GAUSSIAN DISCRIMINANT ANALYSIS

Yurong Chen*
Hunan University
chenyurong1998@outlook.com

1 Introduction

Algorithms that model $p(y|x)$ directly from the training set are called **discriminative algorithms** [1], such as linear regression:

$$f(y|x,\theta) = \theta_0 + \theta_1 x_1 + ... + \theta_n x_n, \tag{1}$$

or logistic regression:

$$f(y|x,\theta) = \frac{1}{1+e^{-\theta^T x}}. \tag{2}$$

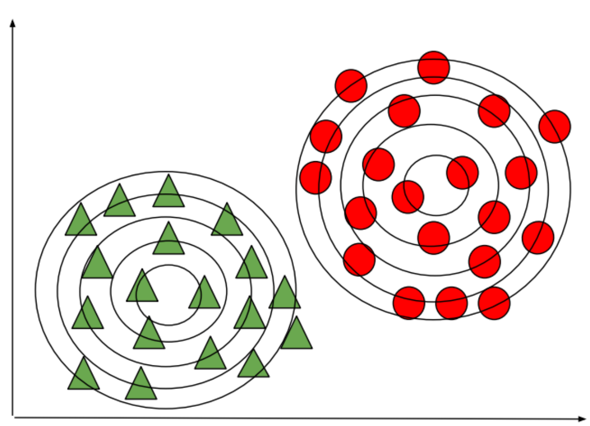
These try to find a decision boundary between different classes during the learning process.

On the other hand, we model $p(x|y)$ and $p(y)$, then classify a new element based on the match against each model. Once we learn the model $p(x|y)$ and $p(y)$, we use Bayes Theorem to derive the $p(y|x)$ as:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}. \tag{3}$$

It is called **generative learning algorithms**.

When we have a classification problem in which the input features are continuous random variable, we can use **Gaussian discriminant analysis** model, as figure shown.



The probability of a prediction in the case of the Generative learning algorithm will be high if it lies near the centre of the contour corresponding to its class and it decreases as we move away from the centre of the contour [2][3].

*https://yurongchen1998.github.io

It assumes $p(x|y)$ is distributed according to a **multivariate normal distribution** and $p(y)$ is distributed according to **Bernoulli**:

$$p(y) = \theta^y(1-\theta)^{(1-y)}; \tag{4}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp(-1/2(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)); \tag{5}$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp(-1/2(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)). \tag{6}$$

Now, we define the log loss function as:

$$l(\theta,\mu_0,\mu_1,\Sigma) = \log\prod_i^m p(x_i|y_i,\theta,\mu_0,\mu_1,\Sigma)p(y_i,\theta). \tag{7}$$

$$l(\theta,\mu_0,\mu_1,\Sigma) = \log\prod_i^m (\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp(-1/2(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)))\theta^y(1-\theta)^{(1-y)}; \tag{8}$$

$$l(\theta,\mu_0,\mu_1,\Sigma) = \sum_i^m [-n/2\log 2\pi - 1/2\log|\Sigma| - 1/2(x-\mu)^T\Sigma^{-1}(x-\mu) + y\log\theta + (1-y)log(1-\theta)]. \tag{9}$$

Take partial derivatives to each parameter to find the maximum likelihood of that parameter:

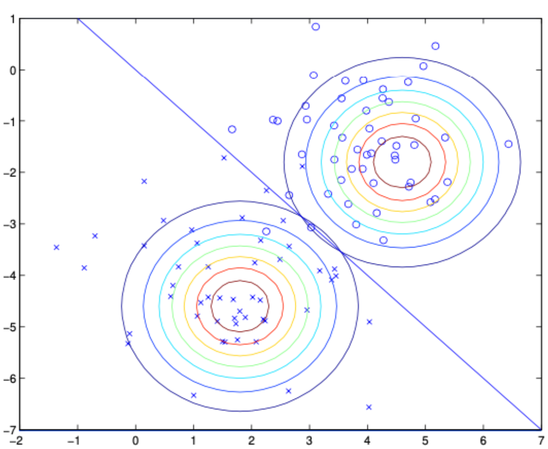
$$\theta = 1/m\sum_i^m 1\ \{y_i = 1\}. \tag{10}$$

$$\mu_0 = \frac{\sum_i^m 1\ \{y_i = 0\}\ x_i}{\sum_i^m 1\ \{y_i = 0\}}. \tag{11}$$

$$\mu_1 = \frac{\sum_i^m 1\ \{y_i = 1\}\ x_i}{\sum_i^m 1\ \{y_i = 1\}}. \tag{12}$$

$$\Sigma = 1/m\sum_i^m (x_i-\mu_k)(x_i-\mu_k)^T\ where k = 1\{y_i = 1\}. \tag{13}$$

Pictorially, what the algorithm is doing can be seen in as follows [3]: Shown in the figure are the training set, as well as



the contours of the two Gaussian distributions that have been fit to the data in each of the two classes. Note that the two Gaussians have contours that are the same shape and orientation, since they share a covariance matrix, but they have different means. Also shown in the figure is the straight line giving the decision boundary at which p(y = 1|x) = 0.5. On one side of the boundary, we'll predict y = 1 to be the most likely outcome, and on the other side, we'll predict y = 0.

References

[1] https://towardsdatascience.com/gaussian-discriminant-analysis-an-example-of-generative-learning-algorithms-2e336ba7aa5c

[2] https://www.geeksforgeeks.org/gaussian-discriminant-analysis/

[3] http://cs229.stanford.edu/notes2020spring/cs229-notes2.pdf