Empirical exercise – The maximum likelihood estimator and the probit model

1. Aim of the exercise

The aim of this exercise is to understand using the maximum likelihood method to estimate the probit model. We will first use simulated data to concentrate on understanding the probit model. We will demonstrate the estimation using real data.

The theoretical context is as follows. The derivation of the probit model depends on an underlying unobserved, or latent, model. The model is stated as $y^* = \beta_0 + \beta_1 x_1 + u$. We view this model as the propensity for an event to occur. As x_1 increases, the propensity that y^* occurs increases. But x_1 can increase indefinitely, and hence the propensity. When will the event y^* occur? We need to set a threshold. Let that threshold be 0. Hence, if $\beta_0+\beta_1 x_1+u > 0$, we consider that the event occurs. Let us indicate this occurrence with $y = 1$. If the propensity is smaller than 0, we indicate it as $y = 0$. We observe y. But what determines y is an unobserved model. We specify y as $y = \mathrm{I}(y^* \geq 0)$, where I is an indicator function that returns 1 if the propensity is larger than the threshold, and 0 otherwise. The interest lies in the probability that the event occurs: $\mathrm{P}(y = 1 \mid x) = \mathrm{P}(y^* > 0 \mid x)$. Hence, $\mathrm{P}(y = 1 \mid x) = \mathrm{P}(y^* > 0 \mid x) = \mathrm{P}(\beta_0 + \beta_1 x_1 + u > 0 \mid x) = \mathrm{P}(u > -(\beta_0 + \beta_1 x_1) \mid x)$. It is here that we make an assumption on how u is distributed. If it has a standard normal distribution, we have the probit model, while if it has a standard logistic distribution we have the logit model. It follows that $\mathrm{P}(u > -(\beta_0 + \beta_1 x_1) \mid x) = 1 - \Phi(-(\beta_0 + \beta_1 x_1)) = \Phi(\beta_0 + \beta_1 x_1)$, where Φ is the cumulative distribution function of the standard normal distribution. It is given by $\Phi(\beta_0 + \beta_1 x_1) = 1/\sqrt{2\pi} \int_{-\infty}^{\beta_0+\beta_1 x_1} \exp(-z^2/2)\, dz$.

Since the model is non-linear in the parameters, we use the maximum likelihood method to estimate the parameters of this model.

We will use the stated cumulative distribution function when constructing the log-likelihood function in the supplied function `exercisemlprobitfunction`.

2. Set seed

Set seed for reproducible results.

```
clear;
rng(1);
```

3. Define the number of observations

Define the number of observations to be used for estimation.

```
N_obs = 500;
```

4. Simulate data

In the code presented below, we first simulate data. We then demonstrate the function of the latent model.

```
x_0 = ones(N_obs,1);
x_1 = random('normal',0,1,N_obs,1);
x_2 = random('normal',0,1,N_obs,1);
x_3 = random('normal',0,1,N_obs,1);
X = [x_0 x_1 x_2 x_3];
mu = 0;
sigma = 1;
e = random('normal',mu,sigma,N_obs,1);
B_true = [1 2 3 4]';
% Latent model
y_star = X*B_true+e;
% Bivariate indicator
y = y_star > 0;
```

5. Estimate the probit model

Estimate the probit model using the supplied function `exercisemlprofunction`. Before the estimation, inspect the content of the function and how the log-likehood function is defined.

```
B_ig = [1 1 1 1]';
B_hat = fminunc(@(B_true)exercisemlprofunction(y,X,B_true),B_ig);
```

6. Load empirical data

In this second exercise we use empirical data. Read the enclosed data definition to have an overview of the data. Our outcome variable takes a value of 1 if the mortgage application of the agent is accepted, and 0 otherwise. We have two regressors at our disposal. An indicator of race, and the amount of the debt the agent has as a fraction of his monthly wage income.

```
clear;
load 'M:\exercisemlpro.mat';
```

7. Define the dependent variables and the matrix of regressors

Define the dependent variables and the matrix of regressors.

```
y = deny;
N = length(y);
x_0 = ones(N_obs,1);
X = [x_0 black piratio];
```

8. Estimate the probit model

Estimate the probit model using the function `exercisemlprobfunction`. The estimate for piratio indicates that an increase in debt-to-income ratio increases the probability of denial.

```
B_ig = [1 1 1]';
B_hat = fminunc(@(B_true)exercisemlprofunction(y,X,B_true),B_ig);
```