Predictive Analytics

Week 5: Model Selection and Estimation II

Semester 2, 2018

Discipline of Business Analytics, The University of Sydney Business School

Week 5: Model Selection and Estimation II

1. Maximum likelihood for regression

2. Maximum likelihood estimation with gradient ascend

Reading: Chapter 5.1 of ISL to review the cross validation.

Exercise questions: Chapter 5.4 of ISL Q3. Try to implement gradient ascend in Python.

Notation

- Let $p(\boldsymbol{y}; \boldsymbol{\theta})$ denote a probability mass function or density function with associated parameter vector $\boldsymbol{\theta}$.

- $Y_1, Y_2, \ldots, Y_N$ are random variables from this distribution. The random variables are independent.

- $\mathcal{D} = \{y_1, \ldots, y_N\}$ are the actual observed sample values.

- $\widehat{\boldsymbol{\theta}}(\mathcal{D})$ is an estimator of $\boldsymbol{\theta}$.

- $\widehat{\boldsymbol{\theta}}$ an estimator (as above) or estimate of $\boldsymbol{\theta}$ according to the context.

Maximum likelihood for regression

The Gaussian linear regression model

- In Lecture 2, most of our regression analysis did not make any **distributional assumptions** about the regression errors. Even though we assumed conditions such as $E(\varepsilon|X) = 0$ and $\text{Var}(\varepsilon|X) = \sigma^2$, we left the probability distribution of ε unspecified.

- We managed to learn quite a lot from these minimal assumptions. For example, the assumptions for the conditional mean and variance of the errors naturally leads us to the mean and variance of the OLS estimator.

- But we may want to learn more. For example, what is the full sampling distribution of the OLS estimator? Knowing this distribution is necessary for making probability statements about the uncertainty in this estimator.

The Gaussian linear regression model

We now add the assumption that $\varepsilon \sim N(0, \sigma^2)$, leading to the model equation

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2).$$

A key feature of the Gaussian linear regression model is that it gives us the full form of the conditional distribution of Y.

$$Y|X = \boldsymbol{x} \sim N\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j,\, \sigma^2\right)$$

Maximum Likelihood Estimation (MLE)

- **Maximum likelihood** (ML) estimation is available when we specify a full probabilistic model for the population. This is a new concept which we now introduce for the specific case of the Gaussian linear regression model.

- Intuitively, ML estimation chooses the values of the parameters that maximise the likelihood of the observed data under the model (for discrete data the likelihood is the probability of the data, but our response Y is continuous).

- ML is one of the most highly used estimation techniques in statistics.

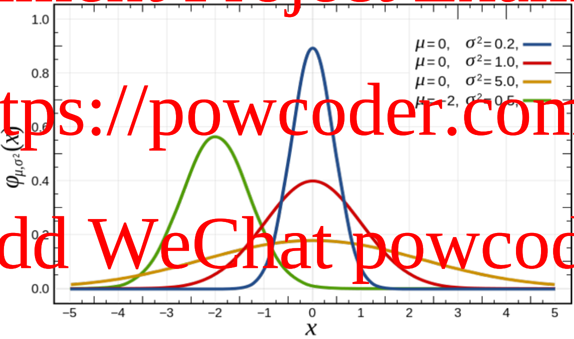
Normal probability density function

Recall the formula for the normal probability density function (PDF) from basic statistics:

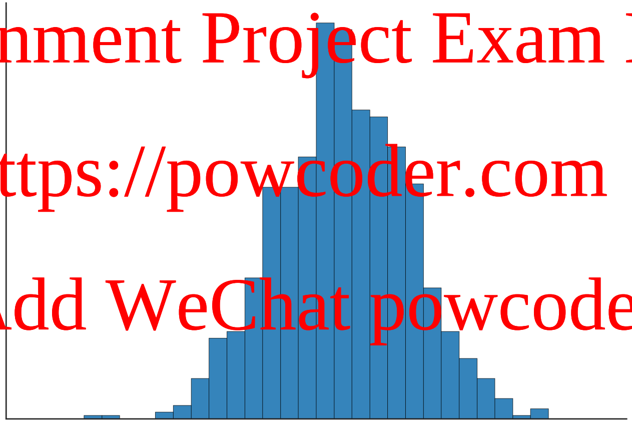
$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

