

Introduction to Regression

Greg Grudic

1 Definitions

1. $x \in \mathfrak{R}$ means that the variable x is a member of the real numbers.
2. $\mathbf{x} \in \mathfrak{R}^d$ means that the d elements of the vector $\mathbf{x} = (x_1, \dots, x_d)$ all are members of the real numbers.
3. If x is a random variable, then every sample of x is generated from some density function $D(x)$.
4. A random variable x is independently and identically distributed (iid) if each sample of x is independent from all other samples, and each is drawn from the same density function $D(x)$.
5. A distribution is stationary if it doesn't change over time.
6. Expected value of x is defined by:

$$\bar{x} = E[x] = \int_{-\infty}^{+\infty} xD(x) dx$$

An unbiased estimate of $E[x]$ can be obtained from N samples of x , denoted by x_1, \dots, x_N as follows

$$\hat{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

7. \hat{x} is an unbiased estimate of \bar{x} means that $E[\hat{x}] = \bar{x}$.
8. Variance of x is defined by:

$$\sigma^2 = V[x] = E[(x - \bar{x})^2]$$

The standard deviation is given by the square root of the above, or σ . An unbiased estimate of $V[x]$ can be obtained from N samples of x , once more symbolized by x_1, \dots, x_N as follows:

$$\hat{V}[x] = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

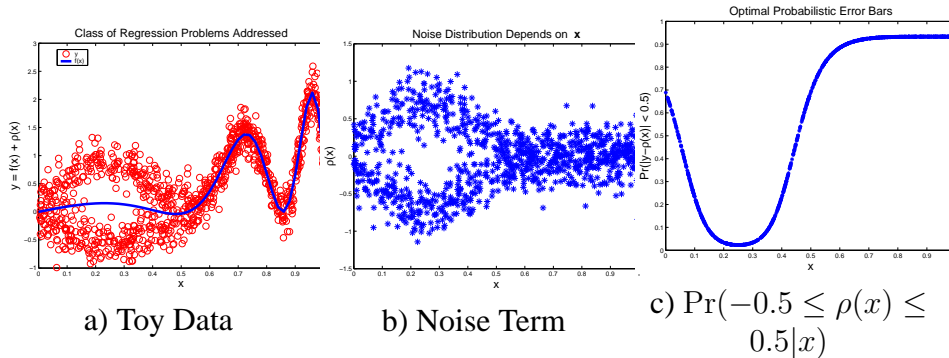


Figure 1: The noise term is a function of the input x and it is not Gaussian.

2 Assumptions on Data

Let $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ be a set of N training examples, where $\mathbf{x} \in \mathfrak{R}^d$ are independently and identically distributed (iid) from some stationary distribution $D(\mathbf{x})$. The standard regression function formulation assumes that the outputs $y \in \mathfrak{R}$ are generated from:

$$y = f(\mathbf{x}) + \rho$$

where $f(\mathbf{x}) \in \mathfrak{R}$ is a real valued function defined on $\mathbf{x} \in \mathfrak{R}^d$, and ρ is a random variable with mean zero (i.e. $E[\rho] = 0$) and finite variance (i.e. $V[\rho] = c, 0 \leq c < \infty, c \in \mathfrak{R}$). This regression formulation assumes that $E[\rho]$ and $V[\rho]$ are independent of \mathbf{x} .

A more general data model, which is generally NOT assumed, would allow ρ be a function of \mathbf{x} . This assumes that $y \in \mathfrak{R}$ is generated from:

$$y = f(\mathbf{x}) + \rho(\mathbf{x}) \tag{1}$$

where the random variable ρ has $E[\rho(\mathbf{x})] = 0$ and $V[\rho(\mathbf{x})] = c(\mathbf{x}), 0 \leq c(\mathbf{x}) < \infty, c(\mathbf{x}) \in \mathfrak{R}$. Figure 1a shows a regression function where the noise is not Gaussian, and it changes as a function of the input x . This noise term is shown in Figure 1b. The variation, as a function of x , in the probability that the noise term lies between -0.5 and 0.5 , is shown in Figure 1c. Note that the noise is constant for $x > 0.7$.

