Bielefeld University · Econometrics · Summer 2024

Statistical and Econometric Models

Lectures: Lennart Oelschläger · Tutorials: Sebastian Büscher ✉

Problem Set – Week 5 (for tutorial on June 03, 2024)

Binary Choice

Problem 5.1:

The file `SEM_SoSe24_week05_data.csv` in the Lernraum provides data a company collected on customers they have send a complimentary sample of their product. The columns `x1`, `x2`, `x3` and `x4` contain information on the customers while the column `y` encodes whether the customer later purchased the product. The company is now interested in modelling this response to the free samples to distribute them in the future more efficiently. For this, you propose to utilise a binary choice model using `R`.

(a) Define the function $\Lambda(\eta_i) = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}$ with $\eta_i = \beta' \boldsymbol{x}_i$ and $\boldsymbol{x}_i = (1 \;\; x_{i,1} \;\; x_{i,2} \;\; x_{i,3} \;\; x_{i,4})'$ in `R`.

(b) Define the probability function for the observation i as

$$\mathbb{P}(Y = y_i | \beta, \boldsymbol{x}_i) = \Lambda(\beta' \boldsymbol{x}_i)^{y_i} (1 - \Lambda(\beta' \boldsymbol{x}_i))^{1-y_i}.$$

(c) Define the log-likelihood function for the logistic regression model and find the maximum likelihood estimator $\hat{\beta}$ by maximising it for the given data.[1]

Problem 5.2:

You consider if a probit model might be better for this data set then the logistic regression model.

(a) Define the probability function for the observation i as

$$\mathbb{P}(Y = y_i | \beta, \boldsymbol{x}_i) = \Phi(\beta' \boldsymbol{x}_i)^{y_i} (1 - \Phi(\beta' \boldsymbol{x}_i))^{1-y_i},$$

with Φ denoting the standard normal cumulative distribution function.

(b) Continue to estimate the probit model by defining and maximising the log-likelihood function.

(c) Which of the two models has the better AIC?

[1] You need good starting parameters for the optimisation. You can use a linear model to obtain decent starting values.

Problem 5.3:

With the models, you can calculate the probability $\hat{p}_i$, given your model, that the customer will purchase the product after receiving a sample. You want to convert this probability to a 0-1 variable $\hat{c}_i$. This can be done using

$$\hat{c}_i = \begin{cases} 0, & \text{if } \hat{p}_i < c \\ 1, & \text{if } \hat{p}_i \geq c \end{cases} .$$

(a) Start categorising by using $c = 0.5$ and calculate the confusion matrix for both models. Which model seems to give better predictions?

	Observed choice	
Predicted choice	0	1
0	TN (# true negatives)	FN (# false negatives)
1	FP (# false positives)	TP (# true positives)

Table 1: Confucion matrix

The value $c = 0.5$ seems a bit arbitrary to you. You want to use the optimal value for your task of predicting the buying behaviour of the customers. Your task is to predict which customer would buy your product after receiving a free sample. The goal is to only send samples to customers who are likely to buy the product afterwards.

- Sending a sample costs you 5 Euro.
- A customer buying your product after receiving a sample will lead to a profit of 20 Euro.
- A customer who did not receive a free sample will not buy your product.

(b) Given this information, find the optimal parameter c^* and visualise the relation between predicted income and c in a plot. Be aware that the number of true positives, true negatives, false positives, and false negatives as functions of c are piecewise continuous functions.

Table 2: Some R code and functions, that might prove to be helpful for these exercises.

R code	function
`install.packages("pracma")`	installs the R package `pracma`
`require("pracma")`	loads the package `pracma`
`X%*%b`	matrix multiplication of matrix `X` with vector `b`
`M*X`	multiplies the elements of matrix `M` with the elements of matrix `X`
`pnorm(x)`	evaluates the standard normal cumulative distribution function Φ at `x`
`trapz(x,y)`	calculates the area under the curve `plot(y x)` (is included in the `pracma` package)
`hessian(f,x)`	returns the hessian matrix of the function `f` at `x` (is included in the `numDeriv` package)
`plot(y~x)`	plots the data pairs $(\mathtt{x}_i, \mathtt{y}_i)$
`lines(w~v, col="red")`	adds the data pairs $(\mathtt{v}_i, \mathtt{w}_i)$ to an existing plot and connects them by a red line