Normal distribution

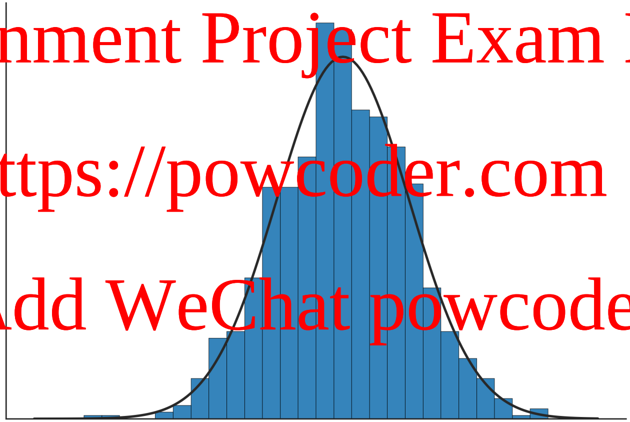
Normal distributions with different μ and σ values



Normal distribution



Normal distribution



The likelihood function

Since

$$Y_i | X_i = \boldsymbol{x}_i \sim N\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \sigma^2\right),$$

the density for an observed value y_i is

$$p(y_i | \boldsymbol{x}_i; \boldsymbol{\beta}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2}{2\sigma^2}}$$

The likelihood function

The **likelihood function** is the joint PDF of the data evaluated at the sample values. In our Gaussian linear regression model, Assumption 4 (independence) of Lecture 2 slide 29 implies that we can multiply the PDFs for each observation:

$$p(\boldsymbol{y}; \boldsymbol{\beta}, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{\left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2}{2\sigma^2}}$$

The log-likelihood function

The complete **log-likelihood** is the log-density of the observed samples,

$$
\begin{aligned}
L(\boldsymbol{\beta}, \sigma^2) &= \log \prod_{i=1}^{N} p(y_i; \boldsymbol{\beta}, \sigma^2) = \sum_{i=1}^{N} \log p(y_i; \boldsymbol{\beta}, \sigma^2) \\
&= -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2
\end{aligned}
$$

- What are the advantageous of taking **log** operation here?

We maximise the log-likelihood as a function of the parameters

$$\max_{\boldsymbol{\beta},\sigma^2} L(\boldsymbol{\beta}, \sigma^2),$$

where

$$L(\boldsymbol{\beta}, \sigma^2) = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

Note that the last term corresponds RSS($\boldsymbol{\beta}$) times a negative multiplier.

We can simplify the log-likelihood function as below, since the removed term will not affect our parameter estimates. Why?

$$\max_{\boldsymbol{\beta}} \; L(\boldsymbol{\beta}),$$

where

$$L(\boldsymbol{\beta}) = -\frac{1}{2N}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

Therefore, ML estimator (maximise) is equivalent to the OLS estimator (minimise) for this model.

Discussion

So far what are the take-aways of OLS?

- We added a new estimation principle to our toolbox and started by understanding it in this sample case.

- ML estimation is broadly applicable and we will use it extensively for supervised learning.

- We need the concept of a log-likelihood for certain model selection methods.

Discussion

When discussing statistical decision theory, we defined the optimal prediction rule

$$\delta(\boldsymbol{x}) = \operatorname*{argmin}_{f(\cdot)} E(L(Y, f(\boldsymbol{x}))|X = \boldsymbol{x})$$

A probabilistic model estimated by ML will allow us to directly approximate the optimal prediction for any loss, since it estimates the full conditional distribution $P(Y|X = \boldsymbol{x})$.

Maximum likelihood estimation with gradient ascend

Maximum likelihood estimation

- How to implement the maximum likelihood estimation (MLE) for regression?

