LOGISTIC REGRESSION – BIS

Classification and Prediction

Lê Hồng Phương

<*phuonglh@gmail.com*>
Vietnam National University of Hanoi
Hanoi University of Science

March 2015

Content

1. Unregularized Logistic Regression
   - Example: Admission Prediction

2. Regularized Logistic Regression
   - Example: Microchip Quality

Logistic Regression Hypothesis

- The logistic regression hypothesis is defined as

$$h_\theta(\mathbf{x}) = g(\theta^T \mathbf{x}),$$

where $g(\cdot)$ is the sigmoid function

$$g(z) = \frac{1}{1 + \exp(-z)}.$$

- Parameter vector of the model:

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_D \end{pmatrix} \in \mathbb{R}^{D+1}$$

Cost Function and Gradient

- Given a training set of N examples $(\mathbf{x}_i, y_i)$, the cost function in logistic regression is:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log h_\theta(\mathbf{x}_i) + (1 - y_i)\log(1 - h_\theta(\mathbf{x}_i))].$$

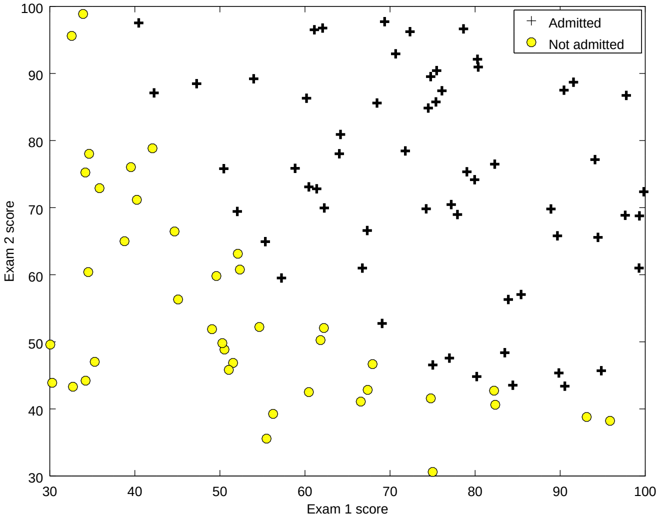
- The gradient of the cost is a vector of the same length as θ:

$$\nabla J(\theta) = \left(\frac{\partial J(\theta)}{\partial \theta_0}, \frac{\partial J(\theta)}{\partial \theta_1}, \dots, \frac{\partial J(\theta)}{\partial \theta_D}\right),$$

where

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{N}\sum_{i=1}^{N}[h_\theta(\mathbf{x}_i) - y_i]x_{ij}, \quad \forall j = 0, 1, \dots, D.$$

Examination Scores



Parameter Estimation

- At $\theta = \vec{0}$:

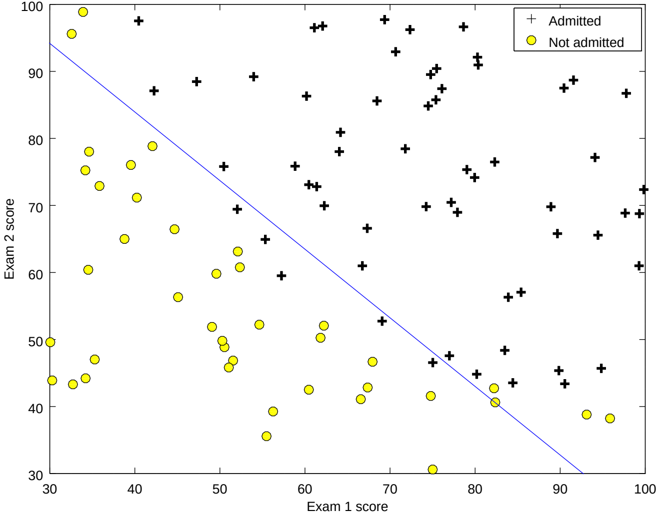
$$J(\theta) = 0.693147$$
$$\nabla J(\theta) = (-0.100000, -12.009217, -11.262842)^T$$

- Estimated parameters:

$$\widehat{\theta} = \begin{pmatrix} -25.161272 \\ 0.206233 \\ 0.201470 \end{pmatrix}$$

- Minimal cost: $J(\widehat{\theta}) = 0.203498$.

Examination Scores – Decision Boundary



Admission Prediction

- Given $\mathbf{x} = (45, 85)$, we predict an admission probability of 0.776289.
- Training accuracy: 89.00%.

Regularized Cost Function and Gradient

- Regularized cost function:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log h_\theta(\mathbf{x}_i) + (1-y_i)\log(1-h_\theta(\mathbf{x}_i))] + \frac{\lambda}{2N}\sum_{j=1}^{D}\theta_j^2.$$

  Note that we should not regularize the parameter θ_0.

