Gaussian Discriminant Analysis

Uzair Ahmad

Introduction

Previously, we've discussed the Naïve Bayes Model, where components of the input vector x_i are discrete-valued. When features of x_i are continuous-valued random variables, instead of using the Gaussian Naïve Bayes Model, we could use the Gaussian Discriminant Analysis Model (GDA), in which we assume that, given the label y_i , the input x_i follows a multivariate Gaussian distribution. The GDA Model is also a generative model that can be applied to classification tasks.

GDA model parameters

GDA is also a generative model, thus we have $P(y_i|\mathbf{x_i}) = \frac{P(\mathbf{x_i}|y_i)P(y_i)}{P(\mathbf{x_i})}$. Since the task is to classify an example, here we could make an assumption of the label such that y_i follows a Bernoulli distribution specified by parameter π. $y_i \sim Bernoulli(\pi)$.

Then, given the fact that the label is known, we can make another assumption about the input variables such that $x_i|y_i = 1$ and $x_i|y_i = 0$ follow the multivariate Gaussian distribution specified by (μ_0, Σ) and (μ_1, Σ), respectively.

$x_i|y_i = 0 \sim N(\mu_1, \Sigma)$

$xi|yi = 1 \sim N(\mu_0, \Sigma)$

Given the distribution parameters are known, we have

$$P(y_i) = \pi^{y_i}(1-\pi)^{1-y_i}$$

$$P(\mathbf{x_i}|\mathbf{y_i} = \mathbf{0}) = \frac{\mathbf{1}}{(\mathbf{2\pi})^{\frac{\mathbf{m+1}}{\mathbf{2}}}|\mathbf{\Sigma}|^{\frac{\mathbf{1}}{\mathbf{2}}}}\mathbf{exp}(-\tfrac{\mathbf{1}}{\mathbf{2}}(\mathbf{x_i} - \boldsymbol{\mu}_\mathbf{0})^\mathbf{T}\mathbf{\Sigma}^{-\mathbf{1}}(\mathbf{x_i} - \boldsymbol{\mu}_\mathbf{0}))$$

$$P(\mathbf{x_i}|\mathbf{y_i} = \mathbf{1}) = \frac{\mathbf{1}}{(\mathbf{2\pi})^{\frac{\mathbf{m+1}}{\mathbf{2}}}|\mathbf{\Sigma}|^{\frac{\mathbf{1}}{\mathbf{2}}}}\mathbf{exp}(-\tfrac{\mathbf{1}}{\mathbf{2}}(\mathbf{x_i} - \boldsymbol{\mu}_\mathbf{1})^\mathbf{T}\mathbf{\Sigma}^{-\mathbf{1}}(\mathbf{x_i} - \boldsymbol{\mu}_\mathbf{1}))$$

where $\mu_0, \mu_1 \in R^m$ is the mean vector; $\mathbf{\Sigma} \in \mathbf{R}^{\mathbf{m \times m}}$ is the covariance matrix; and the $|\mathbf{\Sigma}| \in \mathbf{R}$ is the determinant of $\mathbf{\Sigma}$.

Notice that computing the probability of x under each class conditional density is equivalent to calculating the distance from x to the center of each class, μ_i, using Mahalanobis distance.

$d = \sqrt{(x_i - \mu_1)^T\Sigma^{-1}(x_i - \mu_1)}$

Therefore, GDA can be thought of as a nearest centroids classifier.

Maximum-Likelihood of GDA Parameters

The priors

In Gaussian Discriminant Analysis (GDA), given a dataset with binary labels y where y can be either 0 or 1, we want to determine the prior probability π that $y = 1$.

Formally, π is defined as: $\pi = p(y = 1)$

Likelihood Function

Given our training set of m examples with labels, the likelihood function, based on the Bernoulli distribution of y, is:
$L(\pi) = \prod_{i=1}^{m} \pi^{y^{(i)}} (1-\pi)^{1-y^{(i)}}$

Log-Likelihood Function

Taking the logarithm of our likelihood function:
$l(\pi) = \sum_{i=1}^{m} \left[y^{(i)} \log(\pi) + (1-y^{(i)}) \log(1-\pi) \right]$

Deriving w.r.t. (\pi)

To find the value of π that maximizes the log-likelihood, we differentiate $l(\pi)$ with respect to π.

Differentiating:
$\frac{dl(\pi)}{d\pi} = \sum_{i=1}^{m} \left[\frac{y^{(i)}}{\pi} - \frac{1-y^{(i)}}{1-\pi} \right]$

Setting the above to zero and rearranging terms:
$\sum_{i=1}^{m} y^{(i)} = \pi m$

From which:
$\pi = \frac{\sum_{i=1}^{m} y^{(i)}}{m}$

This is the MLE for π. The result essentially means that the best estimate for the probability of $y = 1$ in our dataset is simply the fraction of the training examples for which $y = 1$.

The μs

In Gaussian Discriminant Analysis (GDA), given the assumption of a multivariate normal distribution for $p(x|y)$, we want to find μ_1 such that the likelihood of observing our data is maximized.

Recall the term from the log-likelihood function for the Gaussian distribution that is dependent on μ:

$-(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1} (x^{(i)} - \mu_{y^{(i)}})$

This term determines the squared difference between our data point $x^{(i)}$ and the mean μ of the class it belongs to, scaled by the inverse covariance matrix Σ^{-1}.

Deriving w.r.t. μ_1

To find the MLE of μ_1, we need to differentiate the above term with respect to μ_1 and then set the derivative equal to zero.

