Classification with Logistic Regression

Chad Wakamiya
Spring 2020
Agenda

**Classification**
Introduction to types of classification and set up.

**Logistic Regression**
The logistic regression formula and intuition.

**Multiclass Classification**
Extending logistic regression for datasets with multiple features.
Classification
Classification

*Classification* is the problem of assigning observations to one or more categories.

**Binary Classification**
Involves *only 2 classes*

**Multiclass Classification**
Involves *more than 2 classes*

**Examples**

*Binary*
- Spam detection
- Churn/no churn customer retention
- Develop Diabetes/don’t
- Default/repay loan

*Multiclass*
- Image recognition
- Natural language processing

Images from https://www.coursera.org/learn/machine-learning
Classification

We have **features** and **labels** data \((X, Y)\):

\[
(x_1, y_1) \\
(x_2, y_2) \\
\vdots \\
(x_n, y_n)
\]

- **Features**: \(x_i\) is a vector (or even matrix) for each data element
  - For a picture: \(x_i = [32 \times 32 \times 3]\): array of numbers
- **Actual Labels**: \(y_i \in \{0, 1\}\)
  - If picture \(i\) is a dog, \(y_i = 1\)
  - If picture \(i\) is a cat, \(y_i = 0\)

\[\hat{y}_i \in \{0, 1\}\]

**Classification Model**

- **Parameters**: \(W\) is the model weights
  - Coefficients in a regression model

*Sometimes -1 vs 1 instead of 0 vs 1*
Logistic Regression
Logistic Function

The logistic function $\sigma(t)$ can be used to classify binary observations.

- When $t$ is large, $\sigma(t) \rightarrow 1$
- When $t$ is small, $\sigma(t) \rightarrow 0$

$$\sigma(t) = \frac{1}{1+e^{-t}}$$

[Graph of the logistic function]

https://en.wikipedia.org/wiki/Sigmoid_function
1. Plot of a binary data set
Logistic Regression vs. Linear Regression

1. Plot of a binary data set
2. Fit a linear regression model. (Not a good estimator!)
1. Plot of a binary data set

2. Fit a **linear regression** model. (Not a good estimator!)

3. Fit a **logistic regression** model (Desired binary behavior!)
Logistic Regression

- Notice that $0 < \sigma(t) < 1$ for all real numbers $t$, so we can use the logistic function to model the probability that an observation belongs to a certain class.
- If $t = w_0 + w_1 x$ we can use the logistic function to write:

$$P(Y = 1 | x) = \sigma(t) = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$

Probability the image is a dog given features $x$

**Example:**

$P(Y = 1 | x) = \frac{1}{2}$ when $w_0 + w_1 x = 0$
Logistic Regression Threshold

We can select some threshold ($\text{Prob} = 0.5$)

- If $P(Y=1|x) > 0.5 \Rightarrow$ Predict Dog ($\hat{Y} = 1$)
- If $P(Y=1|x) < 0.5 \Rightarrow$ Predict Cat ($\hat{Y} = 0$)

$$P(Y = 1|x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$
1. **Problem:** Will student $i$ pass given $i$ studies $x_i$ hours?

   \[ \hat{y}_i = \begin{cases} 
   1, & \text{if pass} \\
   0, & \text{if fail} 
   \end{cases} \]

2. **Model:** We use this curve to predict the probability that the student would pass given $x$ hours of study.

3. **Classify:** If Prob $> 0.5$, we predict the student will pass the exam.

### Data:
- $x =$ hours studied:
- $y =$ pass/not pass:

<table>
<thead>
<tr>
<th>Hours</th>
<th>0.50</th>
<th>0.75</th>
<th>1.00</th>
<th>1.25</th>
<th>1.50</th>
<th>1.75</th>
<th>2.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Prob. of passing vs. hours studying**

If study time $> 2.75$ hrs, we predict the student will pass.
**Multi-feature Logistic Regression**

The logistic function can be extended for multiple features \((d\) dimensions) if we write:

\[
t = x_i^T W = w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \ldots + w_d x_{i,d}
\]

\((\text{in matrix form})\)

\[
P(Y = 1 | x_i) = \sigma(x_i^T W) = \frac{1}{1 + e^{-(w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \ldots + w_d x_{i,d})}}
\]

Probability the image is a dog given features \(x_i\).
Training a Logistic Regression Classifier (Find $W$)
Steps to Train a Classifier Model

1. Choose our **model** to estimate $y_i$
   
   $$f_W(x_i) = \hat{y}_i = \begin{cases} 1, & \text{if Prob > threshold} \\ 0, & \text{otherwise} \end{cases}$$

2. Define a **loss function** ($L$)
   
   → Allows for scoring each sample point (picture) $x_i$

3. Optimize across the parameter space ($W$) to minimize the **loss function** to some small threshold

**Goal:** Find the best values for the **model parameters** $W$. 

---

**Berkeley SCET**
1. Choose a Model (Review)

We select a logistic regression model and set the threshold to 0.5:

\[ f_W(x_i) = \hat{y}_i = \begin{cases} 1, & \text{if } p_i > 0.5 \\ 0, & \text{otherwise} \end{cases} \]

**where:**

\[ p_i = P(Y = 1|x_i) = \sigma(x_i^T W) \]

