CSE422: Artificial Intelligence
Gaussian Discriminant Analysis & Naive Bayes

Rafiad Sadat Shahir

1 Discriminative & Generative Learning Algorithms

The previous learning algorithms model $p(y \mid x;\theta)$, the conditional distribution of y given x. For example, logistic regression modeled $p(y \mid x;\theta)$ as $h_\theta(x) = g(\theta^\top x)$. For a classification problem, an algorithm like logistic regression attempts to find a decision boundary that separates the classes. To classify a new input, the algorithm checks on which side of the decision boundary the input falls.

A different approach is to build a model of the classes. To classify a new input, the algorithm matches the new input against the models of classes. In essence, the algorithm attempts to learn $p(x \mid y)$.

The algorithms that attempt to learn $p(y \mid x)$ are called discriminative learning algorithms, and the algorithms that attempt to learn $p(x \mid y)$ are called generative learning algorithms. If $y \in \{0,1\}$, $p(x \mid y=0)$ models the distribution of the features of class 0, and $p(x \mid y=1)$ models the distribution of the features of class 1.

Upon modeling the likelihood $p(x \mid y)$ and the prior probability $p(y)$, the Bayes rule can be used to derive the posterior probability $p(y \mid x)$:

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}$$

In order to make a prediction, the denominator is insignificant since,

$$\arg\max_y p(y \mid x) = \arg\max_y \frac{p(x \mid y)p(y)}{p(x)} = \arg\max_y p(x \mid y)p(y)$$

2 Gaussian Discriminant Analysis

In the Gaussian discriminant analysis (GDA) model, the assumption is that $p(x \mid y)$ is multivariate normally distributed.

2.1 Multivariate Gaussian Distribution

The multivariate normal distribution (also called the multivariate Gaussian distribution) in d dimensions is parameterized by a mean vector $\mu \in \mathbb{R}^d$ and a covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$. The density $\mathcal{N}(\mu,\Sigma)$ is expressed as

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}$$

2.2 GDA Model

For a binary classification problem where the input features x are continuous random variables, the GDA model is used that models $p(x \mid y)$ using a multivariate normal distribution. The assumptions for the model are as follows.

$$\begin{aligned} y &\sim \mathcal{B}(\phi) \\ x \mid y = 0 &\sim \mathcal{N}(\mu_0, \Sigma) \\ x \mid y = 1 &\sim \mathcal{N}(\mu_1, \Sigma) \end{aligned}$$

Therefore,

$$\begin{aligned} p(y) &= \phi^y (1-\phi)^{1-y} \\ p(x \mid y = 0) &= \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_0)^\top \Sigma^{-1} (x-\mu_0)} \\ p(x \mid y = 1) &= \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1} (x-\mu_1)} \end{aligned}$$

Here, the parameters of our model are ϕ, μ_0, μ_1, and Σ. Though there are two different mean vectors μ_0 and μ_1, the model is usually applied using only one covariance matrix Σ.) The log likelihood of the data can be expressed as

$$\ell(\phi, \mu_0, \mu_1, \Sigma) = \log \prod_{i=1}^{n} p(x^{(i)}, y^{(i)}; \phi, \mu_o, \mu_1, \Sigma)$$

By maximizing ℓ with respect to the parameters, we get

$$\begin{aligned} \phi_y &= \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}{n} \\ \mu_0 &= \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}} \\ \mu_1 &= \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}} \\ \Sigma &= \frac{1}{n} \sum_{i=1}^{n} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^\top \end{aligned}$$

2.3 GDA & Logistic Regression

The quantity $p(y = 1 \mid x; \phi, \mu_0, \mu_1, \Sigma)$ can be expressed as a function of x.

$$p(y = 1 \mid x; \phi, \Sigma, \mu_0, \mu_1) = \frac{1}{1 + e^{-\theta^\top x}}$$

where θ is some appropriate function of $\phi, \Sigma, \mu_0, \mu_1$. This is exactly the form used in logistic regression to model $p(y = 1 \mid x)$.

If $p(x \mid y)$ is a multivariate Gaussian with shared Σ, $p(y \mid x)$ necessarily follows a logistic function. However, $p(y \mid x)$ being a logistic function does not imply that $p(x \mid y)$ is a multivariate Gaussian. Thus, GDA makes stronger modeling assumptions about the data than logistic regression. When the modeling assumptions are correct, GDA will find better fits to the data even for small training set sizes.

In contrast, by making significantly weaker assumptions, the logistic regression is more robust. There are many different sets of assumptions that would lead to $p(y \mid x)$ taking the form of a logistic function. For example, if $x \mid y = 0 \sim \text{Poisson}(\lambda_0)$ and $x \mid y = 1 \sim \text{Poisson}(\lambda_1)$, $p(y \mid x)$ will be logistic. Thus, logistic regression will also work well on Poisson data. However, if GDA is used on such data, it may not perform well.

3 Naive Bayes

In GDA, the feature vectors x were continuous-valued vectors. Naive Bayes(NB) is a learning algorithm in which the feature vectors x were discrete-valued vectors.

