

Statistical Methods for Data Science

Lesson 16 - Multiple, non-linear, and logistic regression.

Salvatore Ruggieri

Department of Computer Science
University of Pisa
salvatore.ruggieri@unipi.it

Simple linear regression model

SIMPLE LINEAR REGRESSION MODEL. In a *simple linear regression model* for a bivariate dataset $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, we assume that x_1, x_2, \dots, x_n are nonrandom and that y_1, y_2, \dots, y_n are realizations of random variables Y_1, Y_2, \dots, Y_n satisfying

$$Y_i = \alpha + \beta x_i + U_i \quad \text{for } i = 1, 2, \dots, n,$$

where U_1, \dots, U_n are *independent* random variables with $E[U_i] = 0$ and $\text{Var}(U_i) = \sigma^2$.

- *Regression line*: $y = \alpha + \beta x$ with *intercept* α and *slope* β
- Least Square Estimators: $\hat{\alpha}$ and $\hat{\beta}$ and $\hat{\sigma}^2$
- Unbiasedness: $E[\hat{\alpha}] = \alpha$ and $E[\hat{\beta}] = \beta$ and $E[\hat{\sigma}^2] = \sigma^2$
- Moreover: $\text{Var}(\hat{\alpha}) = \sigma^2(1/n + \bar{x}_n^2/SXX)$ and $\text{Var}(\hat{\beta}) = \sigma^2/SXX$
- *Standard errors* (estimates of $\sqrt{\text{Var}(\hat{\alpha})}$ and $\sqrt{\text{Var}(\hat{\beta})}$):

$$se(\hat{\alpha}) = \hat{\sigma} \sqrt{\left(\frac{1}{n} + \frac{\bar{x}_n^2}{SXX}\right)} \qquad se(\hat{\beta}) = \frac{\hat{\sigma}}{\sqrt{SXX}}$$

Standard error of fitted values (predictions)

- For a given x_0 , the estimator $\hat{Y} = \hat{\alpha} + \hat{\beta}x_0$ has expectation $E[\hat{Y}] = \alpha + \beta x_0$
- Hence, $\hat{y} = \alpha + \beta x_0$, is the best estimate for the fitted value
- Variance of \hat{Y} is:

[See notes2.pdf]

$$\text{Var}(\hat{Y}) = \sigma^2 \left(\frac{1}{n} + \frac{(\bar{x}_n - x_0)^2}{SXX} \right)$$

- The *standard error* of the fitted value is then the estimate:

$$\text{se}(\hat{Y}) = \hat{\sigma} \sqrt{\left(\frac{1}{n} + \frac{(\bar{x}_n - x_0)^2}{SXX} \right)}$$

where

$$SXX = \sum_1^n (x_i - \bar{x}_n)^2$$

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum_1^n (y_i - \hat{\alpha} - \hat{\beta}x_i)^2$$

See R script

Weighted Least Squares and simple polynomial regression

- Weighted Simple Regression

$$S(\alpha, \beta) = \sum_{i=1}^n (y_i - \alpha - \beta x_i)^2 w_i$$

- ▶ w_i is the weight (or importance) of observation (x_i, y_i)
 - ▶ For integer weights, it is the same as replicating instances
- Polynomial Simple Regression

$$S(\alpha, \beta) = \sum_{i=1}^n (y_i - \alpha - \beta_1 x_i - \beta_2 x_i^2 - \dots - \beta_k x_i^k)^2$$

- ▶ $Y_i = \alpha + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_k x_i^k + U_i$ for $i = 1, 2, \dots, n$

See R script

Non-linear regression and transformably linear functions

- Non-linear Simple Regression, for a generic function $f()$
- $Y_i = f(\alpha, \beta, x_i) + U_i$ for $i = 1, 2, \dots, n$

$$S(\alpha, \beta) = \sum_{i=1}^n (y_i - f(\alpha, \beta, x_i))^2$$

- $\min S(\alpha, \beta)$ maybe without a closed form
 - ▶ use numeric search of the minimum (which may fail to find!), e.g., gradient descent
- Some $f()$ can be favourably transformed, e.g., $f(\alpha, \beta, x_i) = \alpha x_i^\beta$ *[Linearization]*
- Solve $\log Y_i = \log \alpha + \beta \log x_i + U_i$ and then by exponentiation:

$$Y_i = \alpha x_i^\beta e^{U_i}$$

where the error term is a multiplicative factor (must be checked with residual analysis)

See R script

Multiple linear regression

- Multivariate dataset:

$$(x_1^1, x_1^2, \dots, x_1^k, y_1), \dots, (x_n^1, x_n^2, \dots, x_n^k, y_n)$$

- $Y_i = \alpha + \beta_1 x_i^1 + \dots + \beta_k x_i^k + U_i$

- In vector terms:

- ▶ $Y_i = \mathbf{x}_i \cdot \boldsymbol{\beta} + U_i$, where $\boldsymbol{\beta}^T = (\alpha, \beta_1, \dots, \beta_k)$ and $\mathbf{x}_i = (x_i^1, \dots, x_i^k)$