Next: Find the parameters \( W = [w_0, w_1, w_2, \ldots w_d] \)
2. Define a Loss Function

Logistic regression commonly uses the **Cross Entropy** loss function to score predictions:

\[ L_i = -y_i \ln(p_i) - (1 - y_i) \ln(1 - p_i) \]

- If the predicted label is **wrong** the **loss is large** and if the predicted label is **right** the **loss is small**.
- Since \( y_i \) is binary there are 2 cases:

I. \( y_i = 0 \) \( \Rightarrow \) \( L_i = -\ln(1-p_i) \)
   - If \( \hat{y_i} = 0 \) \( \Rightarrow \) \( p_i \) is near 0 \( \Rightarrow \) **Loss is -0**
   - If \( \hat{y_i} = 1 \) \( \Rightarrow \) \( p_i \) is near 1 \( \Rightarrow \) **Loss is large**

II. \( y_i = 1 \) \( \Rightarrow \) \( L_i = -\ln(p_i) \)
   - If \( \hat{y_i} = 0 \) \( \Rightarrow \) \( p_i \) is near 0 \( \Rightarrow \) **Loss is large**
   - If \( \hat{y_i} = 1 \) \( \Rightarrow \) \( p_i \) is near 1 \( \Rightarrow \) **Loss is -0**

\[ p_i = P(Y = 1|x_i) = \sigma(x_i^T W) \]
3. Optimize Across the Parameter Space

- We want the $W$ with the **lowest average loss** across all data points in our training or test set.

  \[
  \text{Average Loss} = \frac{1}{n} \sum_{i=1}^{n} L_i
  \]

- Formally this can be written as:

  \[
  W^* = \arg \min_W \frac{1}{n} \sum_{i=1}^{n} -y_i \ln(\sigma(x_i^T W)) - (1 - y_i) \ln(1 - \sigma(x_i^T W))
  \]

  \[
  \text{Average Loss}
  \]

- With regularization, the average loss function is convex $\Rightarrow$ **solve for $W^*$**
Logistic Regression Classifier Summary

1. Choose our **model** to estimate \( y_i \)

\[
f_W(x_i) = \begin{cases} 
1, & \text{if } p_i > 0.5 \\
0, & \text{otherwise}
\end{cases}
\]

\( p_i = \sigma(x_i^T W) \)

2. Define a **loss function** \((L)\)
   
   \( L_i = -y_i \ln(p_i) - (1 - y_i) \ln(1 - p_i) \)

3. Optimize across the parameter space \((W)\) to **minimize the loss function** to some small threshold

\[
W^* = \arg \min_W \frac{1}{n} \sum_{i=1}^{n} -y_i \ln(\sigma(x_i^T W)) - (1 - y_i) \ln(1 - \sigma(x_i^T W))
\]
Multiclass Classification
Multiclass Classification

Logistic regression can be applied to solve multiclass problems.

Common Approaches

1. One-vs-Rest (One-vs-All)
2. Softmax Regression (Multinomial Logistic Regression)
One-vs-Rest (One-vs-All)

For each class, build a logistic regression to find the probability the observation belongs to that class. For each data point, predict the class with the highest probability.

\[
P(Y = 0 | x) \\
P(Y = 1 | x) \\
P(Y = 2 | x)
\]

Predict the class with the highest probability
**Softmax Regression (Multinomial Logistic Regression)**

In softmax regression the probability that a data point belongs to each class is calculated by:

\[
\begin{bmatrix}
P(Y = 1|x; \theta) \\
P(Y = 2|x; \theta) \\
\vdots \\
P(Y = K|x; \theta)
\end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x)}
\begin{bmatrix}
\exp(\theta^{(1)\top} x) \\
\exp(\theta^{(2)\top} x) \\
\vdots \\
\exp(\theta^{(K)\top} x)
\end{bmatrix}
\]

Separate \( \theta^{(i)} \in \mathbb{R}^d \) for each class

Normalize probabilities so they sum to 1.

Predict the class with the highest probability

If \( K = 2 \), softmax regression reduces to the same binary logistic regression formulas we saw earlier. Check out this [overview of softmax regression](#) for the proof.
References

- **DataX (IEOR 135/290)** - Ikhlac Sidhu and Arash Nourian
  - The content presented in this lecture draws on materials by the IEOR 135 course instructors.

- **Logistic Regression**
  - [https://en.wikipedia.org/wiki/Logistic_regression](https://en.wikipedia.org/wiki/Logistic_regression)

- **Data Science Principles and Techniques (DS 100 at UC Berkeley)** - Ani Adhikari, Joseph E. Gonzalez
  - [http://www.ds100.org/sp20/syllabus/](http://www.ds100.org/sp20/syllabus/)

- **Softmax Regression**

- **One-vs-all**
Images for Notebook
Splitting the Dataset

- Training Set
- Validation Set
- Test Set
## Confusion Matrix

<table>
<thead>
<tr>
<th></th>
<th>Predicted Positive (1)</th>
<th>Predicted Negative (0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actually Positive (1)</td>
<td>True Positive</td>
<td>False Negative</td>
</tr>
<tr>
<td>Actually Negative (0)</td>
<td>False Positive</td>
<td>True Negative</td>
</tr>
</tbody>
</table>