University of Stuttgart | ANALYTIC COMPUTING

Machine Learning Exercise (SS 21)

Assignment 4: Logistic Regression (Solution)

Dr. Decky Aspandi
decky.aspandi-latif@ipvs.uni-stuttgart.de

Akram Hosseini
Akram.Hosseini@ipvs.uni-stuttgart.de

Daniel Frank
daniel.frank@ipvs.uni-stuttgart.de

This assignment sheet consists of 4 pages with the following 3 tasks:

- Task 1: Classification with Linear Regression (40 Points) 2
- Task 2: Log-likelihood gradient and Hessian (30 Points) 3
- Task 3: Discriminative Function in Logistic Regression (30 Points) 4

Submit your solution in ILIAS as a single PDF file.[1] Make sure to list your full name and immatriculation number at the start of the file. Optionally, you can *additionally* upload source files (e.g. PPTX files). Remember to fill out the exercise slot and exercise presentation polls linked in ILIAS. If you have any questions, feel free to ask them in the excercise forum in ILIAS.

Submission is open until Tuesday, 1st of June 2021, 23:59 PM.

[1] Your drawing software probably allows to export as PDF. An alternative option is to use a PDF printer. If you create multiple PDF files, use a merging tool (like pdfarranger) to combine the PDFs into a single file.

Task 1: Classification with Linear Regression (40 Points)

Consider the following 1-dimensional input $x = [-1.0, -2.0, 0.3, 0.6, 3.0, 6.0]$ with corresponding binary class labels $y = [1, 1, 0, 1, 0, 1]$. Use (least-squares) linear regression, as shown in the lecture, to train on these samples and classify them. Your model should include an intercept term.

1. **Task (15 Points)**: Provide the coefficients β of the linear regression (on x and y) and explain shortly how you computed them.

   Solution: Linear Regression of an indicator matrix. We define

   $$X = \begin{pmatrix} 1 & -1.0 \\ 1 & -2.0 \\ 1 & 0.3 \\ 1 & 0.6 \\ 1 & 3.0 \\ 1 & 6.0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} \beta_0^0 & \beta_0^1 \\ \beta_1^0 & \beta_1^1 \end{pmatrix} = \begin{pmatrix} \beta^0 & \beta^1 \end{pmatrix}$$

   where the first column of B refers to the vector β^0 for the 0 class case and the second column respectively to the vector β^1 for the 1 class case.

   To compute the β matrix B, we use the corresponding formula ($B = (X^T X)^{-1} X^T Y$). This way, we get

   $$B = \begin{pmatrix} \frac{7813}{25509} & \frac{17696}{25509} \\ \frac{200}{8503} & -\frac{200}{8503} \end{pmatrix} \approx \begin{pmatrix} 0.306 & 0.694 \\ 0.024 & -0.024 \end{pmatrix}$$

2. **Task (15 Points)**: Classify each of the 6 samples with your linear regression model. Explain how you map the continuous output of the linear model to a class label.

   Solution: Since we computed the β matrix B in the previous step, we can use this to predict an output with the corresponding formula ($\hat{Y} = XB$). This way, we get

   $$\hat{Y} = \begin{pmatrix} \frac{7213}{25509} & \frac{18296}{25509} \\ \frac{6613}{25509} & \frac{18896}{25509} \\ \frac{7993}{25509} & \frac{17516}{25509} \\ \frac{743}{2319} & \frac{1576}{2319} \\ \frac{9613}{25509} & \frac{15896}{25509} \\ \frac{11413}{25509} & \frac{14096}{25509} \end{pmatrix} \approx \begin{pmatrix} 0.283 & 0.717 \\ 0.259 & 0.741 \\ 0.313 & 0.687 \\ 0.320 & 0.680 \\ 0.377 & 0.623 \\ 0.447 & 0.553 \end{pmatrix}$$

   Now we can classify according to the argmax-function of the $\hat{Y}$ matrix. This gives us the following class predictions for every data point $\hat{y} = [1, 1, 1, 1, 1, 1]$. As you can see, a linear function is (of course) not able to differentiate this example correctly. An interesting side effect is that the values of the $\hat{Y}$ matrix look like probabilities (since they sum up to 1 for each row).

3. **Task (10 Points)**: Discuss in your own words, why linear regression is not suitable for classification.

