Bayesian Models

Bayesian models are based on bayes rule. Bayesians always assume that our training data come from some distribution. And given prior probability of parameters $p(\theta)$, we hope to use bayesian models to compute its posterior probability $p(\theta|\mathbf{x})$ so as to help us estimate the true data distribution.

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}$$

Naive Bayes Classifier

Learning a Naive Bayes classifier is just a matter of counting how many times each attribute co-occurs with each class. There're two assumptions in naive bayes classifier:

- All **variables** in the dataset are independent.
- The distribution of each attribute somehow we know, but we don't know its parameters exactly.

Naive bayes classifier is a kind of lazy learning like KNN.

1. Model Setup

- What we know

$$\begin{aligned} p(Y=c|\mathbf{x}) &= \frac{p(\mathbf{x}|Y=c)p(Y=c)}{p(\mathbf{x})} \\ &= \frac{p(Y=c)}{p(\mathbf{x})}p(\mathbf{x}|Y=c) \\ &= \frac{p(Y=c)}{p(\mathbf{x})}\Pi_{i=1}^{d}p(x_i|Y=c) \end{aligned}$$

where x_i is the d^{th} feature of $\mathbf{x}$.

- What we want to know: $argmax_c\ p(Y=c|\mathbf{x})$

Since for all classes, $p(\mathbf{x})$ are same, we could know that

$$argmax_c\ p(Y=c|\mathbf{x}) = argmax_c\ p(Y=c)\Pi_{i=1}^{d}p(x_i|Y=c)$$

Usually, we would use just the frequency to estimate the prior probability. That means $p(Y=c) = \frac{D_c}{D}$ where D is the numbers of data and D_c is the numbers of data which are in c class.

As for the likelihood $p(x_i|Y=c)$, if the feature is qualitative, frequency is also used to estimate it.

$$p(x_i|Y=c) = \frac{D_{x_i,c}}{D_c}$$

But if the feature is quantitative, we would use the assumed probability distribution function to estimate it. For example, we assume that the feature is distributed as gaussian $p(x_i|Y=c) \sim N(\mu_{c,i}, \sigma^2_{c,i})$

$$p(x_i|Y=c) = \frac{1}{\sqrt{2\pi\sigma_{c,i}}} e^{-\frac{(x_i-\mu_{c,i})^2}{2\sigma^2_{c,i}}}$$

2. Smoothing Trick

In high dimension, chances are that we could have the curse of dimensionality issue. In this case, **for qualitative variables**, $D_{x_i,c}$ could be 0 which could make wrong decisions. In order to avoid this issue, we usually use smoothing trick, such as laplacian correction before estimation. The correction goes like

$$\hat{p}(Y=c) = \frac{D_c+1}{D+N}$$
$$p(x_i|Y=c) = \frac{D_{x_i,c}+1}{D_c+N_i}$$

Where N is the number of classes of dataset and N_i is the number of possible values of i^{th} variable.

Ref: *https://towardsdatascience.com/all-about-naive-bayes-8e13cef044cf*