CS 6980: Introduction to Data Science © Spring 2018

Mini-Project 2: Logistic Regression & Disaster Survival

Instructor: Daniel L. Pimentel-Alarcón Due 02/12/2018

In this mini-project you will use logistic regression to determine whether you would have survived the Titanic sinking. To find out, we will use the titanic dataset (`titanic_data.csv`), containing the following information of 887 passengers: 1) whether they survived or not (1 = survived, 0 = deceased), 2) passenger class, 3) gender (0 = male, 1 = female), 4) age, 5) number of siblings/spouses aboard, 6) number of parents/children aboard, and 7) fare:

	Passenger 1	Passenger 2	Passenger 3	···	Passenger 887
Survived	**0**	**1**	**1**	···	**0**
Passenger Class	3	1	3	···	3
Gender	0	1	1	···	0
Age	22	38	26	···	32
Siblings/Spouses	1	1	0	···	0
Parents/Children	0	0	0	···	0
Fare	7.25	71.2833	7.925	···	7.75

Our goal is to construct a classifier that determines/predicts whether an individual would survive or not. Let $y_{\rm i} \in \{0,1\}$ be the *label* indicating whether the $\rm i^{th}$ individual survived, and let $\mathbf{x}_{\rm i} \in \mathbb{R}^6$ denote the feature vector of the $\rm i^{th}$ individual (containing all remaining variables). For example, $y_1 = 0$ and $\mathbf{x}_1 = \begin{bmatrix} 3 & 0 & 22 & 1 & 0 & 7.25 \end{bmatrix}^{\mathsf{T}}$. Our goal is to construct a classifier that given $\mathbf{x}$ determines y.

In this mini-project we will use logistic regression, whose classifier has the form:

$$\underbrace{\frac{1}{1+e^{-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}}}}_{\mathsf{P}(y=1|\mathbf{x})} \ \underset{\hat{y}=0}{\overset{\hat{y}=1}{\gtrless}} \ \underbrace{1-\frac{1}{1+e^{-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}}}}_{\mathsf{P}(y=0|\mathbf{x})}, \tag{2.1}$$

and is parametrized by the coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^6$, which we aim to find by maximizing:

$$\ell(\boldsymbol{\beta}) \ := \ \sum_{\rm i=1}^{\rm N} \log \left[\left(\frac{1}{1+e^{-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}_{\rm i}}} \right)^{y_{\rm i}} \left(1 - \frac{1}{1+e^{-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}_{\rm i}}} \right)^{1-y_{\rm i}} \right], \tag{2.2}$$

which is simply an other way to write (2.1) for N training samples.

(a) Create a function that implements (2.2).

(b) The gradient of $\ell(\boldsymbol{\beta})$, which is also a vector in $\mathbb{R}^6$, is given by:

$$\nabla \ell(\boldsymbol{\beta}) \ = \ \sum_{\rm i=1}^{\rm N} \left(y_{\rm i} - \frac{1}{1+e^{-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}_{\rm i}}} \right) \mathbf{x}_{\rm i} \tag{2.3}$$

Create a function that implements (2.3).

(c) Since we cannot simply set (2.3) to zero and solve for $\boldsymbol{\beta}$, we have to use optimization techniques like gradient ascent, whose update is given by:

$$\boldsymbol{\beta}_{\mathrm{t+1}} \;=\; \boldsymbol{\beta}_{\mathrm{t}} \;+\; \eta \nabla \ell(\boldsymbol{\beta}_{\mathrm{t}}), \tag{2.4}$$

where $\eta \in \mathbb{R}$ is the *step size* (often called *learning parameter*). Gradient ascent simply iterates (2.4) until $\ell(\boldsymbol{\beta}_{\mathrm{t}})$ converges. Create a function that implements gradient ascent.

(d) Randomly split your data into training (80%) and testing (20%).

(e) Run gradient ascent on your training data for different values of η. If η is too big, you may run into numerical errors or loose accuracy. If η is too small, it may take too long to converge. What value of η seems best to maximize $\ell(\boldsymbol{\beta})$? What is the best (largest) $\ell(\boldsymbol{\beta})$ you can achieve?

(f) What coefficient vector $\boldsymbol{\beta}$ do you obtain using your choice of η from (e)? How accurately does this predict survival on the test data?

(g) What would be *your* feature vector. According to your classifier, would *you* have survived? Under which circumstances (passenger class, family aboard, and fare), would your prediction be different?

(h) According to your answer from (f), which seem to be the 3 features that most affect survival? Visualize survivals as a function of these variables.

I have created the following code to help you get started:

```
% © Daniel L. Pimentel-Alarcón, 2018, http://danielpimentel.github.io
close all; clear all; clc;

% =================== LOAD DATA ===================
data = csvread('titanic_data.csv',1,0);
Y = data(:,1)';     % labels
X = data(:,2:end)'; % feature vectors

% =================== SPLIT DATA ===================
Y_train = % COMPLETE HERE: 80% of labels
X_train = % COMPLETE HERE: 80% of features
Y_test = % COMPLETE HERE: 20% of labels
X_test = % COMPLETE HERE: 20% of features

% ========== GRADIENT ASCENT ==========
eta = % COMPLETE HERE: Choose step size
tol = % COMPLETE HERE: Choose tolerance for convergence
beta = gradientAscent(Y_train,X_train,eta,tol); %COMPLETE HERE: Code this function

% ========== TEST ==========
Y_hat = classify_logReg(X_test,beta); %COMPLETE HERE: Code this function
error = sum(abs(Y_hat - Y_test)) / length(Y_test)

% ========== WOULD I HAVE SURVIVED? ==========
my_class = %COMPLETE HERE: What class would you have bought?
my_gender = %COMPLETE HERE: 0=male, 1=female
my_age = %COMPLETE HERE: Your age
my_ss = %COMPLETE HERE: How many spouse/siblings would you have traveled with?
my_pc = %COMPLETE HERE: How many parents/children would you have traveled with?
idx = find(X(1,:)==my_class); % people in the same class as me
my_fare = mean(X(6,idx)); % average fare in my class

% Construct my feature vector
my_x = %COMPLETE HERE: Put together your feature vector

% Classify
my_y = classify_logReg(my_x,beta) %COMPLETE HERE: (You already coded this function above)

% ======= VISUALIZE 3 MOST IMPORTANT VARIABLES =======
%COMPLETE HERE: Hint: you may use bar/pie plots.
```

References

[1] Xiaoli Fern, *Xiaoli Fern*, available at `http://web.engr.oregonstate.edu/~xfern/classes/cs534/notes/logistic-regression-note.pdf`