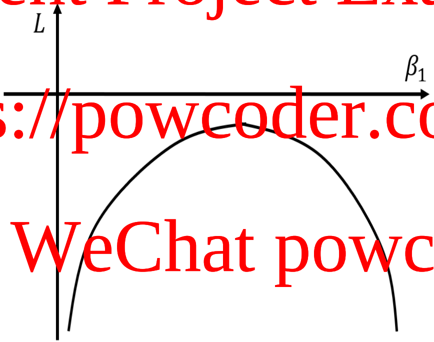
Gradient ascent

- Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function.
- To find a local minimum of a function using gradient descent, we take steps proportional to the **negative of the gradient** (or approximate gradient) of the function at the current point.
- If instead we take steps proportional to the **positive of the gradient**, one approaches a local maximum of that function; the procedure is then known as **gradient ascent**.

Source: wikipedia.

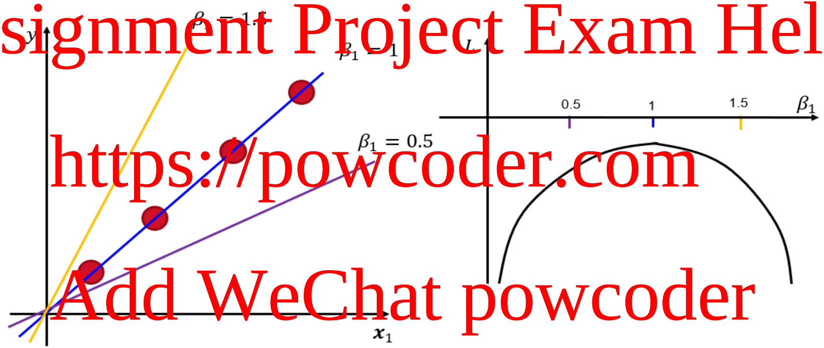
Motivating example

Suppose below is the log-likelihood function plot of a simple linear regression without intercept term:



β_0 is not included for simplicity.

Motivating example



β_0 is not included for simplicity.

Maximum Likelihood Estimation Intuition

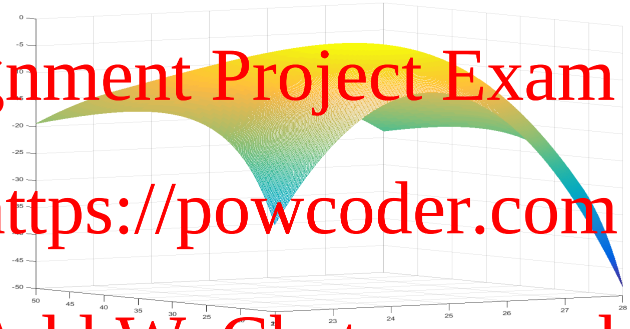


Figure 1: The picture shows an example surface plot of the log-likelihood function $L(\boldsymbol{\beta})$. Gradient vector points to the direction that $L(\boldsymbol{\beta})$ increases.

Maximum Likelihood Estimation

Based on the plot in the pervious slide:

- At the current value β, we move up the hill in the direction of $\frac{\partial L(\beta)}{\partial \beta}$.
- Step size a controls the jump.
- If a is too large, we might jump over the optimal point.
- If a is too small, we might move too slowly.
- How much is the dimension of β based on the above plot? How will be the plot looked like if reduce the dimension of β by 1?

Gradient ascend algorithm

Algorithm Gradient ascend algorithm for maximum likelihood estimation

Initialization: start from some initial guess

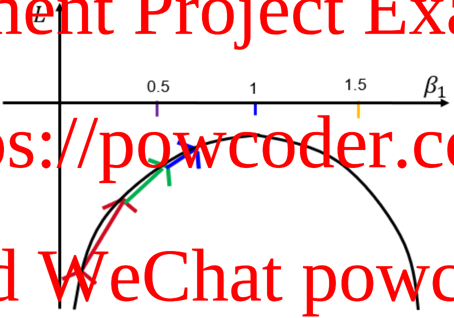
Iterates the following: update until convergence, e.g. likelihood update is less than a threshold

$\beta := \beta + \alpha \frac{\partial L(\beta)}{\partial \beta}$

- $\frac{\partial L(\beta)}{\partial \beta}$ is the **gradient vector** of the function $L(\beta)$. Gradient vector points to the direction that the likelihood function increases. $:=$ is the **assignment** operation.
- α is called learning rate: a small **positive** number that controls the jump in that direction. How to choose α?
- α can be also changed with iteration steps, e.g. $\alpha_t = 1/(1+t)$. t is the iteration step.

