Generative Learning Model: GDA and Naive Bayes

Fangfang Song

June 2018

1 Introduction

In linear regression or logistic regression, we were trying to learn $p(y|x;\theta)$ directly, or trying to learn mapping directly from the space of inputs X to the output, these are called **discriminative learning algorithms**.While,**Generative learning** try to model $p(x|y)$ instead of $p(y|x)$.

After modeling $p(y)$(called the **class prior**) and $p(x|y)$,we then use **Bayes rule** to derive the posterior distribution $p(y|x)$

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{1}$$

If we were calculating $p(y|x)$ in order to make a prediction, we don't actually need to calculate the denominator, since:

$$\begin{aligned} argmax(p(y|x)) &= argmax(\frac{p(x|y)p(y)}{p(x)}) \\ &= argmax(p(x|y)p(x)) \end{aligned} \tag{2}$$

2 Gaussian discriminant analysis

When we have a classification problem in which the input features x are continuous value random variables, we can then use the Gaussian Discriminant Analysis model, which models $p(x|y)$ using a multivariate normal distribution. The model is:

$$y \sim Bernoulli(\phi) \tag{3}$$
$$x|y = 0 \sim N(\mu_0, \Sigma) \tag{4}$$
$$x|y = 1 \sim N(\mu_1, \Sigma) \tag{5}$$
$$\tag{6}$$

Writing out the distributions:

$$p(y) = \phi^y(1-\phi)^{(1-y)} \tag{7}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{(n/2)}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0))$$
$$p(x|y=1) = \frac{1}{(2\pi)^{(n/2)}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_0)) \tag{8}$$

The parameter of the modesl are $\phi, \Sigma, \mu_0, \mu_1$. To estimate the parameter, we use the maximum log-likelihood estimate:

$$\begin{aligned} l(\phi, \mu_0, \mu_1, \Sigma) &= log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) \\ &= log \prod_{i=1}^{m} p(x^{(i)}|y^{(i)}; \phi, \mu, \mu, \Sigma) p(y^{(i)}) \end{aligned} \tag{9}$$

Maximize likelihood estimate of the parameters:

$$l(\phi, \mu_0, \mu_1, \sigma) = log \prod_{i=1}^{m} \frac{1}{(2\pi)^{(n/2)}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))\phi^y(1-\phi)^{(1-y)}$$

$$\begin{aligned} \nabla_\phi l(\phi) &= \nabla_\phi \sum_{i=1}^{m} (y^{(i)} log\phi_y + (1-y^{(i)}) log(1-\phi_y)) \\ &= \sum_{i=1}^{m} y^{(i)} \frac{1}{\phi_y} - (1-y^{(i)}) \frac{1}{1-\phi_y} \end{aligned}$$

Setting $\nabla_\phi = 0$ yield

$$\begin{aligned} 0 &= \sum_{i=1}^{m} (y^i(1-\phi_y) - (1-y^i)\phi_y) \\ &= \sum_{i=1}^{m} y^{(i)} - \sum_{i=1}^{m} \phi_y \end{aligned}$$

So:

$$\phi_y = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}$$

Similarly, Setting $\nabla_{\phi_j|y=0} = 0$ and $\nabla_{\phi_j|y=1} = 0$

2.1 GDA and logistic regression

We can prove that is $p(x|y)$ is multivariate gaussian (with share $\sum$, then $p(y|x)$ necessarily follows a logistic function. While, the converse is not true; i.e. $p(y|x)$ being a logistic function does not imply $p(x|y)$ is multivariate gaussian. This means GDA makes **stronger** modeling assumptions about the data than does logistic regression, and is more data efficient. It turns not that when these modeling assumptions are correct, then GDA will find better fits, and is a better model.In contrast, by making significantly weaker assumptions, logistic regression is also more robust and less sensitive to incorrect modeling assumptions.

3 Naive Bayes

3.1 Introduction

In text classification problems, feature vector x are one hot vector. It is discrete, the number of features equal the number of words in the dictionary and it's huge. Thus it's difficult to write the covariance matrix, so we need to make a very strong assumption. we will assume that x_i are conditionally independent given y, which is called the **Naive Bayes assumptions**.

3.2 Laplace smoothing

Laplace smoothing is a simple change that makes Naive Bayes algorithms works much better, especially for text classification. It is statistically a bad idea to estimate the probability of some event to be zero just because you haven't seen it before in your finite training set. To avoid this, we can use Laplace Smoothing.