- ▶ $\mathbf{Y} = \mathbf{X} \cdot \boldsymbol{\beta} + \mathbf{U}$, where $\mathbf{Y} = (Y_1, \dots, Y_n)$, $\mathbf{U} = (U_1, \dots, U_n)$, and $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$

- Ordinary Least Square Estimation (OLS):

$$S(\boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \mathbf{x}_i \cdot \boldsymbol{\beta})^2 = \|\mathbf{y} - \mathbf{X} \cdot \boldsymbol{\beta}\|^2 \quad \hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} S(\boldsymbol{\beta}) = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y}$$

where $\mathbf{y} = (y_1, \dots, y_n)$ and $\|(v_1, \dots, v_n)\| = \sqrt{\sum_{i=1}^n v_i^2}$ is the Euclidian norm

- Meaning of β_i : change of Y due to a unit change in x_i all the x_j with $j \neq i$ unchanged!
- It is the best (ie., smallest MSE) linear unbiased estimator [**Gauss-Markov Thm.**]

See R script

Omitted variable bias

- $Y_i = \alpha + \beta x_i + U_i$
- Assume there exists a third (unknown) variable Z such that:
 - ▶ X and Z are correlated
 - ▶ Y is determined by Z
- $Y_i = \alpha + \beta_1 x_i + \beta_2 z_i + U'_i$ but we do not know z_i 's
- $E[U_i] = E[\beta_2 z_i + U'_i] = \beta_2 z_i + E[U'_i] = \beta_2 z_i \neq 0$
- The problem **cannot** be solved by increasing the number of observations!

See R script

Multi-collinearity and variance inflation factors

- *Multicollinearity*: two or more independent variables (regressors) are strongly correlated.
- $Y_i = \alpha + \beta_1 x_i^1 + \beta_2 x_i^2 + U_i$
- It can be shown that for $j \in \{1, 2\}$:

$$\text{Var}(\hat{\beta}_j) = \frac{1}{(1 - r^2)} \cdot \frac{\sigma^2}{SXX_j}$$

where $r = \text{cor}(x^1, x^2)$, $\sigma^2 = \text{Var}(U_i)$ and $SXX_j = \sum_1^n (x_i^j - \bar{x}_n)^2$

- Correlation between regressors increases the variance of the estimators
- In general, for more than 2 variables:

$$\text{Var}(\hat{\beta}_j) = \frac{1}{(1 - R_j^2)} \cdot \frac{\sigma^2}{SXX_j}$$

where R_j^2 is the coefficient of determination (R^2) in the regression of x_j from all other x_i 's.

- The term $1/(1-R_j^2)$ is called *variance inflation factor*

See R script

Variable selection

- Recall: when $U_i \sim N(0, \sigma^2)$, we have $Y_i \sim N(\mathbf{x}_i \cdot \boldsymbol{\beta}, \sigma^2)$, hence we can apply MLE
- Log-likelihood is $\ell(\boldsymbol{\beta}) = \sum_{i=1}^n \log \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{y_i - \mathbf{x}_i \cdot \boldsymbol{\beta}}{\sigma} \right)^2} \right)$
- Akaike information criterion (AIC), balances model fit against model simplicity

$$AIC(\boldsymbol{\beta}) = 2|\boldsymbol{\beta}| - 2\ell(\boldsymbol{\beta})$$

- `stepAIC(model, direction="backward")` algorithm
 1. $S = \{x^1, \dots, x^k\}$
 2. $b = AIC(S)$
 3. repeat
 - 3.1 $x = \operatorname{argmin}_{x \in S} AIC(S \setminus \{x\})$
 - 3.2 $v = AIC(S \setminus \{x\})$
 - 3.3 if $v < b$ then $S, b = S \setminus \{x\}, v$
 4. until no change in S
 5. return S

See R script

Regularization methods

$$\hat{\beta} = \operatorname{argmin}_{\beta} S(\beta)$$

- Ordinary Least Square Estimation (OLS):

$$S(\beta) = \|\mathbf{y} - \mathbf{X} \cdot \beta\|^2$$

where $\|(v_1, \dots, v_n)\| = \sqrt{\sum_{i=1}^n v_i^2}$ is the Euclidian norm

- Ridge regression:

$$S(\beta) = \|\mathbf{y} - \mathbf{X} \cdot \beta\|^2 + \lambda_2 \|\beta\|^2$$

where $\|\beta\|^2 = \alpha^2 + \sum_{i=1}^k \beta_i^2$.

- ▶ Notice that λ_2 is not in the parameters of the minimization problem!
- ▶ Variables with minor contribution have their coefficients **close** to zero
- ▶ It improves prediction error by reducing overfitting through a bias-variance trade-off
- ▶ It is **not** a parsimonious method, i.e., does not reduce features

Regularization methods

- Lasso (least absolute shrinkage and selection operator) regression:

$$S(\beta) = \|\mathbf{y} - \mathbf{X} \cdot \beta\|^2 + \lambda_1 \|\beta\|_1^2$$

where $\|\beta\|_1^2 = |\alpha| + \sum_{i=1}^k |\beta_i|$.