3.1 NB Model

To model $p(x \mid y)$, the assumption is that x_j's are conditionally independent given y. This assumption is called the Naive Bayes assumption. Thus, we have

$$\begin{aligned}
&p(x_1, x_2, x_3, \ldots, x_d \mid y) \\
&= p(x_1 \mid y)p(x_2 \mid y, x_1)p(x_3 \mid y, x_1, x_2) \ldots p(x_d \mid y, x_1, x_2, x_3, \ldots, x_{d-1}) \\
&= p(x_1 \mid y)p(x_2 \mid y)p(x_3 \mid y) \ldots p(x_d \mid y) \\
&= \prod_{j=1}^{d} p(x_j \mid y)
\end{aligned}$$

Moreover,

$$\begin{aligned} y &\sim \mathcal{B}(\phi_y) \\ x_j \mid y = 0 &\sim \mathcal{B}(\phi_{j|y=0}) \\ x_j \mid y = 1 &\sim \mathcal{B}(\phi_{j|y=1}) \end{aligned}$$

The likelihood of the data is expressed as

$$\mathcal{L}(\phi_y, \phi_{j|y=0}, \phi_{j|y=1}) = \prod_{i=1}^{n} p(x^{(i)}, y^{(i)})$$

The maximum likelihood estimates are as follows:

$$\begin{aligned} \phi_y &= \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}{n} \\ \phi_{j|y=0} &= \frac{\sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}} \\ \phi_{j|y=1} &= \frac{\sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}} \end{aligned}$$

Having fitted all the parameters, an inference is made by calculating the following:

$$\arg\max_y p(y \mid x) = \arg\max_y p(x \mid y)p(y) = \arg\max_y \left(\prod_{j=1}^{d} p(x_j \mid y) \right) p(y)$$

3.2 Multinomial Naive Bayes

The described Naive Bayes algorithm handles the features with binary values. To generalize to where $x_j \in \{1, 2, \ldots, k_j\}$, we can model $p(x_j \mid y)$ as a multinomial rather than as Bernoulli.

3.3 Gaussian Naive Bayes

For features with continuous values, it is quite common to discretize the features. However, another model called Gaussian Naive Bayes (GNB) can also be used. Gaussian Naive Bayes assumes that the likelihood of the feature $p(x_j \mid y)$ follows the Gaussian distribution. Thus, $p(x_j \mid y)$ is expressed as:

$$p(x_j \mid y = k) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2\sigma_k^2}(x_j - \mu_k)^2}$$

where,

$$\mu_k = \frac{\sum\limits_{i=1}^{n} 1\{y^{(i)} = k\} x_j^{(i)}}{n}$$

$$\sigma_k^2 = \frac{\sum\limits_{i=1}^{n} 1\{y^{(i)} = k\}(x_j^{(i)} - \mu_k)^2}{n}$$

3.4 Laplace smoothing

Assume that $\sum\limits_{i=1}^{n} 1\{x_n^{(i)} = 1\} = 0$. Thus,

$$\phi_{n|y=1} = \frac{\sum\limits_{i=1}^{n} 1\{x_n^{(i)} = 1 \wedge y^{(i)} = 1\}}{\sum\limits_{i=1}^{n} 1\{y^{(i)} = 1\}} = 0$$

$$\phi_{n|y=0} = \frac{\sum\limits_{i=1}^{n} 1\{x_n^{(i)} = 1 \wedge y^{(i)} = 0\}}{\sum\limits_{i=1}^{n} 1\{y^{(i)} = 0\}} = 0,$$

Therefore,

$$\prod_{j=1}^{d} p(x_j \mid y = 1) p(y = 1) = \prod_{j=1}^{d} p(x_j \mid y = 0) p(y = 0) = 0$$

This is due to the fact that $\prod_{j=1}^{d} p(x_j \mid y)$ includes the term $p(x_n \mid y) = 0$. Hence, for an unseen event in the dataset, the algorithm estimates the probability of that event to be zero.

Consider a multinomial random variable z that takes the values in $\{1, 2, \ldots, k\}$. The multinomial can be parameterize with $\phi_j = p(z = j)$. Given a set of n independent observations $\{z^{(1)}, z^{(2)}, \ldots, z^{(n)}\}$, the maximum likelihood estimates are given as follows:

$$\phi_j = \frac{\sum\limits_{i=1}^{n} 1\{z^{(i)} = j\}}{n}$$

To solve the previously mentioned problem, Laplace smoothing can be used that replaces the mentioned estimate with

$$\phi_j = \frac{1 + \sum\limits_{i=1}^{n} 1\{z^{(i)} = j\}}{k + n}$$

Here, 1 is added to the numerator and k to the denominator to ensure that $\sum\limits_{j=1}^{k} \phi_j = 1$ still holds. For the NB model, we obtain the following estimates with Laplace smoothing:

$$\phi_{j|y=1} = \frac{1 + \sum\limits_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{2 + \sum\limits_{i=1}^{n} 1\{y^{(i)} = 1\}}$$

$$\phi_{j|y=0} = \frac{1 + \sum\limits_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{2 + \sum\limits_{i=1}^{n} 1\{y^{(i)} = 0\}}$$