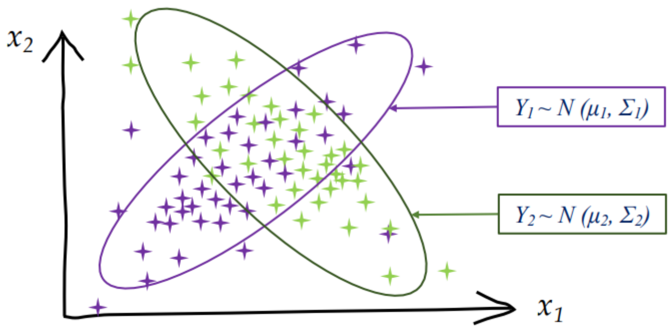
贝叶斯分类

决策树存在的问题

决策树无法解决以下情况（无法进行线性的递归的划分），但是明显两类数据存在一定的概率分布特征。



Bayes Rules

$$P(C_i|x) = \frac{p(x|C_i)P(C_i)}{p(x)}, \qquad i = 1,2$$

$P(C_i)$: prior probability of C_i (before observing x)

$p(C_i|x)$: posterior probability of C_i (after observing x)

$p(x|C_i)$: probability of x given C_i (likelihood)

$p(x)$: probability that x will be observed (evidence)

$$p(x) = P(C_1)p(x|C_1) + P(C_2)p(x|C_2)$$

$$P(\text{error}|x) = \begin{cases} P(C_1|x) & \text{if we decide } C_2 \\ P(C_2|x) & \text{if we decide } C_1 \end{cases}$$

- the average probability of error:

  $P(\text{error}) = \int_{-\infty}^{\infty} P(\text{error}|x)p(x)dx$ This is minimized if for every x we ensure that P(error|x) is as small as possible

- the decision rule:

  Classify x into C_1 if $P(C_1|x) > P(C_2|x)$ $\quad P(\text{error}|x) = \min(P(C_1|x), P(C_2|x))$

- equivalently, classify x into C_1 if

  $$\frac{p(x|C_1)p(C_1)}{p(x)} > \frac{p(x|C_2)p(C_2)}{p(x)}$$

  $$p(x|C_1)p(C_1) > p(x|C_2)p(C_2)$$

Special Cases

对于$\boldsymbol{p}(x \mid C_1)P(C_1) > \boldsymbol{p}(x \mid C_2)P(C_2)$

有两种特殊情况：

- $p(x \mid C_1) = p(x \mid C_2)$:

  决策完全依赖于先验概率$P(C_1)$和$P(C_2)$

- $P(C_1) = P(C_2)$:

  决策完全依赖于条件概率$p(x|C_i)$

General Cases

$$
\begin{aligned}
P\left(C_i \mid \mathbf{x}\right) &= \frac{p\left(\mathbf{x} \mid C_i\right) P\left(C_i\right)}{p(\mathbf{x})} \\
&= \frac{p\left(\mathbf{x} \mid C_i\right) P\left(C_i\right)}{\sum_{k=1}^{K} p\left(\mathbf{x} \mid C_k\right) P\left(C_k\right)}
\end{aligned}
$$

Choose C_i if $P\left(C_i \mid \mathbf{x}\right)=\max_k P\left(C_k \mid \mathbf{x}\right)$

Bayes Network

Input:

- a data sample x = (x$_1$, x$_2$,…,x$_d$)
- a fixed set of classes $C = \{C_1,\dots,C_j\}$.

Output:

- the most probable class c ∈ C:

$$
\begin{aligned}
c_{\text{MAP}} &= \arg\max_{c \in C} P(c|x) \\
&= \arg\max_{c \in C} \frac{P(x|c)P(c)}{P(x)} \\
&= \arg\max_{c \in C} P(x|c)P(c) \\
&= \arg\max_{c \in C} p(x_1, x_2, \dots, x_d|c)\, P(c)
\end{aligned}
$$

对于$p(c)$比较容易求得，只需通过计数即可。但是对于$p(x_1, x_2, \dots x_d|c)$而言，就会非常复杂。

朴素贝叶斯

朴素贝叶斯假设$x_1, x_2, \dots x_d$之间是独立的，即$p(x_1, x_2, \dots x_d|c) = p(x_1|c)p(x_2|c)\dots p(x_d|c)$

