INTRODUCTION TO MACHINE LEARNING

LOGISTIC REGRESSION & SOFTMAX

* Some contents are adapted from Dr. Hung Huang and Dr. Chengkai Li at UT Arlington

Mingon Kang, Ph.D.
Department of Computer Science @ UNLV
Logistic Regression

- Predict results on a **binary** outcome variable
  - E.g.,
    - Whether or not a patient has a disease
    - Whether a new applicant would succeed in the program or not
  - The outcome is not continuous or distributed normally
A problem with linear regression

When we have a binary response variables,

- We code “disease” as 1 and “no disease” as 0, can we just fit a line through those points as we would with linear regression?

- Possible! But some problems.
A problem with linear regression

- The problem of fitting a regular regression line to a binary dependent variable
A problem with linear regression

- The line seems to oversimplify the relationship.
- It gives predictions that cannot be observable values of Y for extreme values of X.
- The approach is analogous to fitting a linear model to the probability of the event, but now we need 1 or 0.
A problem with linear regression

- The linear model does not predict the maximum likelihood estimates for each group (circles)
- Produces unobservable predictions for extreme values of dependent variable
In the OLS regression:

\[ y = Xb + e, \text{ where } y = (0, 1) \]

- \( e \) is not normally distributed because \( Y \) takes on only two values \((0, 1)\)
- The predicted probabilities can be greater than 1 or less than 0
Probabilistic Approach

- Learn $P(Y \mid X)$ directly
  - Cumulative probability distribution
  - Using a sigmoid function
Probabilistic Approach

- $P(Y|X)$ using sigmoid function

  - $P(Y = 0|X) = \frac{1}{1 + \exp(Xb)}$

  - $P(Y = 1|X) = 1 - P(Y = 0|X) = \frac{\exp(Xb)}{1 + \exp(Xb)}$
    
    $= \frac{1}{1 + \exp(-Xb)}$
Understanding the sigmoid

\[ g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}} \]

- \( w_0 = -2, w_1 = -1 \)
- \( w_0 = 0, w_1 = -1 \)
- \( w_0 = 0, w_1 = -0.5 \)
Logistic Regression Model

\[ P(Y = 0|X) = \frac{1}{1 + \exp(Xb)} \]

\[ \log \left( \frac{P(Y = 1|X)}{P(Y = 0|X)} \right) = Xb \]
Logistic Regression Model

\[ \log\left(\frac{p}{1 - p}\right) = Xb \]

- \( p \) is the probability that an event \( Y \) occurs, \( p(Y=1) \)
- \( \frac{p}{1-p} \) is the “Odds ratio”
  - Range of [0 to infinite]
- \( \log(p/(1-p)) \) is log odds ratio, or “logit”
  - Range of [-infinite to infinite]
Logistic function

- Logistic function
  - $Y = \log\left(\frac{p}{1-p}\right)$
Logistic regression

- Odds transformed through the logistic function (known as logit score)
- Ranges from negative infinity to positive infinity: ideal for linear regression

<table>
<thead>
<tr>
<th>Odd</th>
<th>Logit score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.99</td>
<td>1.996</td>
</tr>
<tr>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>0.25</td>
<td>-0.477</td>
</tr>
<tr>
<td>0.01</td>
<td>-1.996</td>
</tr>
</tbody>
</table>
Maximum Likelihood Estimation (MLE)

- **Likelihood function**: probability of the observed data as a function of the unknown parameters.

- To estimate unknown parameters that maximize the likelihood of getting the data we observed in a probabilistic model.

\[
L(\theta; x_1, ..., x_n) = P(X_1 = x_1, ..., X_n = x_n | \theta) \\
= p(x_1 | \theta) \cdots p(x_n | \theta) = \prod_{i=1}^{n} p(x_i | \theta)
\]
Maximum Likelihood Estimation (MLE)

- Example
  - [https://onlinecourses.science.psu.edu/stat414/node/191](https://onlinecourses.science.psu.edu/stat414/node/191)
Likelihood Function

- Logistic regression predicts probabilities rather than classes: **Stochastic approach**
  - Fit the model using likelihood
  - Maximum Likelihood Estimation

\[
L(b; \mathbf{X}) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}
\]

\[
\propto \sum_{i=1}^{n} \log(p(x_i)^{y_i}) + \log((1 - p(x_i))^{1-y_i})
\]

\[
= \sum_{i=1}^{n} y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i))
\]
The log-likelihood turns products into sums:

\[
\sum_{i=1}^{n} (y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)))
\]

\[
= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i \log \left( \frac{p(x_i)}{1 - p(x_i)} \right)
\]

\[
= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i (x_i b)
\]

\[
= \sum_{i=1}^{n} \log\left( \frac{1}{1 + e^{-x_i b}} \right) + \sum_{i=1}^{n} y_i (x_i b)
\]
Likelihood Function

- Derivative with respect to one component $b_j$

\[
\frac{\partial l}{\partial b_j} = - \sum_{i=1}^{n} \frac{e^{x_i b}}{1 + e^{x_i b}} x_{ij} + \sum_{i=1}^{n} y_i x_{ij}
\]

But, no exact solution, So approximation

* Derivative of log and sigmoid function

\[
\frac{d}{dx} (e^x) = e^x
\]

\[
\frac{d}{dx} \left( \frac{1}{1+e^{-x}} \right) = \frac{1}{1+e^{-x}} \left( 1 - \frac{1}{1+e^{-x}} \right)
\]
Negative log-Likelihood Function

- Negative log-likelihood
  - Minimization problem
    \[
    \mathbf{b}^* = \arg\min_{\mathbf{b}} - \sum_{i=1}^{n} (y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)))
    \]
  - Solve by Gradient Descent
Softmax classifier

- A generalization of the binary form of Logistic Regression
- Can be applied for multi-label classification
- Widely used in Deep Learning
Discussion

- How to evaluate the performance?
- How to make a decision with Logistic regression?
Softmax classifier

- In Logistic regression, the score $y = Xb$ is not normalized

- In Logistic regression, $p(y|X) = \frac{1}{1+e^y}$

- In Software classifier, $p(y = k|X) = \frac{e^{yk}}{\sum_{j} e^{yj}}$
## Softmax classifier

<table>
<thead>
<tr>
<th>Label</th>
<th>y</th>
<th>Exp(y)</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>3.2</td>
<td>24.5</td>
<td>0.13</td>
</tr>
<tr>
<td>C2</td>
<td>5.1</td>
<td>164.0</td>
<td>0.87</td>
</tr>
<tr>
<td>C3</td>
<td>-1.7</td>
<td>0.18</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Likelihood Function

- Training data: \( D(X_i, y_i), 1 \leq i \leq n \)

\[
L(b; X) = \prod_{i=1}^{n} p(y_i | X_i)
\]

Take a logarithmic function

\[
\sum_{i=1}^{n} \log(p(y_i | X_i))
\]