Chapter 7 Bayesian Classifier

Jinghao.Zhao

Abstract

This is the notes for bayesian classifier, section 7.1 to 7.3, including formula and key points.

1 Bayesian Decision Theory

Bayesian decision classification works in probability framework. Some concept:

1. Prior probability(先验概率): Probability based on experience. eg: throwing a coin and have 50% possibility for each side.

2. Posterior probability(后验概率): Probability of outcomes of an experiment after it has been performed and a certain event has occurred.即某事件发生后, 该事件由某个因素引起的可能性, 是一种条件概率.

3. Likelihood(似然): Likelihood is a function of the parameters of a statistical model, given specific observed data.

The relation between probabilities above is $P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$ means the rate that A occurred caused by B. In classification, based on posterior probability $P(c_i|x)$, and the loss of misclassify a sample from i to j λ_{ij}. The expected loss for classified a sample x to c_j (The conditional risk) is

$$R(c_i|x) = \sum_{j=1}^{N} \lambda_{ij} P(c_j|x). \tag{1}$$

For (7.5), based on (7.1) and (7.4)

$$R(c_i|x) = \sum_{j=1}^{N} \lambda_{ij} P(c_j|x) = P(c_1|x) + P(c_2|x) + ... + P(c_N|x)(i \neq j) = 1 - P(c_i|x) \tag{2}$$

Then we can get (7.5). Based on Bayes' theorem $P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)}$, the classification problem will be transfered into calculate $P(x|c)$ and P(c) based on samples.

JinghaoZ

2 Naive Bayes Classifier

Based on section 1, Getting P(c) is easy since it is the proportion of each difference class of samples. But $P(x|c)$ is difficult because it is joint probability. Naive bayes classifier solved this problem by attribute conditional independence assumption(属性条件独立性假设,即每个属性独立地影响分类结果) 朴素贝叶斯分类器是基于属性条件独立性假设，将后验概率$p(c|x)$ 重写为每个属性的条件概率的连乘形式， 即

$$h_{nb}(x) = \arg\max_{c \in y} P(c) \prod_{i=1}^{d} p(x_i|c), \tag{3}$$

根据大数定律，类c的先验概率为c占整个数据集的比例，

$$P(c) = \frac{|D_c|}{|D|} \tag{4}$$

对于离散属性，条件概率$p(x_i|c)$ 可以表示为

$$P(x_i|c) = \frac{|D_{c,x_i}|}{|D_c|} \tag{5}$$

其中D_{c,x_i}表示D_c在第i个属性上取值为x_i的样本集合。而对于连续属性，考虑概率密度函数

$$P(x_i|c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} exp(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}) \tag{6}$$

通过计算(1)式可以得到概率最大的类。在实践中，通常采用取对数形式来避免连乘下溢，而且使用拉普拉斯修正防止某个类别中某个属性的值未出现。

$$P(c) = \frac{|\ D_c + 1\ |}{|\ D_c + N\ |} \tag{7}$$

where N is the number of different classes, and

$$P(x_i|c) = \frac{|D_{c,x_i} + 1|}{|D_c + N_i|} \tag{8}$$

Where N_i is the different value of class i.

3 Maximum Likelihood Estimation

Used when probability $p(x|c)$ of class C has a definite form and only defined by parameter θ_c. The likelihood of D_c is

$$P(D_c|\theta_c) = \prod_{x \in D_c} P(x|\theta_c) \tag{9}$$

JinghaoZ

And maximum likelihood is try to find a θ_c that get maximum of (9). To avoid the underflow caused by continuous multiplication, we often use log-likelihood(对数似然)

$$\hat{\theta_c} = \arg\max_{\theta_c} LL(\theta_c),\ where\ LL(\theta_c) = \log P(D_c|\theta_c) = \sum_{x \in D_c} \log P(x|\theta_c) \quad (10)$$

JinghaoZ