Description of the model

In this homework we are going to work with a (binary) classification model known as a logistic or *logit* model. For this classification problem we will assume that we have collected observations $x \in \mathbb{R}^p$, (p observed values for each observation) belonging to two different classes, A and B. We will codify these classes using the value of a variable y, so that $y = 1$ if x belongs to class A, and $y = 0$ if x belongs to class B.

Our classification procedure will be based on (finding and) using a linear function, if it exists, of the form

$$\beta^T x = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p,$$

that takes (large) positive scalar values for observations in A and (large) negative values for observations in B.

To improve the interpretability of the results (and the fitting of the model), the value of the preceding linear function is transformed into a value in the interval $[0, 1]$. Values close to 1 should indicate a high probability for an observation to belong to class A, and values close to 0 should indicate observations in B (with high probability).

The transformation used in logit models, based on the logistic function, is given by

$$\hat{y} = \frac{1}{1 + \exp(-\beta^T x)} \equiv \varphi(\beta; x), \tag{1}$$

where $\hat{y}$ denotes the value predicted by the model.

We will consider two main optimization procedures to obtain the values of β from a sample of values $\{(x_i, y_i)\}_{i=1}^n$:

- *Least squares:* We find the values of β that minimize the sum of the squares of the errors,

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - \varphi(\beta; x_i)\right)^2 .$$

- *Maximum likelihood:* We obtain β as the maximizer of the logarithm of the likelihood function,

$$\max_{\beta} \sum_{i=1}^{n} y_i \log \varphi(\beta; x_i) + \sum_{i=1}^{n} (1 - y_i) \log \left(1 - \varphi(\beta; x_i)\right) .$$

In this homework we have two main goals:

- For a given training set of observations x_i, $i = 1, \ldots, n$, with known classes y_i, we wish to find values for the vector of $(p+1)$ parameters β that provide the best possible fit between the values of $\hat{y}_i$ from (1) and y_i, using the two preceding methods.
- We will test the resulting models using another set of observations, to obtain an out-of-sample measure for the quality of their fits.

Tasks to complete

You are asked to:

1. (*1 point*) Generate random data as observations from a mixture of three normal distributions. To do this:

(a) Set the number of observations to generate as $n = 1000$, and the number of variables for each observation as $p = 20$. Select the proportion of observations in each group as a value α_i following a uniform distribution in $[0.2, 0.3]$ for $i = 1, 2$. Let $n_i = \lfloor \alpha_i n \rfloor$ for $i = 1, 2$ and $n_3 = n - (n_1 + n_2)$.

(b) Generate two random vectors $u_i \in \mathbb{R}^p$, $i = 1, 2$, with unit norm. Define each vector using p values following a standard normal distribution, and divide them by their euclidean norm (2-norm). Define constants $\gamma_1 = 5$ and $\gamma_2 = 4$.

(c) Generate random diagonal matrices D_i, $i = 1, 2$ (in dimension p) using uniform random values from the interval $[0.25, 4]$.

(d) Generate n_i independent observations in dimension p from a multivariate normal distribution with mean the vector $\gamma_i u_i$ and covariance matrix D_i. Assign these observations to a matrix X_i (with n_i rows and p columns), and define a vector Y_i of size n_i with all components equal to 1. Repeat this procedure for $i = 1, 2$.

Note that this procedure is equivalent to generating the observations from a multivariate normal distribution with mean 0 and covariance the identity matrix, and transforming each observation x_{ij} according to

$$\tilde{x}_{ij} = x_{ij} D_i + \gamma_i u_i.$$

(e) Generate n_3 independent observations in dimension p from a multivariate normal distribution with mean 0 and covariance matrix I (the identity matrix). Assign these observations to a matrix X_3 (with n_3 rows and p columns), and define a vector Y_3 with all n_3 components equal to 0.