   Solution: Linear regression is not suitable for classification because:
   1. In linear regression the predicted value is continuous, whereas in classification we want probabilisties in discrete values.
   2. It cannot predict well for unbalanced data, in other words it is sensitive to unbalanced data.
   3. In linear regression threshold value shifts when new data are added.

Task 2: Log-likelihood gradient and Hessian (30 Points)

Consider a binary classification problem with data $D = \{(x_i, y_i)\}_{i=1}^n$, $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. We define

$$f(x) = \phi(x)^T \beta \ , \quad p(x) = \sigma(f(x)) \ , \quad \sigma(z) = 1/(1 + e^{-z})$$

$$L^{\text{nll}}(\beta) = -\sum_{i=1}^{n} \Big[y_i \log p(x_i) + (1 - y_i) \log[1 - p(x_i)] \Big]$$

where $\beta \in \mathbb{R}^d$ is a vector. (Note: $p(x)$ is a short-hand for $p(y = 1|x)$.)

1. **Task (15 Points)**: Compute the derivative $\frac{\partial}{\partial \beta} L(\beta)$.
   Tip: Use the fact that $\frac{\partial}{\partial z} \sigma(z) = \sigma(z)(1 - \sigma(z))$.
2. **Task (15 Points)**: Compute the 2nd derivative $\frac{\partial^2}{\partial \beta^2} L(\beta)$.

   Solution: Let $p_i \equiv p(x_i)$. We have $\frac{\partial}{\partial \beta} p_i = p_i(1 - p_i)\phi(x_i)^T$

$$\begin{aligned}
L(\beta) &= -\sum_{i=1}^{n} \Big[y_i \log p_i + (1 - y_i) \log[1 - p_i] \Big] \\
\frac{\partial}{\partial \beta} L(\beta) &= -\sum_{i=1}^{n} \Big[y_i \frac{p_i(1 - p_i)}{p_i} \phi(x_i)^T + (1 - y_i) \frac{-p_i(1 - p_i)}{1 - p_i} \phi(x_i)^T \Big] \\
&= -\sum_{i=1}^{n} \Big[y_i(1 - p_i) - (1 - y_i)p_i \Big] \ \phi(x_i)^T \\
&= \sum_{i=1}^{n} \Big[p_i - y_i \Big] \ \phi(x_i)^T = (p - y)^T X \\
\frac{\partial^2}{\partial \beta^2} L(\beta) &= \frac{\partial}{\partial \beta} \sum_{i=1}^{n} \phi(x_i) \Big[p_i - y_i \Big] \\
&= \sum_{i=1}^{n} \phi(x_i) p_i (1 - p_i) \phi(x_i)^T = X^T W X \ , \quad W = \text{diag}(p \circ (1 - p))
\end{aligned}$$

Task 3: Discriminative Function in Logistic Regression (30 Points)

Logistic Regression defines class probabilities as proportional to the exponential of a discriminative function:

$$P(y|x) = \frac{\exp f(x,y)}{\sum_{y'} \exp f(x,y')}$$

1. **Task (30 Points)**: Prove that, in the binary classification case, you can assume $f(x,0) = 0$ without loss of generality.

   This results in

   $$P(y=1|x) = \frac{\exp f(x,1)}{1+\exp f(x,1)} = \sigma(f(x,1)).$$

   (Hint: First assume $f(x,y) = \phi(x,y)^T\beta$, and then define a new discriminative function f' as a function of the old one, such that $f'(x,0) = 0$ and for which $P(y|x)$ maintains the same expressibility.)

Solution: Assume $f(x,y) = \phi(x,y)^T\beta$. Define new discriminative function $f'(x,y) = f(x,y) - f(x,0)$.

Proof that new discriminative function f' still fulfills class probabilites:

$$\begin{aligned}
P'(y|x) &= \frac{\exp f'(x,y)}{\sum_{y'} \exp f'(x,y')} \\
&= \frac{\exp(f(x,y) - f(x,0))}{\sum_{y'} \exp(f(x,y') - f(x,0))} \\
&= \frac{\exp(f(x,y)) \cdot \exp(-f(x,0))}{\sum_{y'} \exp(f(x,y')) \cdot \exp(-f(x,0))} \\
&= \frac{\exp(-f(x,0))}{\exp(-f(x,0))} \cdot \frac{\exp(f(x,y))}{\sum_{y'} \exp(f(x,y'))} \\
&= \frac{\exp f(x,y)}{\sum_{y'} \exp f(x,y')} = P(y|x)
\end{aligned}$$