- ▶ Notice that λ_1 is not in the parameters of the minimization problem!
 - ▶ Variable with minor contribution have their coefficients **equal** to zero
 - ▶ It improves prediction error by reducing overfitting through a bias-variance trade-off
 - ▶ It **is** a parsimonious method, i.e., does reduce features
- Penalized linear regression:

$$S(\beta) = \|\mathbf{y} - \mathbf{X} \cdot \beta\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1^2$$

- ▶ Both Ridge and Lasso regularization parameters
- How to solve the minimization problems? **Lagrange multiplier method** or **reduction to Support Vector Machine** learning
 - How to find the best λ_1 and/or λ_2 ? Cross-validation!

See R script

Multivariate linear regression

- The multivariate linear model accommodates two or more dependent variables

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{U}$$

where

- ▶ \mathbf{Y} is $n \times m$: n observations, m dependent variables
 - ▶ \mathbf{X} is $n \times (k + 1)$: n observations, k dependent variables +1 constants
 - ▶ $\boldsymbol{\beta}$ is $(k + 1) \times m$: k parameters $\boldsymbol{\beta}$ +1 parameter α for each of the m dependent variables
 - ▶ \mathbf{U} is $n \times m$: n observations, m error terms
- It is **not** just a collection of m multiple linear regressions
 - Errors in rows (observations) of \mathbf{U} are independent, as in a single multiple linear regression
 - Errors in columns (dependent variables) are allowed to be correlated.
 - ▶ E.g., errors of plasma level and amitriptyline due to usage of drugs
 - ▶ Hence, coefficients from the models covary! More later on confidence intervals for coefficients

See R script

Towards logistic regression

- Consider a bivariate dataset

$$(x_1, y_1), \dots, (x_n, y_n)$$

where $y_i \in \{0, 1\}$, i.e., Y_i i binary variable

- Using directly use linear regression:

$$Y_i = \alpha + \beta x_i + U_i$$

results in poor performances (R^2)

See R script

Towards logistic regression

- Consider a bivariate dataset

$$(x_1, y_1), \dots, (x_n, y_n)$$

where $y_i \in \{0, 1\}$, i.e., Y_i i binary variable

- Group by x values:

$$(d_1, f_1), \dots, (d_m, f_m)$$

where d_1, \dots, d_m are the distinct values of x_1, \dots, x_n and f_i is the fraction of 1's:

$$f_i = \frac{|\{j \in [1, n] \mid x_j = d_i \wedge y_j = 1\}|}{|\{j \in [1, n] \mid x_j = d_i\}|}$$

and the linear model:

$$F_i = \alpha + \beta x_i + U_i$$

See R script

Towards logistic regression

- Rather than f_i , we model the logit of f_i

$$\text{logit}(F_i) = \alpha + \beta x_i + U_i$$

where logit and its inverse (logistic function) are:

$$\text{logit}(p) = \log \frac{p}{1-p} \quad \text{inv.logit}(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$$

See R script

Logistic regression and generalized linear models

- Since Y_i 's are binary, $F_i = P(Y_i = 1|X = x_i) \sim \text{Ber}(f_i)$, and U_i is not necessary

$$\text{logit}(F_i) = \alpha + \beta x_i$$

and then $F_i = P(Y_i = 1|X = x_i) = \text{inv.logit}(\alpha + \beta x_i) = \frac{e^{\alpha + \beta x_i}}{1 + e^{\alpha + \beta x_i}}$

- Linear regression predict the value Y_i
- Logistic regression predict the probability $P(Y_i = 1)$
- Generalized linear models:
 - ▶ family = distribution + link function
 - ▶ E.g., Binomial + logit for logistic regression
 - ▶ For $Y_i \in \{0, 1\}$, actually Bernoulli + logit *[Binary logistic regression]*
- Since distribution is known, MLE can be adopted for estimating α and β :

$$\ell(\alpha, \beta) = \sum_{i=1}^n [y_i \log(\text{inv.logit}(\alpha + \beta x_i)) + (1 - y_i) \log(1 - \text{inv.logit}(\alpha + \beta x_i))]$$

See R script

Elastic net logistic regression

- Penalized linear regression minimizes:

$$\|\mathbf{y} - \mathbf{X} \cdot \boldsymbol{\beta}\|^2 + \lambda_2 \|\boldsymbol{\beta}\|^2 + \lambda_1 \|\boldsymbol{\beta}\|_1^2$$

- ▶ $\lambda_1 = 0$ is the Ridge penalty
- ▶ $\lambda_2 = 0$ is the Lasso penalty
- Elastic net regularization for logistic regression minimizes:

$$-\ell(\boldsymbol{\beta}) + \lambda \left(\frac{(1 - \alpha)}{2} \|\boldsymbol{\beta}\|^2 + \alpha \|\boldsymbol{\beta}\|_1^2 \right)$$

- ▶ $\alpha = 0$ is the Ridge penalty
- ▶ $\alpha = 1$ is the Lasso penalty
- ▶ λ is to be found, e.g., by cross-validation

See R script