Training data inputs can be represented in matrix form. For N training example of d dimensional inputs:

$$\begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} = \begin{pmatrix} x_{11} & \dots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{N1} & \dots & x_{Nd} \end{pmatrix}$$

3 Regression Models

3.1 Linear Least Squares Regression

For data with d inputs, the model is assumed to have the following form:

$$\hat{y} = \hat{f}(\mathbf{x}) = \hat{\beta}_0 + \sum_{j=1}^d \hat{\beta}_j x_j = \hat{\beta}_0 + (\hat{\beta}_1, \dots, \hat{\beta}_d) \mathbf{x}^T$$

where $\beta = (\beta_0, \dots, \beta_d)^T$ are the model coefficients that are estimated from the training data. In *Least Squares Regression*, the model coefficients are chosen to minimize the following error metric (or Residual Sum of Squares (RSS)) on the training data $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$:

$$\hat{\beta} = \arg \min_{\beta} \left\{ \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^d \beta_j x_{ij} \right)^2 \right\}$$

A formula for $\hat{\beta}$ can be derived as follows:

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^N (y_i - \hat{f}(\mathbf{x}_i))^2 \\ &= \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^d \beta_j x_{ij} \right)^2 \\ &= (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) \end{aligned}$$

where $\mathbf{y} = (y_1, \dots, y_N)^T$ and

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1d} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & \cdots & x_{Nd} \end{pmatrix}$$

RSS is a quadratic function and can be solved by differentiating with respect to β , setting the resulting equation to zero (which is where the minimum is), and solving for β . Differentiating we get:

$$\frac{\partial RSS}{\partial \beta} = -2\mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta)$$

Assuming that \mathbf{X} is nonsingular (and hence $(\mathbf{X}^T \mathbf{X})^{-1}$ exists), setting to zero gives:

$$\mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta) = 0$$

Solving for β gives:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

The predicted model outputs for the training data are then given by:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

3.1.1 Parameter Inference

If we assume that the training data is generated from a function that has *exactly* the following form:

$$y = \beta_0 + \sum_{j=1}^d \beta_j x_j + \varepsilon$$

where ε is a Gaussian Noise term (i.e. Normally distributed) with $E[\varepsilon] = 0$ and finite variance σ^2 , then we can make a number of interesting conclusions about the parameters $\hat{\beta}$. Under these assumptions, $\hat{\beta}$ is generated from a $d + 1$ dimensional Normal Distribution, allowing Hypothesis testing on $\hat{\beta}$ (i.e. is $\hat{\beta}_j$ zero, and what is the probability that it falls in some range of values). For details see Chapter 3 of [1].

3.2 Regularization or Shrinkage Least Squares Methods

If any two inputs are well correlated, then $(\mathbf{X}^T \mathbf{X})^{-1}$ will not exist. You can calculate the correlation coefficient κ between any two variables x_i and x_j using:

$$\kappa = \frac{Cov(x_i, x_j)}{\sigma_i \sigma_j}$$

where σ_1 and σ_j are the standard deviation of x_i and x_j respectively, and the covariance $Cov(x_i, x_j)$ is given by:

$$Cov(x_i, x_j) = E[(x_i - \bar{x}_i)(x_j - \bar{x}_j)]$$

Note that $-1 \leq \kappa \leq 1$ and $(\mathbf{X}^T \mathbf{X})^{-1}$ will not exist if $|\kappa|$ is close to 1.

When $(\mathbf{X}^T \mathbf{X})^{-1}$ is singular, regularization methods can be used to find the model coefficients $\hat{\beta}$. Two commonly used regularization methods are Ridge regression and Lasso Regression.

