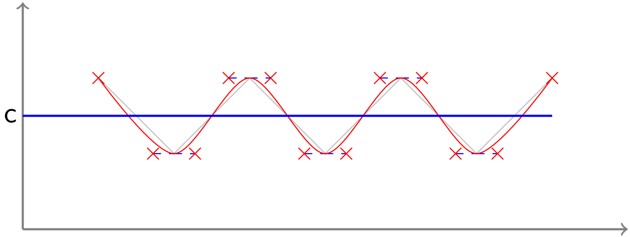


Figure 3: y = c+ ϵ

- For large enough h, training error $\rightarrow 0$
- Need regularization to address the overfitting problem,
$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^{n} w_k^2 + \mu \sum_{k=1}^{h} \sum_{j=1}^{d} \theta_{kj}^2$$

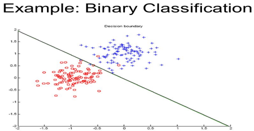
Structual regularization

- Select a set of θ_{kj} where $\theta_{kj} = 0$
- make the network sparse to start with, based on the domain knowledge
- for example, Convolutional Neural Network (CNN). Convolutional operator actually imposes a special structure into neural networks. In the image processing, to reduce the dimension, for one pixel, only its neighbor will come up with a structure, other pixels' parameters are set 0.

Classification problem

- classification is a special case of regression
- y_i is discretized
- Regression: when y is continuous or n is large, fitting the numbers spanning the y axis. Classification: output value is only one of n possible values.
- In the binary classification case, classifier is a surface defined by a function,making the points in class 1 all belong to one side. We try to find such function.

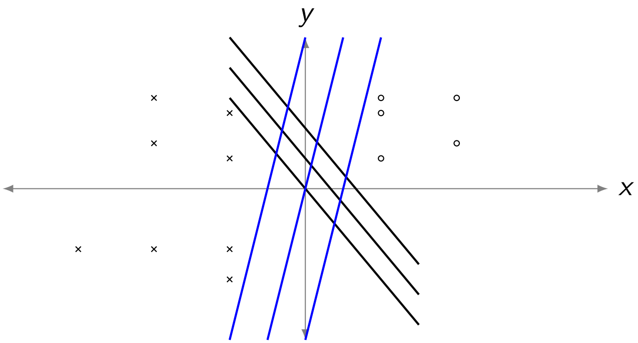
Binary classification



$$y_i = \begin{cases} 1 & \text{if } x_i \in C_1 \\ -1 & \text{if } x_i \in C_2 \end{cases}$$

- find surface function: $f(x_i) = w^T x_i - b$ is a hyperplane in n dimension, when $x_i \in \mathrm{R^n}$.
- Set constraints: if $x_i \in C_1$, then $w^T x_i - b \geq 1$;if $x_i \in C_2$, then $w^T x_i - b \leq -1$
- it can be reformulated as $y_i(w^T x_i - b) \geq 1$, i = 1,2..N
- such a w and b might not exist, which means the points are not linear separable, we need to add slack variable to allow error: $y_i(w^T x_i - b) \geq 1 - \epsilon_i (\epsilon_i \geq 0)$

SVM



- distance between $w^T x - b = 1$ and $w^T x - b = -1$ is $\frac{2}{w^T w}$
- maximize margin means minimize $\frac{1}{2} w^T w$
- when the slack variable is considered, the objective function to minimize will be $\frac{1}{2} w^T w + \lambda \sum_{i=1}^{n} \epsilon_i$