Gradient ascend illustration

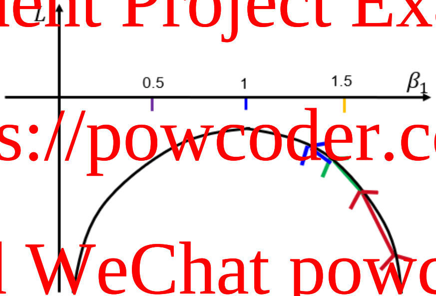
If starting point of β_1 is to the left of the local maximum:



β_1 is updated to be lager and larger. Positive gradient.

Gradient ascend illustration

If starting point of β_1 is to the right of the local maximum:



β_1 is updated to be smaller and smaller. Negative gradient.

Calculating the gradient

Given the log-likelihood function as below:

$$L(\beta) = -\frac{1}{2N}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

We need calculate the gradients (gradient vector) for all parameters:

$$\beta_0, \beta_1, \beta_2, \ldots, \beta_p$$

Calculating the gradient

The gradient vector provides direction of update for each parameter):

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_0} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)$$

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_1} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)x_{i1}$$

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_2} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)x_{i2}$$

$$\vdots$$

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_p} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)x_{ip}$$

Gradient Ascend

So all the parameters are updated simultaneously:

$$\beta_0 := \beta_0 + \alpha \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_0} = \beta_0 + \alpha \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)$$

$$\beta_1 := \beta_1 + \alpha \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_1} = \beta_1 + \alpha \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right) x_{i1}$$

$$\beta_2 := \beta_2 + \alpha \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_2} = \beta_2 + \alpha \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right) x_{i2}$$

$$\cdots$$

$$\beta_p := \beta_p + \alpha \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_p} = \beta_p + \alpha \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right) x_{ip}$$

Gradient Ascend in matrix form

We can write the Gradient Ascend (the ones in previous slides) for linear regression with multiple features in a matrix form. The matrix form looks much more simple. First define:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{1}$$

$$f(\boldsymbol{X}) = \begin{bmatrix} f(x_1) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{1j} \\ f(x_2) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{2j} \\ \vdots \\ f(x_N) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{pj} \end{bmatrix} \tag{2}$$

Gradient Ascend in matrix form

Further define:

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \begin{bmatrix} \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_0} \\ \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_1} \\ \vdots \\ \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_2} \end{bmatrix} \tag{3}$$

Then it can be shown that (essentially re-write slide 29):

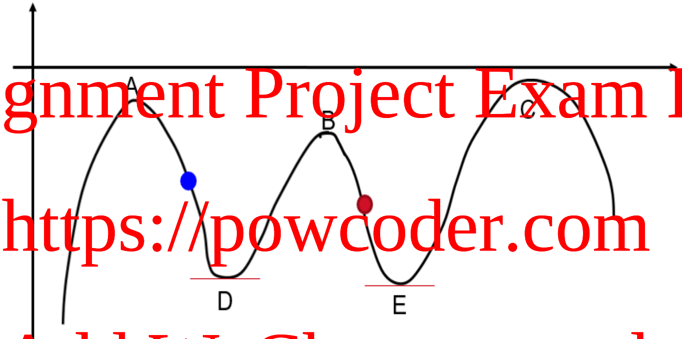
$$\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \frac{1}{N} \boldsymbol{X}^T \left(\boldsymbol{y} - f(\boldsymbol{X}) \right)$$

Hence gradient ascent in matrix form is:

$$\beta := \beta + \frac{\alpha}{N} X^T (y - f(X))$$

- Note the size of each matrix and vector in the above formula.
- Try to implement this in Python.

Why local maximum?



- If the starting point is the red dot, then gradient descent can only converge to local minimum B.
- If the starting point is the blue dot, then gradient descent can only converge to local minimum A.
- The gradients at D and E are 0. The gradients at A, B or C are also 0.

Review questions

- What is maximum likelihood?

- Try to derive the log-likelihood function with Gaussian linear regression.

- What is gradient ascend and how it works?