3.2.1 Ridge Regression

In Ridge Regression, given training data $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, the ridge model coefficients $\hat{\beta}^{ridge}$ are defined as follows:

$$\hat{\beta}^{ridge} = \arg \min_{\beta} \left\{ \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^d \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^d \beta_j^2 \right\}$$

This is equivalent to the following optimization problem:

$$\begin{aligned} \hat{\beta}^{ridge} = \arg \min_{\beta} & \left\{ \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^d \beta_j x_{ij} \right)^2 \right\} \\ \text{subject to:} & \sum_{j=1}^d \beta_j^2 \leq s \\ & s > 0 \end{aligned}$$

Obtaining the ridge model coefficients $\hat{\beta}^{ridge}$ is relatively straight forward. One way is to center all your inputs by replacing x_{ij} with $x_{ij} - \bar{x}_j$. The coefficients $(\hat{\beta}_1^{ridge}, \dots, \hat{\beta}_d^{ridge})$ are given by

$$(\hat{\beta}_1^{ridge}, \dots, \hat{\beta}_d^{ridge})^T = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

where \mathbf{I} is the d by d identity matrix, and \mathbf{X} contains the centered data points as follows:

$$\mathbf{X} = \begin{pmatrix} x_{11} - \bar{x}_1 & \dots & x_{1d} - \bar{x}_d \\ \vdots & \ddots & \vdots \\ x_{N1} - \bar{x}_1 & \dots & x_{Nd} - \bar{x}_d \end{pmatrix}$$

Then the offset coefficient is obtained by the following equation:

$$\hat{\beta}_0^{ridge} = \bar{y} - (\hat{\beta}_1^{ridge}, \dots, \hat{\beta}_d^{ridge}) \bar{\mathbf{x}}^T = \left[\frac{1}{N} \sum_{i=1}^N y_i \right] - \left[\sum_{j=1}^d \bar{x}_j \hat{\beta}_j^{ridge} \right]$$

3.2.2 Lasso Regression

In Lasso Regression, given training data $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, the lasso model coefficients $\hat{\beta}^{lasso}$ are calculate as follows:

$$\hat{\beta}^{lasso} = \arg \min_{\beta} \left\{ \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^d \beta_j x_{ij} \right)^2 \right\}$$

subject to: $\sum_{j=1}^d |\beta_j| \leq s$

$s > 0$

Therefore, the main difference between Lasso and Ridge is that Lasso measures shrinkage by $\sum |\beta|$ while ridge uses $\sum \beta^2$. For Lasso, this has the interesting and desirable effect of setting β coefficients to zero. This not true for Ridge Regression. However, Ridge regression is easy to solve, while Lasso has has been computationally expensive until very recently (see <http://www-stat.stanford.edu/~hastie/Papers/LARS/>).

3.3 Non-Linear Least Squares Regression

3.3.1 Standard Nonlinear Formulations

When the problem is believed to be non-linear, the following model structure can be used:

$$\hat{y} = \hat{f}(\mathbf{x}) = \hat{\beta}_0 + \sum_{j=1}^p \hat{\beta}_j \varphi_j(\mathbf{x}) = \hat{\beta}_0 + (\hat{\beta}_1, \dots, \hat{\beta}_p) \Phi(\mathbf{x})^T$$

Where the p *basis functions* $\Phi(\mathbf{x}) = (\varphi_1(\mathbf{x}), \dots, \varphi_p(\mathbf{x}))^T$ are nonlinear functions of the inputs \mathbf{x} . Examples include $\varphi_1(\mathbf{x}) = x_1 x_2$, $\varphi_2(\mathbf{x}) = x_1^2$, $\varphi_3(\mathbf{x}) = \sin(x_3)$, etc.

The model coefficients $\beta = (\hat{\beta}_0, \dots, \hat{\beta}_p)^T$ can be obtained using the same techniques described above with the following differences. For standard least squares regression the \mathbf{X} matrix becomes:

$$\mathbf{X} = \begin{pmatrix} 1 & \varphi_1(\mathbf{x}_1) & \cdots & \varphi_p(\mathbf{x}_1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \varphi_1(\mathbf{x}_N) & \cdots & \varphi_p(\mathbf{x}_N) \end{pmatrix}$$