- The gradient of the cost is a vector of the same length as θ:

$$\nabla J(\theta) = \left(\frac{\partial J(\theta)}{\partial \theta_0}, \frac{\partial J(\theta)}{\partial \theta_1}, \dots, \frac{\partial J(\theta)}{\partial \theta_D}\right),$$

  where

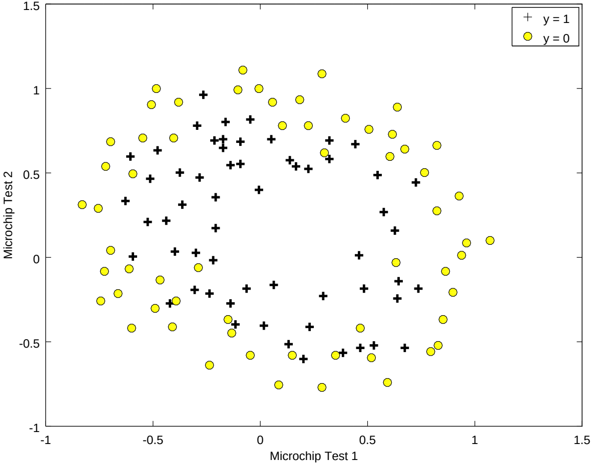
$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{N}\sum_{i=1}^{N}[h_\theta(\mathbf{x}_i) - y_i]x_{i0}$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{N}\sum_{i=1}^{N}[h_\theta(\mathbf{x}_i) - y_i]x_{ij} + \frac{\lambda}{N}\theta_j, \quad \forall j = 1, \dots, D.$$

Microchip Quality

- Predict whether microchips from a fabrication plant passes quality assurance (QA).
- During QA, each microchip goes through various tests to ensure it is functioning correctly.
- We have test results for some microchips on two different tests. From these two tests, we would like to determine whether the microchip should be accepted or rejected.

Microchip Quality – Scatter Plot



Microchip Quality – Data

- It is obvious that our dataset cannot be separated into positive and negative examples by a straight-line through the plot.
- Since logistic regression is only able to find *a linear decision boundary*, a straightforward application of logistic regrssion will not perfom well on this dataset.
- Solution? Use feature mapping technique.

Feature Mapping

- One way to fit the data better is to create more features from each data point.
- For each data point $\mathbf{x} = (x_1, x_2)$, we map the features into all polynomial terms of x_1 and x_2 up to the sixth power:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \Longrightarrow \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \\ x_1^3 \\ \dots \\ x_1 x_2^5 \\ x_2^6 \end{pmatrix}$$

Feature Mapping

- A vector of two features has been transformed to a 28-dimensional vector.
- A logistic regression classifier trained on this higher-dimension feature vector will have a more complex decision boundary and will appear nonlinear when drawn in our 2-dimensional plot.
- Note that while the feature mapping allows us to build a more powerful classifier, it is also more susceptible to overfitting.
- Regularization technique helps us to prevent overfitting problem.

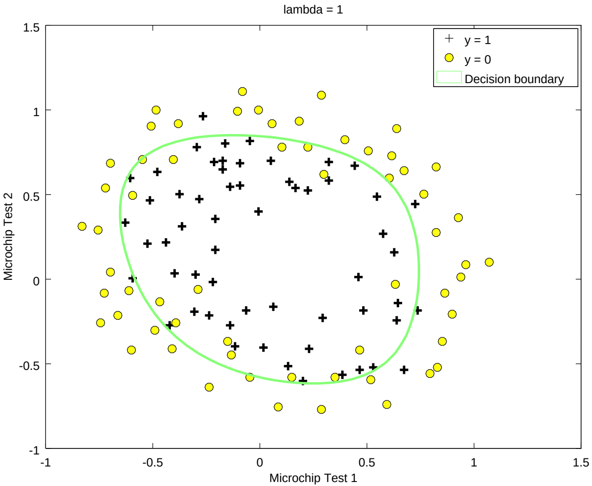
Feature Mapping

Octave/Mathlab implementation of feature mapping:

```
function out = mapFeature(X1, X2)
  degree = 6;
  out = ones(size(X1(:,1)));
  for i = 1:degree
    for j = 0:i
      out(:, end+1) = (X1.^(i-j)).*(X2.^j);
    end
  end
end
```

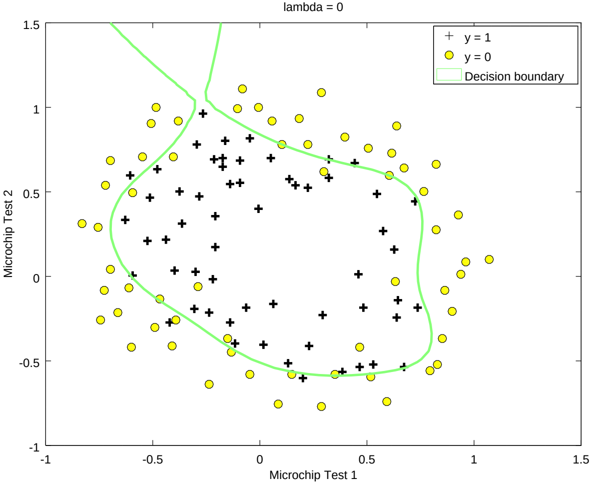
Microchip Quality – Decision Boundary

Training data with decision boundary ($\lambda = 1$)



Microchip Quality – Decision Boundary

No regularization ($\lambda = 0$) – overfitting



Microchip Quality – Decision Boundary

Too much regularization ($\lambda = 100$)