The term from the log-likelihood relevant to μ_1 (i.e., for the instances where $y^{(i)} = 1$) is:

$-(x^{(i)} - \mu_1)^T \Sigma^{-1} (x^{(i)} - \mu_1)$

Differentiating this with respect to μ_1, and after some simplifications, you'll find that the derivative is proportional to:

$x^{(i)} - \mu_1$

When we set this derivative equal to zero (to find the maximum likelihood), it implies:

$x^{(i)} = \mu_1$

Summing Over All Instances

The above relation holds for a single instance. For all instances where $y^{(i)} = 1$:

$\sum_{i:y^{(i)}=1} x^{(i)} = \sum_{i:y^{(i)}=1} \mu_1$

Since the right side is merely summing up the same value μ_1 for all instances where $y^{(i)} = 1$, we can simplify this to:

$\mu_1 = \frac{\sum_{i:y^{(i)}=1} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}}$

This equation essentially states that μ_1 (mean for class 1) is the average of all training examples that belong to class 1.

The Σ

Now, Let's derive the Maximum Likelihood Estimate (MLE) for the covariance matrix Σ in the context of Gaussian Discriminant Analysis. We have the multivariate Gaussian distribution for (p(x|y)), where the probability of data point (x) given its class label (y) is defined as:

$$p(x|y) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_y)^T \Sigma^{-1}(x - \mu_y)\right)$$

Here, n is the dimensionality of x, μ_y is the mean of the Gaussian for class y, and Σ is the covariance matrix shared by all classes.

Objective

Our goal is to find Σ such that the likelihood of observing our data is maximized.

Likelihood Function

The log-likelihood function based on the above Gaussian distribution and given m training examples and n number of features/dimensions in each sampleis:

$$l(\Sigma) = \sum_{i=1}^{m} \log\left(\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1}(x^{(i)} - \mu_{y^{(i)}})\right)\right)$$

Deriving w.r.t. Σ

The most challenging part of the derivation is the term:

$$(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1}(x^{(i)} - \mu_{y^{(i)}})$$

Differentiating the log-likelihood with respect to Σ and simplifying, we eventually get an equation for Σ that looks like:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$$

This equation states that the MLE for Σ is essentially an average of the outer product of the centered data points (centered using their respective class means).

To put it intuitively, this formulation captures the average spread of the data around their class means across all classes. The resulting Σ will be the covariance matrix that maximizes the likelihood of observing the given data under the multivariate Gaussian distribution assumption in GDA.

Training phase

According to the assumptions we've made above, GDA has the following parameters.

- π, which specifies $P(y_i)$;
- μ_0, Σ which specifies $P(\mathbf{x_i}|y_i = 0)$;
- μ_1, Σ which specifies $P(\mathbf{x_i}|y_i = 1)$;

Thus, we could write down the log-likelihood of the data $l(\pi, \mu_0, \mu_1, \Sigma) = ln \prod_{i=1}^N P(\mathbf{x_i}|y_i)P(y_i)$. By Maximizing l, we could get the optimal parameters as followings

$$\begin{aligned}
\pi^* &= \frac{\sum_{i=1}^N \mathbb{1}(yi=1)}{N} \\
\mu_0^* &= \frac{\sum_{i=1}^N \mathbf{x_i}\mathbb{1}(yi=0)}{\sum_{i=1}^N \mathbb{1}(yi=0)} \\
\mu_1^* &= \frac{\sum_{i=1}^N \mathbf{x_i}\mathbb{1}(y_i=1)}{\sum_{i=1}^N \mathbb{1}(y_i=1)} \\
\sum_{y_i}^* &= \frac{\sum_{i=1}^N (\mathbf{x_i}-\mu_{y_i})(\mathbf{x_i}-\mu_{y_i})^T}{m}
\end{aligned}$$

where $\mathbb{1}(y_i = 0)$ is the indicator function such that if $y_i = 0$,the $\mathbb{1}(yi = 0)$ outputs 1, otherwise 0.

Prediction phase

After a model is trained, we can make prediction on the label of a given data such that $y_i = argmax_{y_i} P(\mathbf{x_i}|y_i, \theta)P(y_i)$.

If class priors are uniform, then the test data point can be classified finding

$$P(x|y_i, \theta) = argmin_{y_i} P(\mathbf{x}|y_i) = (\mathbf{x} - \mu_i)^T \Sigma_{y_i}^{-1} (\mathbf{x} - \mu_i)$$

Here we assume that the covariance is common among all classes. In case each class has a different covariance, the resulting boundary will be quadratic also known as **Quadratic Discriminant Analysis**.

```python
class GDA():
    def __init__(self):
        self.pi = None
        self.mu0 = None
        self.mu1 = None
        self.sigma = None

    def fit(self, x, y):
        self.pi = np.mean(y)
        self.mu0 = np.mean(x[y[:,0]==0], axis=0)
        # centroid of class 0
        self.mu1 = np.mean(x[y[:,0]==1], axis=0)
        # centroid of class 1

        n_x = x[y[:,0] == 0] - self.mu0
        p_x = x[y[:,0] == 1] - self.mu1

        self.sigma = ((n_x.T).dot(n_x) + (p_x.T).dot(p_x))/x.shape[0]
        self.sigma_inv = np.linalg.inv(self.sigma)

    def predict(self, x):
        p0 = np.sum(np.dot((x-self.mu0),self.sigma_inv)*(x-self.mu0),axis=1)*self.pi
        p1 = np.sum(np.dot((x-self.mu1),self.sigma_inv)*(x-self.mu1),axis=1)*self.pi
        return p1 >= p0
```