这样复杂的计算就得到了化简：

From:

$$
c_{\mathrm{MAP}}=\arg\max_{c \in C} p\left(x_1, x_2, \ldots, x_{\mathrm{d}} \mid c\right) \mathrm{P}(c)
$$

To:

$$
c_{\mathrm{NB}}=\arg\max_{c \in C} P(c) \prod_{i=1}^{d} p\left(x_i \mid c\right)
$$

$p(x_i|c)$可以通过以下方式估算：

$$
\widehat{P}\left(x_i \mid c\right) \leftarrow \frac{\operatorname{count}\left(x_i, c\right)}{\sum_{x \in |x|} \operatorname{count}(x, c)}
$$

处理Zero Counts问题

注意到，对于

$$
c_{\mathrm{NB}}=\arg\max_{c \in C} P(c) \prod_{i=1}^{d} p\left(x_i \mid c\right)
$$

只要有一项$p(x_i|c) = 0$，那么整体就会等于0。注意到，这是因为我们在通过计数的方式估算$p(x_i|c)$，所以很容易出现数据集中在给定c的情况下，x_i出现的次数为零的情况。

所以我们可以使用***Laplace Smoothing***:

$$\hat{P}\left(x_i \mid c\right) \leftarrow \frac{\operatorname{count}\left(x_i, c\right)+1}{\sum_{x \in |x|}(\operatorname{count}(x, c)+1)}$$

本质上就是在之前的式子上下同时加1做了一个近似，但是这保证了$p(x_i|c) \neq 0$恒成立。

整体流程

Naive_Bayes_Learn(*examples*)

```
begin
    for each class c do
        p̂(c) ← estimate p(c)
        for each attribute value x_i of each attribute x do
            p̂(x_i|c) ← estimate p(x_i|c);
        end
    end
end
```

Classify_New_Instance(*x*)

```
begin
    c_NB = arg max_{c∈C} P(c) ∏_{i=1}^{d} p(x_i|c)
end
```

Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$P(PlayTennis = y) = 9/14 \quad P(PlayTennis = n) = 5/14$$
$$P(Outlook = sunny|y) = 2/9 \quad P(Outlook = sunny|n) = 3/5$$
$$P(Outlook = overcast|y) = 4/9 \quad P(Outlook = overcast|n) = 0/5$$
$$P(Outlook = rain|y) = 3/9 \quad P(Outlook = rain|n) = 2/5$$
$$P(Temp = hot|y) = 2/9 \quad P(Temp = hot|PlayTennis = n) = 2/5$$
$$P(Temp = mild|y) = 4/9 \quad P(Temp = mild|n) = 2/5$$
$$P(Temp = cool|y) = 3/9 \quad P(Temp = cool|n) = 1/5$$
$$P(Humidity = high|y) = 3/9 \quad P(Humidity = normal|n) = 1/5$$
$$P(Humidity = normal|y) = 6/9 \quad P(Humidity = high|n) = 4/5$$
$$P(Wind = strong|y) = 3/9 \quad P(Wind = strong|n) = 3/5$$
$$P(Wind = weak|y) = 6/9 \quad P(Wind = weak|n) = 2/5$$

$$\text{New instance}: \langle sunny, cool, high, strong\rangle$$

$$P(y)P(sunny|y)P(cool|y)P(high|y)P(strong|y) = .005$$
$$P(n)P(sunny|n)P(cool|n)P(high|n)P(strong|n) = .021$$
$$\rightarrow v_{NB} = n$$

处理连续值

把连续值拆分成长度相等的区间即可。

Pros and Cons

- 优点
    - 训练和测试都非常快。训练的计算量很小。
    - 当独立的假设成立的时候，NB的表现会更好；在多类型预测的表现上也不错；
    - 易于维护，对于删除和添加数据集的数据的情况比较好处理（重新计数即可）。

- 缺点
  - 朴素贝叶斯的核心在于我们假设**属性x_i之间是相互独立的**，但往往事实并非如此，显然两个属性之间存在一定联系，这会带来一定误差；
  - 朴素贝叶斯分类器的复杂度是固定的而且比较低，所以可能会导致**欠拟合**（为了解决这种方法可以重新引入贝叶斯网络）；

应用场景

- 实时预测（因为NB的运行效率很高）；
- 多类型预测；
- 文字分类，比如垃圾邮件的判定。

Example of Spam Filtering