ML Fundamentals: Generative Learning Algorithms

February 2, 2021

1 Generative Learning algorithms

There are algorithms which directly models $P(y|x)$ ii.e. logistic regression or linear regression. Given the training data, these algorithms try to find a decision boundaries. These algorithms are called **Discriminative Algorithms**. On the other hand **Generative Algorithms** tries to model $P(x|y)$ and $p(y)$ given training data. And, using Baye's rule, they classify data points into either classes.

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

Where, $p(y)$ is called class priors, and $p(y|x)$ is class conditional distribution. Given the data point, decision can be made as follow:

$$\begin{aligned}\arg\max_y p(y \mid x) &= \arg\max_y \frac{p(x \mid y)p(y)}{p(x)} \\ &= \arg\max_y p(x \mid y)p(y)\end{aligned}$$

1.1 Gaussian discriminant analysis

We will assume $p(x|y)$ is distributed according multivariate normal distribution, with parameters $\mu \in R^n$ and $\Sigma \in R^{nxn}$, Where co-variance matrix is symmatric and positive semidefinite. Probability of x given y, can be written as,

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

1.2 The Gaussian Discriminant Analysis model

For the classification problem, where input x are continuous valued random variable, we can use GDA (Gaussian Discriminant Analysis) model, which models $p(x|y)$ using a multi-variate normal distribution. The model is,

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

And, writing these probabilities would give,

$$\begin{aligned} p(y) &= \phi^y(1-\phi)^{1-y} \\ p(x \mid y=0) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right) \\ p(x \mid y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\end{aligned}$$

And the parameters of the models are $\phi, \mu_0, \mu_1\Sigma$

Likelihood of those parameters can be written as,

$$\begin{aligned}\ell(\phi, \mu_0, \mu_1, \Sigma) &= \log\prod_{i=1}^m p\left(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma\right) \\ &= \log\prod_{i=1}^m p\left(x^{(i)} \mid y^{(i)}; \mu_0, \mu_1, \Sigma\right) p\left(y^{(i)}; \phi\right)\end{aligned}$$

And, with maximum likelihood estimation (i.e. taking derivative, and assigning to zero), we get the following values.

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}$$
$$\mu_0 = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\} x^{(i)}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}}$$
$$\mu_1 = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\} x^{(i)}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}$$
$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)} - \mu_{y^{(i)}}\right)\left(x^{(i)} - \mu_{y^{(i)}}\right)^T$$

GDA model with $p(x|y)$ following mutlivariate guassian distribution, would give a sigmoid decision boundary, like the one with logistic regression modeling $p(y|x)$ directly. However converse is not true (as poisson would also lead to sigmod decision boundary).

GDA makes stronger data assumptions, can can work with lesser data when assumptions are true.

1.3 Naive Bayes

Naive bayes assumes each x_i are conditionally independent given y. Hence, following would be correct derivation,
$p(x_1, \ldots, x_{50000} \mid y)$
$= p(x_1 \mid y)\, p(x_2 \mid y, x_1)\, p(x_3 \mid y, x_1, x_2) \cdots p(x_{50000} \mid y, x_1, \ldots, x_{49999})$
$= p(x_1 \mid y)\, p(x_2 \mid y)\, p(x_3 \mid y) \cdots p(x_{50000} \mid y)$
$= \prod_{i=1}^{n} p(x_i \mid y)$
We can write the joint likelihood of the data as follows,

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

Maximizing these parameters over training data would give MLE as follow:

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}$$
$$\phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}}$$
$$\phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}$$

1.4 Laplace Smoothing

If there are m class, we can initialise each class with one sample, to avoid data sparsity problem, hence
Before Laplace:

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\}}{m}$$

After applying Laplace theorem:

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\} + 1}{m + k}$$

1.5 Multivariate Bernoulli Event Model vs Multinomial Event Model

In the specific context of text classification, Naive Bayes as presented is a method of type Multivariate Bernoulli Event Model, where in each token of the vocabulary is considered as a coin toss. That is, person (spammer or non-spammer) sending email runs through dictionary, deciding whether to include each word i in that email independently according to the probabilities $p(x_i = 1 \mid y) = \phi_{i|y}$. And, hence the probability of a message was given by $p(y)\prod_{i=1}^{n} p(x_i \mid y)$. MLE for Multivariate Bernoulli Event model would give following estimation:

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}$$
$$\phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}}$$
$$\phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}$$

Other way to model this would be to consider it multinomial events of length of input text, where each event from same multinomial distribution over the tokens in the vocabulary. So, to compute the probability $p(y)\prod_{i=1}^{n} p\left(x_i \mid y\right)$, where n is the numbers of tokens in the input text. MLE for Multinomial Event Model would give following estimation:

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\} n_i}$$
$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\} n_i}$$
$$\phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}.$$