For ridge regression, the \mathbf{X} matrix becomes:

$$\mathbf{X} = \begin{pmatrix} \varphi_1(\mathbf{x}_1) - \bar{\varphi}_1 & \cdots & \varphi_p(\mathbf{x}_1) - \bar{\varphi}_p \\ \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{x}_N) - \bar{\varphi}_1 & \cdots & \varphi_p(\mathbf{x}_N) - \bar{\varphi}_p \end{pmatrix}$$

where

$$\bar{\varphi}_j = \frac{1}{N} \sum_{i=1}^N \varphi_j(\mathbf{x}_i)$$

3.3.2 Kernel Regression

The problem with the standard non-linear regression as posed above is that there are infinitely many possibilities for the nonlinear basis functions $\phi_j(\mathbf{x})$, and choosing which to use and how many can become computationally expensive. In kernel regression, the basis functions are restricted to be kernels of the following form:

$$\varphi_i(\mathbf{x}) = K(\mathbf{x}_i, \mathbf{x})$$

where \mathbf{x}_i for $i = 1, \dots, N$ are training example inputs. Therefore, the kernel model looks like:

$$\hat{y} = \hat{\beta}_0 + \sum_{i=1}^N \hat{\beta}_i K(\mathbf{x}_i, \mathbf{x}) \quad (2)$$

Typically used kernel functions include the Gaussian Kernel:

$$K(\mathbf{a}, \mathbf{b}) = \exp \left[-\frac{\|\mathbf{a} - \mathbf{b}\|^2}{2\sigma^2} \right]$$

where $\mathbf{a}, \mathbf{b} \in \mathfrak{R}^d$, the kernel parameter is $\sigma > 0$, and

$$\|\mathbf{a} - \mathbf{b}\|^2 = \sum_{j=1}^d (a_j - b_j)^2$$

The polynomial kernel:

$$K(\mathbf{a}, \mathbf{b}) = (q + \mathbf{a}^T \mathbf{b})^k$$

where $\mathbf{a}, \mathbf{b} \in \mathfrak{R}^d$, the kernel parameter are $k = 1, 2, 3, \dots$ and $q \in \mathfrak{R}$. The sigmoid Kernel:

$$K(\mathbf{a}, \mathbf{b}) = \tanh(\kappa \mathbf{a}^T \mathbf{b} + \theta)$$

where $\mathbf{a}, \mathbf{b} \in \mathfrak{R}^d$, the kernel parameter are $\kappa \in \mathfrak{R}$ and $\theta \in \mathfrak{R}$.

For kernel ridge regression, the \mathbf{X} matrix becomes:

$$\mathbf{X} = \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) - \bar{\varphi}_1 & \dots & K(\mathbf{x}_N, \mathbf{x}_1) - \bar{\varphi}_N \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_1, \mathbf{x}_N) - \bar{\varphi}_1 & \dots & K(\mathbf{x}_N, \mathbf{x}_N) - \bar{\varphi}_N \end{pmatrix} \quad (3)$$

where

$$\bar{\varphi}_j = \frac{1}{N} \sum_{i=1}^N K(\mathbf{x}_j, \mathbf{x}_i) \quad (4)$$

Then the offset coefficient is obtained by the following equation:

$$\hat{\beta}_0^{ridge} = \left[\frac{1}{N} \sum_{i=1}^N y_i \right] - \left[\sum_{j=1}^N \bar{\varphi}_j \hat{\beta}_j^{ridge} \right] \quad (5)$$

Ridge regression is often used to solve for the model coefficients, and this framework is called *Kernel Ridge Regression*. Kernel Ridge Regression is a powerful tool for building nonlinear regression models, often outperforming other techniques.

4 Practical Considerations

There are a number of practical considerations when building regression models. These include:

1. Scale all inputs to the same range. Usually -1 to $+1$ before building the model.
2. Get ride of highly correlated inputs.
3. Feature (i.e. input) selection. Features that do not improve the model should be discarded. Lasso regression is a good method for feature selection. We will cover others in class. See Chapter 3 of [1].

5 Other Regression Formulations

Other examples of regression algorithms include Gaussian Process Regression, Support Vector Regression, Regression Trees, and Nearest Neighbor Regression.

References

- [1] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: data mining, inference and prediction*. Springer, 2001.