(f) Create a matrix $X \in \mathbb{R}^{n \times (p+1)}$ by putting together X_1, X_2 and X_3, and adding a column of ones, e, as the first column of the matrix. Create a vector Y by putting together (in the same order as in X) the values of Y_1, Y_2 and Y_3.

$$X = \begin{pmatrix} & X_1 \\ e & X_2 \\ & X_3 \end{pmatrix}, \quad Y = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix}.$$

(g) Plot the observations projected onto the direction u_1 and a direction orthogonal to it (you can select it in any way you prefer). If the observations are not somewhat separated (but note that it would be of interest that they partially overlap), repeat the preceding procedure.

(h) Split the observations into a training and a testing set. Select $n_s = \lfloor 0.2n \rfloor$ ($n_t = n - n_s$) random indices between 1 and N, I_s, and split the matrix X and the vector Y into two by assigning the rows with indices in I_s to matrix X_s and vector Y_s, and the remaining observations (those with indices not in I_s) to matrix X_t and vector Y_t.

2. (*2 points*) Estimate the value of the regression coefficients by using the function **minimize** from the Python module **Scipy.optimize**, based on the least-squares procedure. Briefly comment on your results, and any difficulties you may have found in the solution process.

3. (*2 points*) Modify the preceding optimization model by adding (lower and upper) bounds on the values of the β coefficients. Solve it again by using (at least) two different procedures, which should accept the introduction of bounds on the variables. Compare these methods (for example, by looking at the number of iterations, number of function, gradient and hessian evaluations as well as total running time required) as well as the one used for the preceding question. Briefly comment on possible interpretations of the values of the coefficients.

4. (*1 point*) For the values of β obtained in the preceding questions, compute the number of correctly and incorrectly classified observations in the testing set, by obtaining the values for $\hat{y}_i$ from (1) and comparing them to the corresponding values of y_i. In particular, for the observations in I_s compute the proportions of incorrectly classified observations, such as
$$\#\{i : (\hat{y}_i \geq 0.5)\&(y_i = 0)\}/n_s, \quad \#\{i : (\hat{y}_i < 0.5)\&(y_i = 1)\}/n_s.$$

5. (*2 points*) Estimate the value of the regression coefficients by using the function **minimize** based on the maximum likelihood procedure. Try (at least) two of the available solvers and compare their performance (for example, by looking at the number of iterations, number of function, gradient and hessian evaluations as well as total running time required). Briefly comment on possible interpretations of the values of the coefficients.

6. (*2 points*) An idea that has been shown to be useful in some cases is the inclusion of a regularization term in the optimization problem. These regularization terms aim to introduce additional interesting properties in the solution, while improving the characteristics of the mathematical model.

   In our case, we would like to ensure that the value of β does not become too large, and at the same time that the problem is more convex (implying that its solutions are better defined). One way to do this is to add a term of the form $\frac{\rho}{2}\|\beta\|^2$ to the objective function.

   Modify the preceding optimization model by including this regularization term to obtain the modified objective function
$$\max_{\beta} \frac{1}{n_t}\sum_{i=1}^{n_t} y_i \log \varphi(\beta; x_i) + \frac{1}{n_t}\sum_{i=1}^{n_t}(1 - y_i)\log\left(1 - \varphi(\beta; x_i)\right) - \frac{\rho}{2}\|\beta\|_2^2.$$

   Solve this problem using (at least) two algorithms and try (at least two) different values for ρ from 0.1 to 10. Compare the solution values and times with those obtained in the previous question.

General instructions

- Due date: Friday, January 17, at 11 p.m.
- Upload the code to Aula Global as a Jupyter notebook.
- Name your notebook file as "Surname-Name-A1.ipynb".
- You are strongly advised to include descriptions for your formulations and comments in the same notebook, by using markdown cells.
- If this would prove too complicated, exceptionally you may present this information in a separate pdf file. In this case, name the file "Surname-Name-A1.pdf".
- If you have used a specific dataset (not a randomly generated one), include among the uploaded files.